UNIVERSITY OF SOUTHAMPTON

Faculty of Engineering and Physical Science

School of Electronics and Computer Science

Developing Dynamic and Adaptive Risk-based Access Control Model for The Internet of Things

by

Hany Fathy Atlam

Thesis for the degree of Doctor of Philosophy

January 2020

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING
SCHOOL OF ELECTRONIC AND COMPUTER SCIENCE

Doctor of Philosophy

Developing Dynamic and Adaptive Risk-based Access Control Model for The Internet of Things

By Hany Fathy Atlam

The Internet of Things (IoT) is considered as the next stage of the evolution of the Internet. It promotes the concept of anytime, anywhere connectivity for anything. The IoT has the ability to connect billions of devices to share their information and create new services that improve our quality of life. Although the IoT provides countless benefits, it creates several security issues. One of the approaches to resolve these issues is to build an effective access control model.

Due to the dynamic nature of the IoT, static access control approaches cannot provide an appropriate security solution, as they are static and context-insensitive. Therefore, this research proposes a novel adaptive risk-based access control model to determine access permissions dynamically. This model performs a security risk analysis on the access request by using IoT contextual and real-time information to make the access decision. The proposed model has four inputs: user context, resource sensitivity, action severity and risk history. These inputs are used to estimate the risk value associated with each access request to make the access decision. In addition, this research adds abnormality detection capability by using smart contracts to track and monitor user activities during the access session to detect and prevent malicious actions.

One of the main problems to implement the proposed model was to determine the appropriate risk estimation technique that ensures flexibility and scalability of the IoT system. Hence, a review of most common risk estimation techniques was carried out and the fuzzy logic system with expert judgment was selected to implement the risk estimation process. In addition, to overcome scalability and learning issues of the proposed fuzzy risk estimation technique, Adaptive Neuro-Fuzzy Inference System (ANFIS) and Neuro-Fuzzy System (NFS) were utilized to implement the risk estimation technique. The results demonstrated that it outperformed the results produced by the fuzzy logic system, increased the accuracy and can adapt to changes of various IoT applications. In addition, this research presented a solution for the cold start problem associated with risk-based models that use risk history as one of the risk factors. The results demonstrated that the proposed risk-based model can operate immediately when first used or connected without reconfiguration or adjustment.

By using MATLAB Simulink, the operation of smart contracts was simulated to track and monitor user activities during the access session. The results demonstrated that it provides an effective way to detect and prevent malicious actions in a timely manner. To validate the applicability of the proposed adaptive risk-based model in real-world IoT scenarios, access control scenarios of three IoT applications including healthcare, smart home and network router were presented. The results demonstrated that the proposed risk-based model adds more advantages over existing access control models and can be applied to various and real-world IoT applications.

In memory of my father

You will forever remain alive in my heart and memory

I will forever love you and appreciate what you did for me.

I miss you every day

Acknowledgements

I am glad that Allah almighty has granted me the wisdom, strength, and good health to complete this thesis. I am sincerely grateful for many individuals who crossed my path during my studies, who provided me with their support and care.

My sincere thanks and gratitude to my supervisors, **Prof. Gary Wills** and **Prof. Robert Walters**, for their continuous support and invaluable advice and direction of my PhD study and related research. They have truly been my teachers, both in and outside of academic life. Their guidance helped me in all the time of research and writing of this thesis.

I would like to express my thanks to security experts who participated in my PhD research. I also would like to thank my friends and colleagues at the University of Southampton who made my whole university life such an enjoyable experience.

Last but not least, my deepest gratitude goes to my family, who were behind me every step of the way, my mother, my wife and other family members. You all are always my passion and power to fight, no matter what difficulties I face. All my honours are yours, forever. I am also thankful to my lovely daughters Shaden and Kady who were patient with me through this journey.

Declaration of Authorship

I, Hany Atlam, declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

Title of thesis: Developing Dynamic and Adaptive Risk-based Access Control Model for The Internet of Things

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:
    - Atlam, H. F., Wills, G. B (2019). An efficient security risk estimation technique for Risk-based access control model for IoT. *Internet of Things*, pp.1-20, Elsevier.
    - Atlam, H. F., Walters, R. J., Wills, G. B., & Daniel, J. (2019). Fuzzy Logic with Expert judgment to Implement an Adaptive Risk-based Access Control Model for IoT. *Mobile Networks and Applications*, pp.1-13, Springer.
    - Atlam, H. F., Alenezi, A., Alassafi, M. O., & Wills, G. B. (2018). Blockchain with Internet of Things: Benefits, challenges, and future directions. *International Journal of Intelligent Systems and Applications*, 10(6), pp. 40-48.
    - Atlam, H.F., Robert J.Walters, & Gary B. Wills. (2018). Fog Computing and the Internet of Things: A Review. *Big Data and* Cognitive *Computing*, 2(10), pp. 1–18.
    - Atlam, H. F., Walters, R. J., & Wills, G. B. (2018). Internet of Things: State-of-the-art, Challenges, Applications, and Open Issues. *International Journal of Intelligent Computing Research*, 9(3), pp 928–938.

- Atlam, H. F., Alenezi, A., Hussein, R. K., & Wills, G. B. (2018). Validation of an Adaptive Risk-based Access Control Model for the Internet of Things. *International Journal of Computer Network and Information Security*, pp. 26–35.
- Atlam, H. F., Walters, R. J., & Wills, G. B. (2018). Internet of Nano Things: Security Issues and Applications. *2nd International Conference on Cloud and Big Data Computing (ICCBDC 2018)*, pp 71–77.
- Atlam, H. F., Alassafi, M. O., Alenezi, A., Walters, R. J., & Wills, G. B. (2018). XACML for Building Access Control Policies in Internet of Things. In *Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security (IoTBDS 2018)*, pp. 253–260.
- Atlam, H. F., Alenezi, A., Alharthi, A., Walters, R., & Wills, G. (2017). Integration of cloud computing with internet of things: challenges and open issues. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 670–675).
- Atlam, H. F., Alenezi, A., Walters, R. J., Wills, G. B., & Daniel, J. (2017). Developing an adaptive Risk-based access control model for the Internet of Things. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 655–661.
- Atlam, H. F., Alenezi, A., Walters, R. J., & Wills, G. B. (2017). An Overview of Risk Estimation Techniques in Risk-based Access Control for the Internet of Things. In *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017)*, pp. 254–260.
- Atlam, H. F., Walters, R. J., & Wills, G. B. (2018). Intelligence of Things: Opportunities & Challenges. *3rd Cloudification of the Internet of Things (CIoT 2018), pp 1-6.*
- Atlam, H. F., & Wills, G. B. (2019). Intersections Between IoT and Distributed Ledger. In: Role of Blockchain Technology in IoT Applications. *Advances in Computers*, pp. 1-41, Elsevier.
- Atlam, H. F., & Wills, G. B. (2018). Technical Aspects of Blockchain and IoT. In: Role of Blockchain Technology in IoT Applications. *Advances in Computers*, pp. 1-35, Elsevier.
- Atlam, H. F., & Wills, G. B. (2019). IoT Security, Privacy, Safety and Ethics. In Digital Twin Technologies and Smart Cities, pp. 1-27, Springer.
- Atlam, H. F, Alenezi, A, Alassafi, M. O, Alharthi, A, Wills, G. B. (2020). Security, Cybercrime and Digital Forensics for IoT. In Principles of Internet of Things (IoT) Ecosystems: Insight Paradigm, Springer**,** pp. 551-577.

Signed:……Hany Atlam……

Date:…….....30/03/2020.....

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACL	Access Control List (ACL)
ACM	Access Control Matrix
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANOVA	Analysis of Variance
ANN	Artificial Neural Network
ABAC	Attribute-Based Access Control
BR	Bayesian Regulation
BOA	Bisector of Area
CoG	Centre of Gravity
CIA	Confidentiality, Integrity and Availability
CGF	Conjugate Gradient with Fletcher-Reeves Resrarts
DAC	Discretionary Access Control
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
EPR	Electronic Patient Record
FFBP	Feed-Forward Back Propagation
FIS	Fuzzy Inference System
GRBAC	Generalized Role Based Access Control
GUI	Graphical User Interface
ITU	International Telecommunication Union
IoT	Internet of Things
IWF	IoT World Forum
LOM	Largest of Maximum
LM	Levenberg-Marquardt
M2M	Machine-to-Machine
MAC	Mandatory Access Control

MIT	Massachusetts Institute of Technology
MAE	Mean Absolute Error
MAPE	Mean Average Percentage Error
MOM	Mean of Maximum
MSE	Mean Square Error
MF	Membership Function
MC	Mount Cedar
MLP	MuliLayer Perceptron
MFEP	Multi-Factor Evaluation Process
MLS	Multi-Level Security
NFC	Near Field Communication
NFS	Neuro-Fuzzy System
NVRAM	Non-Volatile Random-Access Memory
RFID	Radio Frequency Identification
ROM	Read-Only Memory
RLSE	Recursive Least Square Estimator
RAdAC	Risk Adaptable Access Control
RBAC	Role-Based Access Control
RMSE	Root Mean Squared Error
SCG	Scaled Conjugate Gradient
SOM	Smallest of Maximum
UHF	Ultra-High Frequency
WSN	Wireless Sensor Network
XACML	eXtensible Access Control Markup Language

Chapter 1: Introduction

During the past decade, the Internet of Things (IoT) has gained significant attention in academia as well as industry. The main reasons behind this massive interest are the unlimited capabilities that the IoT can provide (Perera et al., 2014). The IoT represents a revolutionary technology that enables almost everything everywhere to be connected over the Internet. It enables various devices and objects around us in the environment to be addressable, recognizable and locatable via cheap sensors. These devices can be connected and communicate with each other over the Internet using either wired or wireless communication networks (Leloglu, 2017). The IoT devices involve not only normal electronic devices or technological development products like vehicles, phones, etc, but also other objects such as food, animals, clothes, trees, etc. The key purpose of the IoT system is to allow various objects to be connected in anyplace anytime by anyone preferably using any path/network (Patel & Patel, 2016).

Although the IoT brought several benefits, it also creates multiple challenges, especially in security (Iqbal et al., 2016). Achieving a high level of security is a challenge due to the heterogeneous and distributed nature of the IoT system. In addition, applying sophisticated security algorithms could affect usability and user satisfaction due to resource constraints in IoT devices (Habib & Leister, 2015). One of the significant elements to handle security challenges in the IoT is the access control model. This model is used to control access to system resources by allowing only authorized users who have been successfully authenticated. An access control model consists of three main elements; subject, target and rules. Subjects are system users who make the access request to access system resources (targets). Rules are used to determine the access decision whether granting or denying the access (Dos Santos et al., 2014; Liu et al., 2016). The main purpose of the access control is to reject unauthorized users and limit operations of authorized users on a certain device. In addition, it prevents the activity that could cause a security breach (Dos Santos et al., 2014). A powerful access control model should fulfil security requirements of confidentiality, integrity, and availability (Suhendra, 2011).

There are two categories of access control approaches; static and dynamic. Static access control approaches use rigid and predefined policies to provide the access decision. These policies always give the same result in different situations. Hence, this rigid method cannot provide a reliable security solution for IoT systems, which are dynamic in nature (Chen et al., 2007). On the other hand, dynamic access control approaches use not only static policies but also real-time and dynamic features to determine access decisions. These dynamic features involve context, trust, history events, risk, and operational need (Shaikh et al., 2012).

Risk-based access control model is one of the dynamic models. It uses the security risk associated with the access request as a criterion to determine access decisions. Although this model is still in its first stage of approval, there is an increasing demand to specify its essential elements and procedures (Dos Santos et al., 2014). A risk-based model has many advantages. For instance, it provides more flexibility in accessing system resources by using real-time and contextual information collected while making the access request to decide whether granting or denying access. In addition, it takes into consideration the exceptional access requests that are necessary for medical and military applications where providing access can save thousands of lives (Khambhammettu, Boulares, Adi, & Logrippo, 2013). Also, it provides an efficient solution to unexpected situations which require violating the policy, as policies are incomplete and imperfect. The ultimate goal of the risk-based access control model is to create a system that encourages information sharing to maximize organization's benefits while keeping users accountable for their actions and stop the expected damage due to sensitive information disclosure (Chen et al., 2007).

1.1 Research Objective

The major goal of the IoT system is to increase information sharing and at the same time ensures that the highest possible security measures are applied to prevent sensitive information disclosure. However, current access control models are built using static and predefined policies that give the same result in different situations. This binary decision (grant/deny) cannot create an effective and efficient level of security in a dynamic, heterogeneous and distrusted environment like IoT systems (Castiglione et al., 2016; Shen et al., 2018). Therefore, the need to adopt dynamic access control approaches should be one of the essential priorities to provide an efficient and flexible access control model for the IoT. With billions of sensors in the IoT environment, several contextual and dynamic features can be collected to build a dynamic access control model. This, in turn, provides more flexibility to adapt to different situations and conditions while making access decisions in various IoT applications.

The objective of this research is to develop a dynamic and adaptive risk-based access control model for the IoT. This model utilizes real-time and contextual features collected while making the access request to determine access decisions. The proposed risk-based model uses user attributes related to

the surrounding environment such as time and location, sensitivity of data to be accessed by the user, severity of actions that will be performed by the user, and user risk history as inputs for the risk estimation algorithm to estimate the risk value for each access request to determine the access decision. In contrast to current access control models, the proposed model provides adaptive features by using smart contracts to track and monitor user's activities during access sessions to detect and prevent malicious actions.

1.2 Research Challenges

Risk-based access control model provides a flexible way to increase information sharing and at the same time ensures the security of information. After reviewing existing literature regarding risk-based access control models (see in Chapter 2 and 3), the literature failed to:

- Provide a dynamic risk-based access control model that can utilize contextual and real-time features collected at the time of making access requests and be adopted in various IoT applications.
- Present a clear risk estimation method to estimate the risk value associated with the access request quantitatively in a dynamic environment.
- Provide a scalable risk estimation technique that can cope with the constant increase in the number of IoT devices and adapt to changes of various IoT applications.
- Provide a plug and play risk-based access control model that intended to work perfectly when first used or connected, without reconfiguration or adjustment.
- Consider a way to detect and prevent malicious actions during access sessions.
- Provide a way to evaluate the accuracy and performance of the risk-based access control model in real-world IoT applications.

1.3 Research Questions

The major objective of this research is to provide a dynamic and adaptive risk-based access control model that uses real-time and contextual information to provide access decisions for various IoT applications. To achieve the research objective, the following research question and sub-questions need to be addressed:

RQ: What is the appropriate adaptive risk-based access control model for the IoT system?

This question is divided into six sub-questions:

SRQ1: What is the appropriate risk estimation technique to estimate the risk associated with the access request?

SRQ2: What are acceptable risk values to make the access decision in IoT applications?

SRQ3: How to provide plug and play risk-based model that can work when first used or connected to an IoT system?

SRQ4: How to provide a fast and scalable risk estimation technique to handle the constant increase in the number of IoT devices?

SRQ5: How will the user/agent activities be monitored during the access session?

SRQ6: To what extent is the proposed risk-based model applicable to real-world IoT scenarios?

1.4 Thesis Structure

Chapter 2 provides the background and literature review of access control models in the IoT. It opens by providing an overview of the IoT and its related security challenges. This is followed by providing a discussion of IoT security and access control models. Then, chapter 2 reviews the literature regarding risk-based access control models by highlighting risk factors and risk estimation approaches. This is followed by providing an overview of smart contracts by highlighting their structure and benefits and how smart contracts work.

Chapter 3 provides a discussion of risk estimation techniques. It starts by providing an overview of quantitative risk estimation approaches discussed in related risk-based models by highlighting their advantages and weaknesses. Then, chapter 3 provides an overview of the fuzzy logic approach with expert judgment and how it can be used to implement the risk estimation process of the proposed risk-based model.

Chapter 4 presents the proposed adaptive risk-based access control model for the IoT. It starts by discussing research problems the literature failed to address. Then, it introduces the proposed adaptive risk-based access control model by discussing its main elements and the process flow. This is followed by discussing how the proposed adaptive risk-based model will resolve research problems extracted from the literature. Then, chapter 4 presents research methods used in this research to address research questions.

Chapter 5 presents the implementation of the risk estimation process using the fuzzy logic system with expert judgment. It provides a step-by-step discussion of the implementation of the proposed risk estimation technique and how security experts have validated fuzzy rules and decided acceptable risk values of risk decision bands. This is followed by presenting

a solution for the cold start problem. Then, chapter 5 presents a set of experiments to evaluate the efficiency of the proposed fuzzy risk estimation technique to measure the response time with the different number of access requests and determine the most efficient MF, defuzzification method, and rule aggregation operator.

Chapter 6 presents the implementation of the proposed risk estimation technique using the ANFIS. It starts by providing an overview of the ANFIS. Then, it presents the implementation of the risk estimation technique using the ANFIS by showing different experimental results of training the ANFIS model using both hybrid and backpropagation learning methods.

Chapter 7 presents the implementation of the proposed risk estimation technique using the NFS. It starts by providing an overview of the NFS by highlighting its objectives and types of NFS methods. Then, chapter 7 presents the implementation of the risk estimation technique using the NFS by showing different experimental results of training the NFS model using four learning algorithms. Then, it compares the results of the NFS with the ANFIS and the fuzzy logic system.

Chapter 8 provides a discussion of access monitoring and model evaluation. It starts by providing an overview of access monitoring. Then, it discusses simulating the operation of smart contracts to monitor user activities during the access session by using Simulink. Then, chapter 8 discusses the evaluation of the proposed risk-based access control model by presenting access control scenarios of three IoT applications including healthcare, smart home and network router.

Chapter 9 summarises the main points of this research and presents the contribution and future work.

1.5 List of Publication

1.5.1 Journals

[1] Atlam, H. F., Wills, G. B (2019). An efficient security risk estimation technique for Risk-based access control model for IoT. *Internet of Things*, **Elsevier**, pp.1-20. https://doi.org/10.1016/j.iot.2019.100052

[2] Atlam, H. F., Walters, R. J., Wills, G. B., & Daniel, J. (2019). Fuzzy Logic with Expert judgment to Implement an Adaptive Risk-based Access Control Model for IoT. *Mobile Networks and Applications*, **Springer**, pp.1-13. https://doi.org/10.1007/s11036-019-01214-w

[3] Alassafi, M. O., Atlam, H. F., Alshdadi, A. A., Alzahrani, A. I, AlGhamdi, R. A, Buhari, S.M. (2019). A validation of security determinants model for cloud adoption in Saudi

organisations' context. *International Journal of Information Technology,* **Springer,** pp. 1-11. https://doi.org/10.1007/s41870-019-00360-4

[4] Alenezi, A., Atlam, H. F. & Wills, G. B. (2019). Experts Reviews on Cloud Forensic Readiness Framework for Organizations. Journal of Cloud Computing, **Springer**, pp. 1-14. https://doi.org/10.1186/s13677-019-0133-z

[5] Atlam, H. F., Walters, R. J., & Wills, G. B. (2018). Internet of Things: State-of-the-art, Challenges, Applications, and Open Issues. *International Journal of Intelligent Computing Research*, *9*(3), 928–938. https://doi.org/10.20533/ijicr.2042.4655.2018.0112

[6] Atlam, H. F., Alenezi, A., Alassafi, M. O., & Wills, G. B. (2018). Blockchain with Internet of Things: Benefits, challenges, and future directions. *International Journal of Intelligent Systems and Applications*, *10*(6). https://doi.org/10.5815/ijisa.2018.06.05

[7] Atlam, H.F., Robert J.Walters, & Gary B. Wills. (2018). Fog Computing and the Internet of Things: A Review. *Big Data and Cognitive Computing*, *2*(10), 1–18. https://doi.org/10.3390/bdcc2020010

[8] Atlam, H. F., Alenezi, A., Hussein, R. K., & Wills, G. B. (2018). Validation of an Adaptive Risk-based Access Control Model for the Internet of Things. *International Journal of Computer Network and Information Security*, Jan, 26–35. https://doi.org/10.5815/ijcnis.2018.01.04

[9] Hussein, R. K., Alenezi, A., Atlam, H. F., Mohammed, M. Q., Walters, R. J., & Wills, G. B. (2017). Toward Confirming a Framework for Securing the Virtual Machine Image in Cloud Computing. *Advances in Science, Technology and Engineering Systems*, *2*(4), 44–50. https://doi.org/10.25046/aj020406

[10] Atlam, H. F., Hemdan E.,& Wills, G.B. (2019) Interent of Things Forensics: A review. *Internet of Things,* **Elsevier** (Under Review)

1.5.2 Conferences

[1] Atlam, H. F., Walters, R. J., & Wills, G. B. (2018). Internet of Nano Things: Security Issues and Applications. *2nd International Conference on Cloud and Big Data Computing (ICCBDC 2018)*, pp 71–77. https://doi.org/10.1145/3264560.3264570

[2] Alenezi, A, Atlam, H. F., & Wills, G. B. (2019). IoT Forensics: A State-of-the-Art Review, Challenges and Future Directions. In *4th International Conference on Complexity, Future Information Systems and Risk*. https://doi.org/10.5220/0007905401060115

[3] Atlam, H. F., Alenezi, A ., Alzahrani, A. G. & Wills, G. B. (2020). IoT Big Data Analytics: Tools and Future Directions. *In Proceedings of the 5th International Conference on Internet of Things, Big Data and Security (IoTBDS 2018),* (In Press).

[4] Alzahrani, A. G., Alenezi, A ., Atlam, H. F. & Wills, G. B. (2020). A Framework for Data Sharing between Healthcare Providers using Blockchain. In *Proceedings of the 5th International Conference on Internet of Things, Big Data and Security (IoTBDS 2018)* (In Press).

[5] Atlam, H. F., Walters, R. J., & Wills, G. B. (2018). Intelligence of Things: Opportunities & Challenges. *3rd Cloudification of the Internet of Things (CIoT 2018, pp 1-6.* https://doi.org/10.1109/CIOT.2018.8627114

[6] Atlam, H. F., Alassafi, M. O., Alenezi, A., Walters, R. J., & Wills, G. B. (2018). XACML for Building Access Control Policies in Internet of Things. In *Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security (IoTBDS 2018)* (pp. 253–260). https://doi.org/10.5220/0006725102530260

[7] Atlam, H. F., Alenezi, A., Walters, R. J., Wills, G. B., & Daniel, J. (2017). Developing an adaptive Risk-based access control model for the Internet of Things. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 655–661). https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2017.103

[8] Atlam, H. F., Alenezi, A., Alharthi, A., Walters, R., & Wills, G. (2017). Integration of cloud computing with internet of things: challenges and open issues. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 670–675). https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2017.105

[9] Atlam, H. F., Alenezi, A., Walters, R. J., & Wills, G. B. (2017). An Overview of Risk Estimation Techniques in Risk-based Access Control for the Internet of Things. In *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017)* (pp. 254–260). https://doi.org/10.5220/0006292602540260

[10] Alenezi, A., Zulkipli, N. H. N., Atlam, H. F., Walters, R. J., & Wills, G. B. (2017). The Impact of Cloud Forensic Readiness on Security. In *Proceedings of the 7th International Conference on Cloud Computing and Services Science (CLOSER 2017)* (pp. 511–517). https://doi.org/10.5220/0006332705390545

[11] Zinopoulou, M., Atlam, H. F., N.H, N. Z., A., R., & Wills, G. B. (2016). Building on a Secure Foundation for the Internet of Things. In *IoT Security Foundation*.

1.5.3 Book Chapters

[1] Atlam, H. F., & Wills, G. B. (2019). Intersections Between IoT and Distributed Ledger. In: Role of Blockchain Technology in IoT Applications. *Advances in Computers.* **Elsevier** https://doi.org/10.1016/bs.adcom.2018.12.001

[2] Atlam, H. F., & Wills, G. B. (2018). Technical Aspects of Blockchain and IoT. In: Role of Blockchain Technology in IoT Applications. *Advances in Computers*, **Elsevier**, pp. 1-35. https://doi.org/10.1016/bs.adcom.2018.10.006

[3] Atlam, H. F., & Wills, G. B. (2019). IoT Security, Privacy, Safety and Ethics. In Digital Twin Technologies and Smart Cities, **Springer**, pp. 1-27. https://doi.org/10.1007/978-3-030-18732-3_8

[4] Atlam, H. F, Alenezi, A, Alassafi, M. O, Alharthi, A, Wills, G. B. (2020). Security, Cybercrime and Digital Forensics for IoT. In Principles of Internet of Things (IoT) Ecosystems: Insight Paradigm, **Springer,** pp. 551-577 https://doi.org/10.1007/978-3-030-33596-0_22

Chapter 2: Background & Literature Review

This chapter provides the background and literature review related to the research problem introduced in chapter 1. It opens by providing an overview of the IoT system by discussing IoT layered architecture, characteristics and common applications. Section 2.2 provides a discussion on IoT security by highlighting security requirements and challenges of the IoT system. Section 2.3 provides an overview of access control involving access control architecture and different types of access control approaches. Section 2.4 introduces context-awareness in the IoT system. Section 2.5 provides the literature review regarding risk-based access control models and its main elements. This review examines existing work regarding context-aware models, risk-based models and risk factors. Section 2.6 presents an overview of smart contracts by highlighting its main benefits and how it works. The chapter closes by providing a summary of the main points discussed through the chapter and introduces the next chapter.

2.1 An Overview of IoT

The IoT is considered as a universal presence that allows all objects/things in our environment to be connected over the Internet with the capability to interconnect with each other without human intervention. The IoT involves a variety of objects that can be connected using either wired or wireless networks. These objects have a unique addressing scheme that allows them to interact and cooperate with each other to create novel applications and services such as smart homes, smart transportation, connected cars, smart grids, smart cities, smart traffic control, etc., which improve our quality of life.

The IoT concept is not new, it has passed through several phases before reaching what it is today. The IoT notion starts in 1982 when four students from Carnegie Mellon University invented the ARPANET-connected coke machine to indicate whether drinks contained in the coke machine are cold or not. Their main idea was to count how many coke bottles had remained in each row and for how long. So, if the loaded bottle is left for a long time in the machine, it is labelled “cold”. This

experiment has inspired a lot of inventors all over the world to create their own connected appliances (Farooq & Waseem, 2015).

In the early 1990s, IBM scientists presented and patented an Ultra-High Frequency (UHF) Radio Frequency Identification (RFID) that covers wide distance and provides fast data transfer. Although IBM performed a few pilot experiments, it never commercialized this new technology. In the mid-1990s, IBM suffered from tough financial problems which made them sell their patent to Intermec, a barcode system provider, which utilized this technology to build multiple applications. However, due to the high cost of this technology at this time and low capacities of sales, this technology did not spread as was expected (Roberto et al., 2015).

In 1999, the Auto-IDentification Centre at the Massachusetts Institute of Technology (MIT) has received funds from various organizations to utilize RFID technology to link different objects together. This happened when two professors, David Brock and Sanjay Sarma, used RFID tags to track products through the supply chain. Their idea was to use RFID tag's serial number to track the products to save costs, since producing a more advanced chip with large memory storage will be more expensive. Data linked with RFID tags were kept in a database that can be accessed over the Internet.

Many researchers and organizations believe that the term "Internet of Things" was first introduced in 1999 by Kevin Ashton, who was the executive director of the MIT Auto-IDentification Centre (Ashton, 2009). Ashton has said, "*The Internet of Things has the potential to change the world just as the Internet did. Maybe even more so*" (Ashton, 2009). While others argue that Neil Gershenfeld is the first to speak about the idea of the IoT in his book entitled "*When Things Start to Think*" which published in 1999 (Gershenfeld,1999). The IoT was officially presented by the International Telecommunication Union (ITU) in 2005 (ITU, 2005).

The IoT has been defined by many organizations and researchers. However, the definition provided by the ITU in 2012 is the most common. It stated: "*a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving, interoperable information and communication technologies*" (ITU, 2012). In addition, Guillemin and Friess (2009) have suggested one of the simplest definitions of the IoT system, as shown in Figure 2.1. It stated: "*The Internet of Things allows people and things to be connected anytime, anyplace, with anything and anyone, ideally using any path/network and any service*".

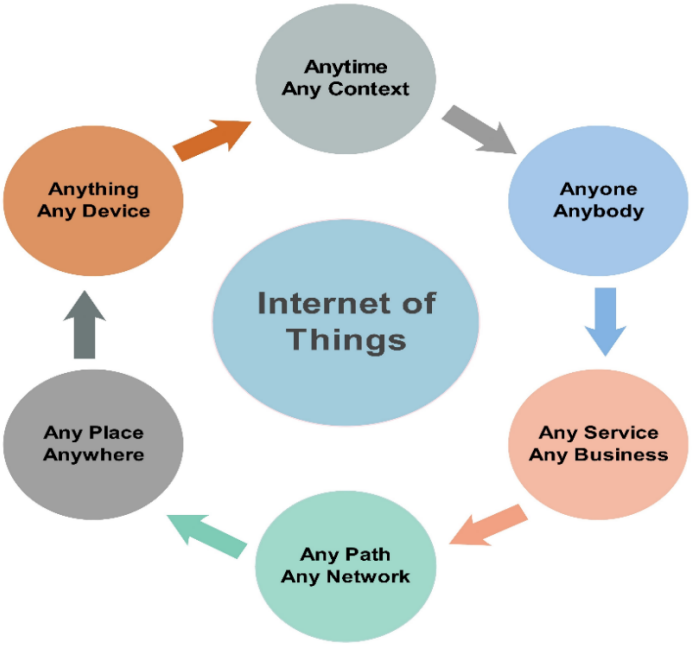


Figure 2.1: The IoT can connect anything anywhere using any path (Guillemin & Friess, 2009)

2.1.1 IoT Expansion

The IoT has the capability to modify business models and value chains in different organizations. It is not just a smart object connected to the Internet. In some stage, all objects will have the ability to connect and communicate over the Internet. The number of connected devices exceeds the population worldwide from 2008 (Statista, 2018). With unlimited capabilities and advantages of the IoT system, novel applications and services can be created every day. According to Statista (2018), the number of IoT devices is expected to reach about 31 billion worldwide by the end of 2020. This number is expected to increase significantly to reach about 75 billion devices by the end of 2025, as depicted in Figure 2.2.

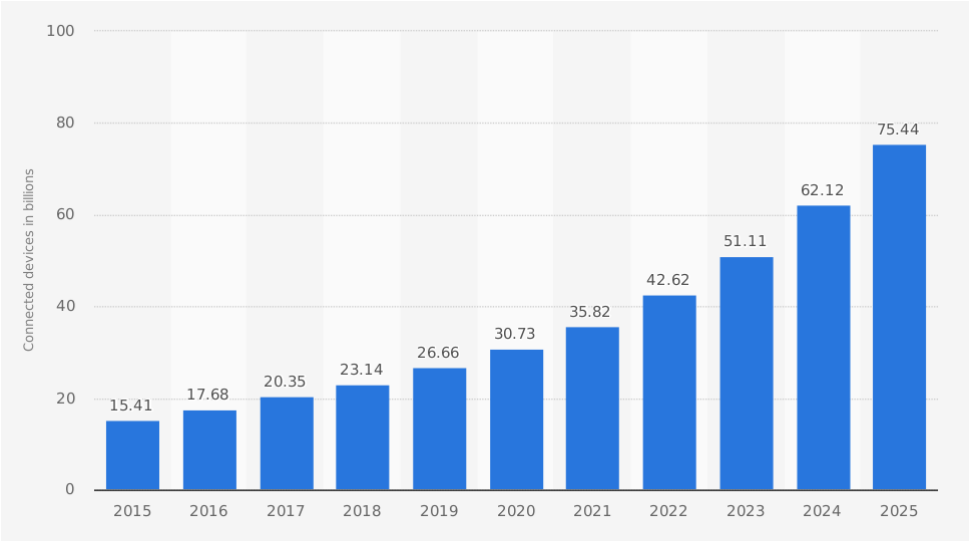


Figure 2.2: Expected IoT growth from 2015 to 2025 (Statista, 2018)

In addition, the IoT market is growing almost exponentially. According to Statista (2018), the estimated revenue of the IoT in 2015 was $ 743 billion. This number is expected to increase dramatically to reach $ 1710 billion by the end of 2019.

2.1.2 Architecture of IoT

The IoT World Forum (IWF) architecture committee released an IoT reference model in October 2014 (Stallings, 2015). This reference model works as a common framework to help the industry to accelerate IoT deployments. Also, it is intended to consolidate and encourage the collaboration and development of IoT deployment models. The IoT reference model is designed as seven-layers so that each layer provides additional information for establishing a common terminology, as shown in Figure 2.3. It also classifies where various types of processing are operated through different layers of the IoT reference model. Further, this model enables various manufacturers to build IoT products that are compatible with each other, which ultimately convert the IoT from a conceptual model into a real and approachable system.

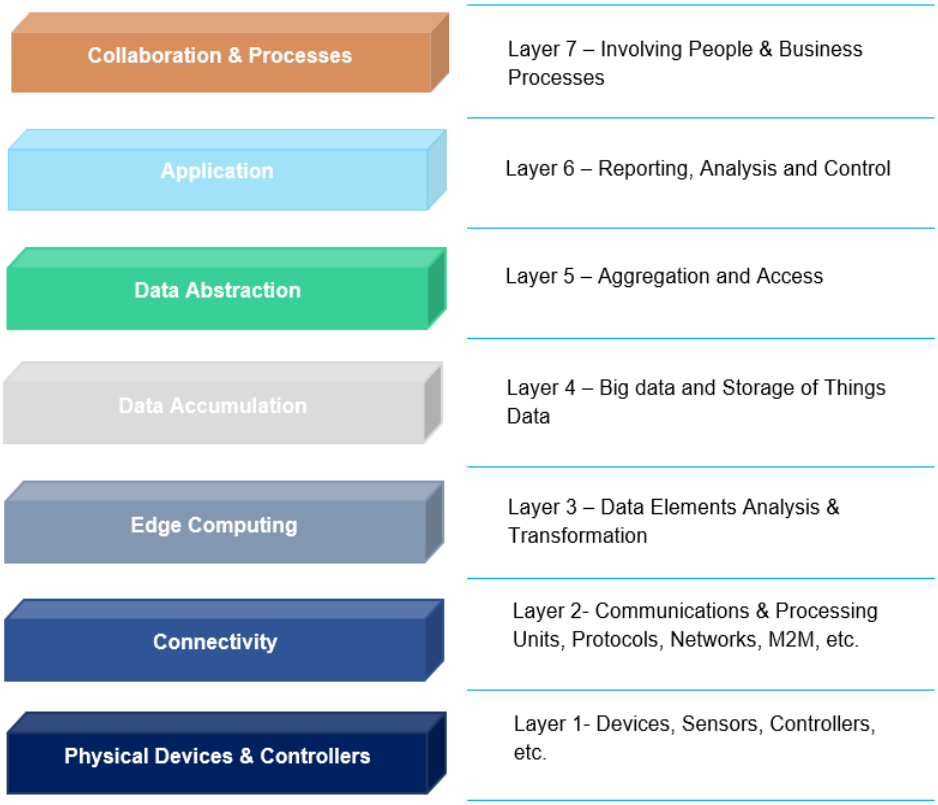


Figure 2.3: The IoT reference architecture layers (Stallings, 2015).

Layer 1 is the physical layer. It contains physical devices and controllers that manage various objects. These objects represent things in the IoT environment that involve various types of devices to collect, send and receive information. For instance, sensors that collect information about the surrounding environment (Cisco, 2014). Communications and connectivity are in layer 2. This layer is used to interconnect different IoT things with each other using interconnection devices such as switches,

gateway, router and firewalls. Layer 3 is the edge computing. This layer takes data coming from the connectivity layer and converts it into information appropriate for storage and higher-level processing. At this layer, the processing components work with a huge volume of data and it executes data transformation operations to reduce data size.

Layer 4 is the data accumulation. This layer is concerned with storing data coming from different IoT devices. These data are filtered and processed by the edge computing layer that absorbs large quantities of data and placed them in storage to be accessible by the higher levels. Different types of data in various formats and from heterogeneous processors may come up from the edge computing layer for storage. Layer 4 is the data abstraction layer. This layer aggregates and formats stored data in a way that make them accessible to different applications in a more manageable and efficient way. Layer 6 is the application layer. This layer is concerned with the information interpretation for various IoT applications. This layer encompasses a variety of applications that use IoT input data or control IoT devices (Stallings, 2015). The collaboration and processes are in layer 7. This layer identifies individuals who can communicate and collaborate to make the IoT system more useful.

2.1.3 Essential Characteristics of IoT

The IoT represents a promising technology that aims to improve people's quality of life by generating new applications and digitized services that facilitate people daily activities. There is a set of common features of the IoT system, which includes:

- **Large Scale**: IoT devices are counted in billions. This large-scale network needs to be controlled to allow devices to communicate with each other. In addition, this large-scale network generates a huge amount of data which produce a critical issue regarding data interpretation and analysis.

- **Intelligence**: Combining sophisticated software algorithms with hardware allow IoT devices to become smart. These abilities allow IoT devices to make intelligent decisions in various situations and interact intelligently with other communicating devices.

- **Sensing**: Sensors are the main part of the IoT system. These sensors are used to perceive changes in the surrounding environment and generate data that reveal their status. With various sensing technologies, sensors provide a good understating of the surroundings which increase human awareness about the physical world.

- **Dynamic Environment**: The IoT can connect almost all objects of our environment without being able to determine the IoT network boundaries which makes it a dynamic system in nature. Also, IoT devices can operate and be adjusted dynamically based on changing conditions and situations.

- **Heterogeneity**: The IoT system involves billions of devices with heterogeneous features such as operating systems, platforms, communication protocols and others. These heterogeneous features make the management operation a complex task to perform.

- **Lightweight**: Most IoT devices are designed to be small and lightweight with limited capabilities of memory storage and computation power, so they are built to work with minimal energy consumption.

- **Connectivity**: One of the main features of the IoT system is the ability to connect various devices with different characteristics and use their shared information to create novel applications and services.

- **Self-configuring**: Devices need to be configured to perform a specific task. But for IoT devices, they have the capability of self-configuring which enable them to operate without the human intervention. These devices could configure themselves to the up-to-date software in association with the device manufacturer without user involvement (Stallings, 2015). For example, mobile phones now can be upgraded automatically to up-to-date software without user involvement.

- **Unique Identity**: Within the IoT network, each IoT object is identified and recognized using a unique identifier such as the IP address. These identities are provided by IoT manufacturers to use it to upgrade devices to appropriate platforms. Further, these devices have interfaces that enable the users to collect the required information from devices, record their status and manage them remotely.

- **Context-awareness**: In the IoT environment, there are multiple sensors that sense their surroundings, collect and store the required information. These sensors may take decisions based on the collected data which make it a context-aware**.**

2.1.4 IoT Applications

The IoT system can interconnect almost all physical and virtual objects in our environment that yield new services and applications. These applications can be adopted in different domains to increase our quality of life. This section provides a discussion on common IoT applications.

2.1.4.1 Healthcare

The IoT has proven it can provide several benefits for the healthcare domain by creating new services that help patients and keep the field innovative. There are multiple wearable devices developed to monitor and track the patient’s health conditions. These devices help older patients to live independently without fear. Also, these devices can be utilized to observe and store patients’ health conditions constantly and send warning messages in abnormal situations in which if the situation is

minor, the device itself can recommend a treatment for the patient. While if it is a major situation, the device can send urgent messages to the hospital or ambulances to be immediately dispatched (Akkaş & Sokullu, 2017).

2.1.4.2 Smart Agriculture

With the existence of multiple sensors within the IoT environment, farmers can use collected data to produce a better return on the investment. Using sensors, the soil parameters such as humidity, salt level and temperature can be collected and measured to increase productivity. Furthermore, with the existence of several wireless technologies such as geographical information system and remote sensing, there are many chances to collect relevant information about the soil quickly and efficiently which can help to substitute human effort with automatic machinery to increase agricultural production (Krishna et al., 2017). There is significant growth in the adoption of IoT devices in agriculture. It is predicted that the number of IoT devices in agriculture will reach about 75 million by the end of 2020 (Akkaş & Sokullu, 2017).

2.1.4.3 Supply Chain and Logistics

Using RFID and Near Field Communication (NFC), products can be tracked from the manufacturer to the distribution location. RFID tags attached to the products are used to identify each product uniquely and collect relevant information and convey it in real-time along with location information. These tags are used to transmit messages showing exactly what products, sizes and style variations as well as temperature and humidity of products. Moreover, automated data capture gives real-time visibility of stock and avoids manual counting and human errors. In simple words, the IoT is set to revolutionize the supply chain with both operational efficiencies and revenue opportunities (Guo et al., 2012).

2.1.4.4 Smart Home

Smart home is one of the most popular applications of the IoT system. Thanks to sensor and actuation technologies along with Wireless Sensor Networks (WSNs), people can connect a variety of smart appliances inside their homes to resolve their interests. Smart homes offer greater energy-efficiency in which smart appliances can be set to automatically run and then turn off when their job is done. For example, lights can shut off automatically when no one is in the room. Also, the thermostat can be set to let the indoor temperature drop during the day before returning it to a more comfortable level just before residents arrive in the evening (Pătru et al., 2016).

2.1.4.5 Smart City

A smart city refers to the adoption of IoT devices such as sensors, meters, lights, etc. to monitor and collect information about the surrounding environment of a city to provide new digitized services to improve public services and city infrastructure. IoT solutions are involved in many areas of smart

cities such as smart street lighting, trash management, smart parking and traffic management (Zanella et al., 2014).

For the smart traffic, collected sensor information about traffic can be sent to citizens' phones to monitor traffic in real-time and allow drivers to choose the best roads to save driving efforts and time. Also, drivers can be warned in the case of accidents to redirect away from congestion. For trash management, IoT sensors can be deployed across trash bins to send messages to specific authorities to report bins that need to be emptied (Khatoun & Zeadally, 2017).

2.1.4.6 Smart Grid

IoT sensors can be utilized to collect relevant information about energy consumption in homes to use energy efficiently and save money. For example, suggesting better ways to save energy. Also, IoT sensors information can be used to deliver consumers all relevant information about various energy suppliers in an automated way for choosing the best for consumers.

The concept of the smart grid adds intelligence at the power flow cycle from supplier to consumer. This type of intelligence can be used to help consumers to be aware of power consumption and dynamic pricing. Also, one of the main applications of the smart grid is the smart meter which collects, records and analyses power consumption at different times of the day. This information can be used by consumers to adjust their power consumption and change their lifestyles to reduce costs (Zanella et al., 2014).

2.1.4.7 Connected Car

Smart car or connected car started to be deployed into our community. This type of cars can access the Internet and share their data with other devices. The number of cars equipped with this facility is increasing every day, which will allow the appearance of several applications for connected cars in the near future (Kalmeshwar & Prasad, 2017). The connected car provides several advantages over the normal one. It can reduce car accidents and decrease car drivers' errors by allowing the driver to operate the car remotely. These driverless cars also can save time and reduce driving stress. Several car manufacturers such as BMW, Ford and Volvo have confirmed that there will be fully autonomous cars by the end of 2021 (Xu et al., 2014).

2.1.4.8 Wearables

Wearables have a huge interest in markets all over the world. Many companies started to produce these devices in huge quantities to satisfy increased demands including Google and Samsung. According to Statista (2018), the number of connected wearable devices is expected to reach 830 million at the end of 2020. Wearables devices are equipped with sensors and can connect to the Internet for data sharing. These sensors collect data about the user which is later processed to extract

meaningful information. Most common wearables devices are in fitness, health and entertainment (Cirani & Picone, 2015).

2.2 IoT Security

Security is one of the major challenges standing as a barrier in the way of successful adoption of IoT applications. The value of the IoT system comes from connecting all small and large systems together and allowing them to communicate with each other over the Internet. Securing IoT devices and data transmission and communication should be one of the fundamental priorities to consider (Elkhodr et al., 2013b).

The IoT is a dynamic system in nature in which every poorly secured object can disturb the security and resilience of the entire system as IoT devices are connected like a chain. The ease of connection and access of IoT devices opens doors for severe security issues especially with the large-scale distribution of heterogeneous devices, their ability to connect to other devices without requesting permissions or even notifying their owners and probability of flooding these devices with severe security threats.

Handling security challenges in the IoT context should be an essential priority to increase adoption of IoT applications. Users need to be fully confident about the security of their IoT devices and their related applications. They need to ensure that their devices are effectively secured from various known threats as IoT devices become more integrated into people daily life's activities (Iqbal et al., 2016).

2.2.1 Security Requirements for IoT

Security of the IoT system can be improved by employing classical security measures. Typical CIA (Confidentiality, Integrity, and Availability) security requirements should be employed to provide a secure IoT system.

Confidentiality means exchanging messages between a sender and receiver should be protected against any malicious or unauthenticated user (Maple, 2017). For the IoT system, confidentiality need not only be guaranteed inside communication networks but also when transmitting messages between various IoT devices. While integrity is used to guarantee that the content of messages exchanged between the sender and receiver are protected against any manipulation or modification. In the IoT system, integrity checks can be carried out at each node involved in the message exchange between the sender and receiver. Availability is used to guarantee that a malicious user is not capable of disrupting or harmfully affecting communication or quality of service provided by IoT devices or communication networks (Yu et al., 2016).

Although CIA security measures are essential for the IoT, there are other security requirements that need to be implemented for each level of the IoT architecture, as shown in Figure 2.4. Node authentication is one of the main security issues in the IoT physical layer to avoid unauthenticated node access and keep the communication channel between IoT nodes secure from various types of attacks. So, a lightweight cryptographic algorithm is needed to encrypt transmitted data especially for resources-constrained IoT devices (Suo et al., 2012).

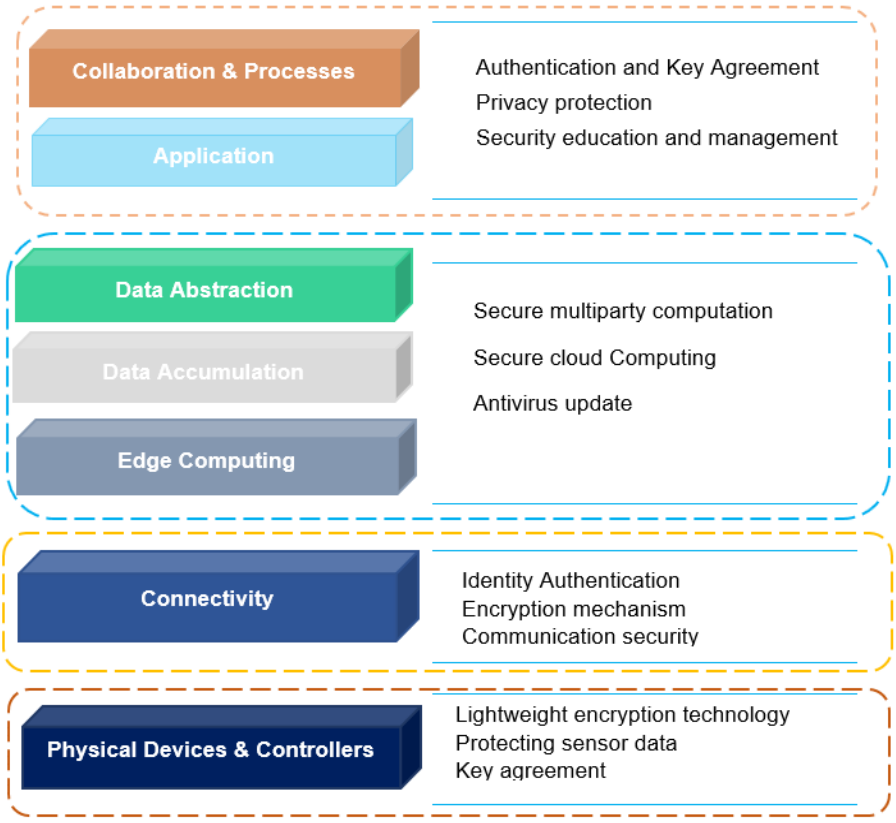


Figure 2.4: Security requirements at each level of the IoT architecture

For the connectivity layer, communication security measures are needed as well as identity authentication to prevent access of illegal nodes. Also, Distributed Denial of Service (DDoS) attack is common at this level, so there is a need to protect against this attack (Abdur et al., 2017). For data abstraction, accumulation and edge computing levels, many application security mechanisms are needed to secure data stored in Cloud computing. Strong encryption algorithms are needed besides up-to-date anti-virus. For the application and collaboration level, there is a need to adopt authentication and key agreement to protect user's privacy. Moreover, security awareness and password management are essential for information security at this level (Suo et al., 2012).

2.2.2 Security Challenges

Like all new technologies, security is still the biggest issue that stands in the way of effective developments of the IoT system. There are several security challenges that need to be addressed to

increase peoples trust in adopting IoT devices. This section provides a brief discussion of common security challenges in the IoT system.

2.2.2.1 Resource Limitations

Most IoT devices have limited processing and storage capabilities due to small and lightweight features, which make them run on low energy consumption. Therefore, sophisticated security algorithms are not suitable for these resource-constrained devices as they are not able to execute complex processing operations in real-time. Instead, they can only employ fast and lightweight security algorithms (Musaddiq et al., 2018).

2.2.2.2 Big Data

The IoT system involves billions of devices which generate a huge amount of data. These data are variable in term of structure and often arrive in real-time. The volume, velocity, and variety characteristics of Big data raise issues regarding storage and analysis operations. The IoT system is considered one of the main sources of Big data. Although Cloud computing provides a good solution for data storage for a long time, processing this massive volume of data is a substantial challenge, as the entire performance of various applications is significantly dependent on the data management service. Moreover, one of the major issues of Big data is data integrity. Ensuring the security and integrity of this huge amount of data is becoming difficult as data sources massively increased in a way that more security measures need to be adopted (Waqas Aman, 2013).

2.2.2.3 Secure Communication

Securing IoT devices is not enough to ensure full security of the IoT system. Instead, the communication channel connecting various nodes such as IoT devices and cloud services need to be protected from various types of attacks. Most IoT devices transmit their data in a plaintext format without being encrypted which make it an easy target to various types of network attacks. Hence, a proper encryption technique should be employed (Maheshwari & Dagale, 2018). Also, using separate networks can increase security by isolating devices and creating private communication channels.

2.2.2.4 System Resilience

System resilience is one of the main challenges that need to be addressed in the IoT. Resilience refers to the ability of the system to respond to unpredicted situations without regressing (Kitchin & Dodge, 2017). Hence, if an IoT device is compromised, the system should be able to protect other network nodes. However, in the normal case, if there is an infected device, resetting it or even replacing it can solve the issue, but the main problem in the IoT system is that there is a network of devices that make identifying the compromised device or fixing the issue to maintain the overall system security a very difficult task to achieve. So, there is a need for a systematic method to restore IoT devices from a known state as well as providing an efficient tool to isolate compromised devices (Kitchin & Dodge, 2017).

2.2.2.5 Digital Forensics

With billions of IoT devices, the IoT system has become a significant source of evidence which can provide vital information from the physical world to help investigators throughout the investigation process, however, there are issues. For example, it's important to identify where data is generated and where it is stored, which is difficult to determine in the IoT context. Further, since IoT devices have limited storage, data sent to Cloud computing violates data persistence (Zhang et al., 2014). In addition, the dynamic and heterogeneous nature of the IoT system enables the integration of various domains such as computers, tablets, mobile devices, Cloud computing, various types of sensors and RFID technologies. So, investigating an incident in the IoT will involve all these domains which add more complication in the investigation process (Zia et al., 2017).

2.2.2.6 Heterogeneity

The IoT system is a heterogeneous system in nature. It comprises various devices with different hardware and software capabilities. These devices were built by different manufacturers with little security in mind, which makes them an easy target for attackers. Also, if these devices depend on open-source software with threats, updating their firmware will be hard (Alur et al., 2015).

2.2.2.7 Authentication and Access Control

Providing an efficient authorization and access control mechanism for the IoT system is one of the major fundamentals to provide a secure system. IoT devices should gain access to services or applications only after providing their identities correctly. However, there are many problems associated with device authentication such as the use of weak or default passwords that lead to giving access to attackers who can manipulate device data or even physically damage it. Adopting security by design in IoT devices, enabling two-factor authentications and enforcing the use of strong passwords can help to resolve these challenges (Habib & Leister, 2015).

2.3 Access Control

The main purpose of the access control is to deny operations performed by unauthorized users. Also, it tries to prohibit any activity that could cause a security breach (Dos Santos et al., 2014). A powerful access control model should satisfy the security requirements of confidentiality, integrity, and availability (Suhendra, 2011). It is essential to make a reasonable distinction between authentication, authorization and access control. Authentication is defined as the operation of seeking to verify the identity of a user (Hulsebosch et al., 2007). While allowing or denying access to an authenticated user to perform certain operations on certain resources is called authorization. Access control is the process of enforcing authorization policies. Once a user/agent is authenticated and the authorization level is identified, access control is used to enforce user permissions to prevent user/agent from accessing anything that he/she should not be allowed to (Suhendra, 2011).

The history of the phrase "Access Control" has started in transportation in the first half of the twentieth century. The concept of the limited-access road was suggested in 1907 to control fast-growing motor traffic. Although early cars were not as fast as today's standard, car's drivers were enforced to control their speed on highways. They were enforced to enter and exit via one-way ramps to control the access to highways which lead to a reduction in the probability of cross-traffic accidents and increases the speed of traffic flows (Houlis, 2018).

By the early of the 1960s, electronic solutions adopt access control to address the problem of lost keys to restrict access only for specific individuals. Early access control solutions utilized basic keypads with personal identification numbers to gain access; this was then updated to swipe cards and has since been developed into the key cards, which are still being used today.

Currently, access control is implemented at different levels in many areas such as operating system and database management system to control resources and allow only legal users/agents to use system resources in an authorized way. An access control model consists of five main elements; subjects, objects, actions, privileges, and access policies.

- **Subjects*:*** Active entities in the form of users and processes that request access to objects.
- **Objects**: Passive entities containing information being accessed by subjects.
- **Actions**: An operation to be performed on a certain object such as read, write, execute, etc.
- **Privileges**: Authorizations permissions to perform certain actions on certain objects.
- **Access policies**: The set of rules that determine the access decision whether granting or denying access.

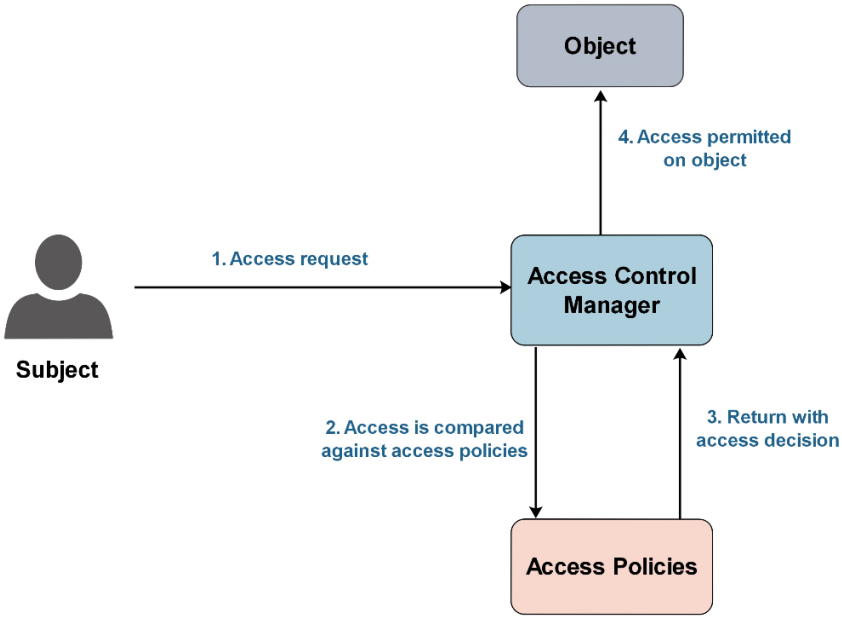


Figure 2.5: Flow of an access control operation

The flow of an access control process is shown in Figure 2.5. It starts when a subject/user send an access request to the access control manager to access a certain object. Then, the access control

manager compares the subject's credentials against access control policies to determine the access decision. The decision will be either granted or denied. If the access is granted, the access control manager will allow the user to access the object, while if the access is denied, the access control manager will terminate the session (after sending warning message regarding insufficient credentials).

2.3.1 Access Control Architecture for IoT

The main issue associated with building an access control model for the IoT is the lack of ability to process access request and make the required decision as IoT devices are resource-constrained with limited storage and computation capabilities. Typically, there are three ways to implement an access control model for the IoT system; centralized, centralized and contextual, and distributed (Hernández-Ramos & Jara, 2013).

2.3.1.1 Centralized Approach

In this approach, the access control logic is located at a central entity. This entity could be a server with a direct communication to IoT devices that it manages or another entity in a different location. Therefore, IoT devices send their collected data to the central entity that is responsible for making access control decisions, as depicted in Figure 2.6. The key advantage of this approach is that the access control logic is located in an external entity without constraints of resources, which enable the use of standard security and advanced access control technologies (Hernández-Ramos & Jara, 2013).

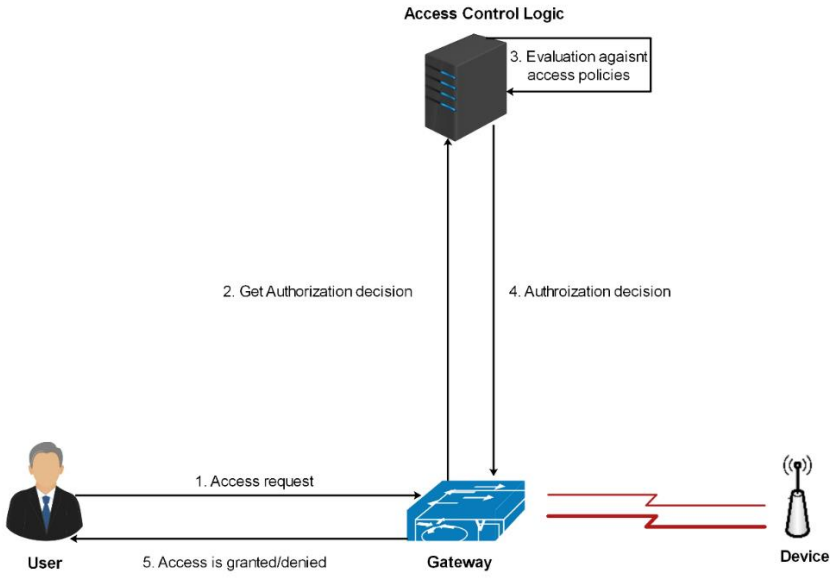


Figure 2.6: Access control flow in the centralized approach (Hernández-Ramos & Jara, 2013)

On the other hand, there are some major drawbacks associated with the centralized approach. Since IoT devices send their data to the central entity, access decisions are not based on the contextual information related to IoT devices itself. Also, end-to-end security is compromised since a central

entity is needed to determine access decisions. Therefore, this entity will need to view the content of the access query, which also compromises the privacy of the requester. Moreover, since a single entity stores and manages all data coming from different IoT devices, it becomes a single point of failure in which an attacker can compromise a huge volume of sensitive and confidential information.

2.3.1.2 Distributed Approach

In this approach, the access control logic is embedded into IoT devices. These devices are being provided with the necessary resources to obtain, process and send information to other services and entities. Therefore, IoT devices make access decisions without the need for a central entity. The flow process of an access control process using the distributed approach is shown in Figure 2.7. The use of the distributed approach provides some key advantages. For instance, IoT devices are no longer passive entities, they have the capability to manage their information. Also, the elimination of the central entity enables end-to-end security for the access request and eliminates the single point of failure (Roman et al., 2013).

The most noticeable issue with this approach is the need to extend IoT devices with an access control logic. Also, the implementation of static access control models will be difficult in resource-constrained IoT devices. Subsequently, this approach must be investigated in-depth by analysing the feasibility of different access control models or implementing new proposals that meet the demands of the distributed approach (Hernández-Ramos & Jara, 2013).

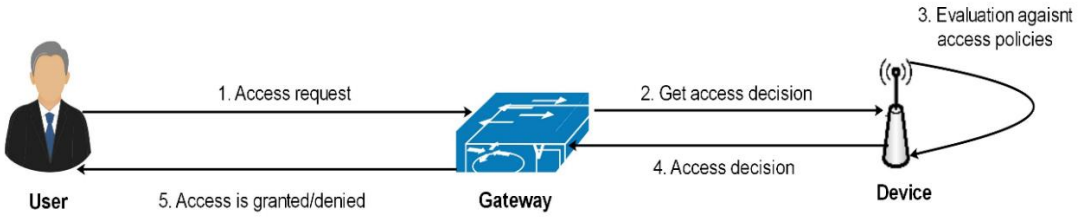


Figure 2.7: Access control flow in the distributed approach (Hernández-Ramos & Jara, 2013)

2.3.1.3 Centralized and Contextual Approach

This is a hybrid approach in which IoT devices are not completely passive entities since they participate partially in access decisions. The access control logic is implemented at a central entity as in the centralized approach, but the contextual features from IoT devices are sent to the central entity. These features are used to make access decisions, as shown in Figure 2.8. This approach is not feasible without providing contextual information of IoT devices at the time of the access request in terms of location and environmental status. If there is a delay when IoT devices transmit data to the central entity, the value obtained by the IoT device will be different at the time of making the access decision.

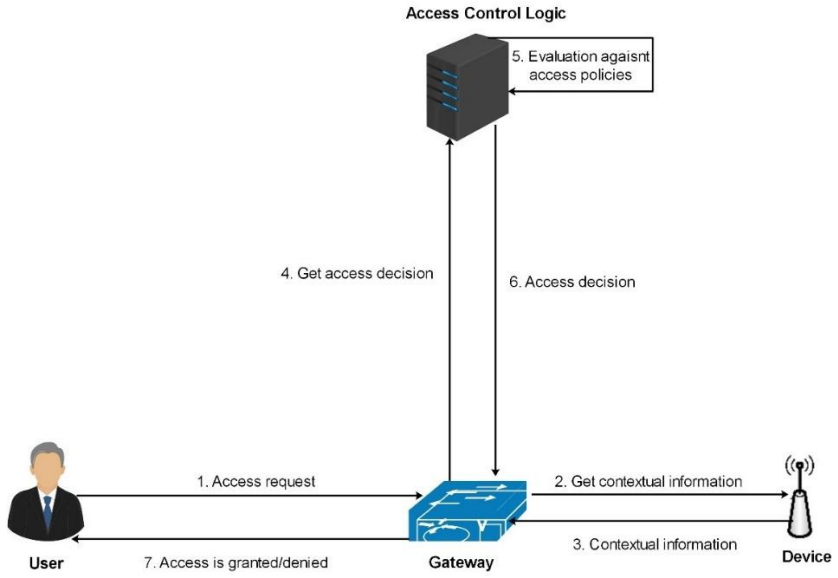


Figure 2.8: Access control flow in the centralized and distributed approach (Hernández-Ramos & Jara, 2013)

2.3.2 Access Control Models

Providing an efficient access control mechanism for the IoT system is one of the major fundamentals to provide a secure system. IoT devices should gain access to services or applications only after providing their identities correctly (Lee, 2015). To ensure confidentiality and integrity of system resources, the access control is used to guarantee that authorized users granted appropriate access permissions (Langaliya & Aluvalu, 2015). There are many access control models, which can be divided into two classes: static and dynamic access control models.

2.3.2.1 Static Access Control Models

Static access control (also called classical/traditional access control) models are rigid in nature as they depend on predefined policies that always give the same outcome regardless of the situation. They are context insensitive.

Although static access control approaches were successfully applied in different environments to solve various problems, these approaches are designed to provide a relationship between information associated with an access control rule logic and a resource for which access is requested. The implementation of an access control approach is subject to manipulation, which can range from an unexpected situation, including poorly written access policies to a number of malicious entities acquiring access to a set of existing accounts. Therefore, static access control approaches provide a set of advantages, but they also have drawbacks. One of these drawbacks is that it can not handle unpredicted situations as they are based on static and predefined policies (Metoui, 2018).

The next section provides an overview of the main types of static access control models by highlighting benefits and drawbacks of each approach.

2.3.2.1.1 Access Control List

Initially, access control was implemented as a table called Access Control Matrix (ACM), where each row and column is composed of a subject and object respectively. Each record represents a set of access rights for the corresponding subject (Mahalle et al., 2013). Then, the Access Control List (ACL) appeared. ACL is a list of certain objects which contains legitimate subjects along with their access rights. ACLs are the default representation of access rights on UNIX systems. Although ACL is efficient and effective, it is not scalable with a large number of subjects and objects. Also, it is difficult to modify multi-object rights for individual users (Hu et al., 2006).

2.3.2.1.2 Discretionary Access Control

Discretionary Access Control (DAC) is designed for multi-user databases and systems with few previously known users. All system resources are under full control from the user. DAC grants access depending on the user identity and authorization which are defined through open policies. The owner of the resource can grant access to any user. DAC mainly deals with the inheritance of permissions, user-based authorization, auditing of system events and administrative privileges (Langaliya & Aluvalu, 2015).

The key advantage associated with the user of DAC is the capability to provide fine-grained control over system objects. Also, DAC is easy to implement and provides a flexible way to allow system owners or system administrators to create customized access policies for each user. For example, a user can be granted read and write access as well as read-only access to another user for the same resource.

On the other hand, DAC introduces some issues. Enabling users to have full control over object access permissions opens the system to various vulnerabilities such as Trojan Horse. Also, maintenance of the system and verification of security principles are enormously hard for the DAC systems since users dominate access rights to owned objects.

2.3.2.1.3 Mandatory Access Control

In the Mandatory Access Control (MAC) model, each object is assigned a label which specifies security privileges of the object based on the sensitivity of information of the object. Also, each subject is assigned a label that specifies which object the requester can access (Bugiel et al., 2013; Hulsebosch et al., 2005). MAC model provides necessary security measures where a user can only perform tasks related to his/her privileges. In the MAC, the security policy is controlled by a security policy administrator and the user does not have the capability to override it. MAC is mainly concerned with the confidentiality and integrity of information, so it primarily implemented in military and government applications. (Zhu & Jin, 2007).

Compared to DAC, MAC is not vulnerable to Trojan Horse since users do not have the ability to declassify information. Also, MAC is straightforward and is considered a good model for commercial

systems that operate in hostile environments, where the risk of attack is very high, confidentiality is a primary access control concern or the objects being protected are valuable (Vijayakumar et al., 2012).

MAC by far is the most secure access control environment but does not come without a price. MAC requires a considerable amount of planning before it can be effectively implemented. Once implemented, it imposes a high system management overhead due to the constant need to update object and account labels to accommodate new data, new users and changes in the categorization and classification of existing users (Jin et al., 2012).

2.3.2.1.4 Role-based Access Control

Role-Based Access Control (RBAC) is a widely accepted model in almost all large enterprises (Bijon et al., 2013). RBAC model consists of three elements: users (subjects requesting access), roles (collections of permissions) and operations (actions on target resources). Access permissions are related to roles and the appropriate role is granted to the user. A single user can be associated with one or more roles, and a single role can include one or more user. RBAC provides a classification of users based on their roles (Kumar et al., 2002).

RBAC model restricts access to objects based on the subject's role rather than their identifications. Roles are allocated to subjects according to their clearance, qualification, and responsibilities inside the organization. A set of permissions is grouped together to form a role. A user can be allocated to different roles and the role can be assigned to different users. RBAC model might have many users, each user will be assigned to a specific role or may be assigned to multiple roles and each role consists of a set of permissions/rights. An example of RBAC in a hospital, where doctors can both read and write prescriptions, whereas pharmacists are limited to read prescriptions only. RBAC helps to ensure system integrity and availability by explicitly controlling not only which resources can be accessed but also how access can occur. Also, in large organizations, the consolidation of access control for many users into a single role entry allows for much easier management of the overall system and much more effective verification of security policies (Bijon et al., 2013).

Although RBAC provides great advances in access control, the administrative issues of large systems still exist. In large systems, memberships, role inheritance, and the need for fine-grained customized privileges make the administration process potentially impractical. Additionally, RBAC cannot be used to ensure permissions on sequences of operations (Sandhu et al., 1996).

2.3.2.2 Dynamic Access Control Models

The core principle of dynamic access control models is that they take into consideration not only access policies to make access decisions, but also dynamic and contextual features which are estimated at the time of the access request (Wang & Jin, 2011). This provides more flexibility and can adapt to different situations and conditions while making the access decision.

The need to adopt dynamic access control approaches should be one of the essential priorities to provide efficient and flexible access control model. However, most existing access control approaches are relying on static and rigid access policies and manual processes. These approaches are unable to provide a roadmap to improve automation significantly. This lack of automation leads to a heavy involvement of human analysis, which is costly, error-prone, and vulnerable to various types of attacks based on social engineering. Additionally, current classical approaches have issues with resolving risks and threats in real-time, especially when handling a previously unidentified threat. This is because these approaches make their access decision based on a set of policies built by a security analyst, who cannot resolve different access control situations in real-time, but can deal only with problems that were recognized before (Brooks et al., 2012).

In addition, existing access control approaches lack feedback and possible options for resolving access control situations when a legitimate user or agent is unable to continue its activity due to the necessity of access to a requested resource or service when access control refuses this access for some reason. One of the common messages regarding denied access attempt states, that 'access was denied' without providing any other details. Such a message causes the user to ask a system administrator to make an exception for his/her activity, which interrupts ongoing business processes and increases the load on system administrators. Moreover, static access control approaches suggest that there is a need for a system administrator role with wide access to services, data and unrestricted access with respect to time. However, compromised access to an account with the system administrator role leads to exposing the entire system to malicious actions, and if the account is widely used, it is not possible to limit the risk of such an account being compromised (Suhendra, 2011; Zhou et al., 2013).

Instead of static policies, dynamic access control approaches use real-time information and features to make access decisions. These real-time features can include trust, risk, context, history and operational need. These dynamic models adapt to different situations and conditions while making access decisions (Li et al., 2008; Shaikh et al., 2012).

2.4 Context Awareness in IoT

With great developments in the sensor technology, sensors are becoming an integral part to sense and collect relevant information about the environmental features. Sensors are getting more powerful, cheaper and smaller in size, which have stimulated large-scale deployments. With the massive increase in the number of sensors, huge amounts of data are generated. The data needs to be analysed and interpreted to extract meaningful information. Context-aware computing has played a vital role to resolve this issue. It facilitates storing context information linked to sensor data, so the interpretation can be done easily and more meaningfully. In addition, understanding context makes it easier to perform Machine-to-Machine (M2M) communication which is a core element in the IoT vision (Perera et al., 2014).

Context-awareness is an essential feature of ubiquitous and pervasive computing systems. It is the key technology that enables intelligent interactions between users and IoT systems (Perera et al., 2014). Typically, context awareness describes devices that can sense their physical environment and change their behaviour accordingly (Elkhodr et al., 2013a).

The term context is defined by Perera et al. (2014) as "*any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*". For instance, location, identity, time, history events and activity are representing the primary context types for characterizing the situation of a particular entity (Perera et al., 2014).

2.4.1 Context Types and Categorization

Different researchers have identified context types differently based on their perspectives. Abowd et al. (1999) introduced one of the leading mechanisms of defining context types. They identified the location, identity, time, and activity as the primary context types. Further, they divided various types of context into two main categories: primary and secondary context.

- **Primary Context**: Any information retrieved without using existing context and without performing any kind of fusion operations on sensor data such as GPS sensor readings as location information.
- **Secondary Context***:* Any information that can be computed using the primary context. The secondary context can be computed by using sensor data fusion operations or data retrieval operations such as web service calls and identify the distance between two sensors by applying sensor data fusion operations on two raw GPS sensor values. Further, the retrieved context such as phone numbers, addresses, email addresses, birthdays, a list of friends from a contact information provider based on personal identity as the primary context can also be identified as secondary context.

In addition, Perera et al. (2014) introduced a context categorisation scheme, primary and secondary, that can be used to classify a given data value. They acknowledged location, identity, time, and activity as the most important context information. Figure 2.9 shows the primary and secondary categorization of location, identity, time and activity.

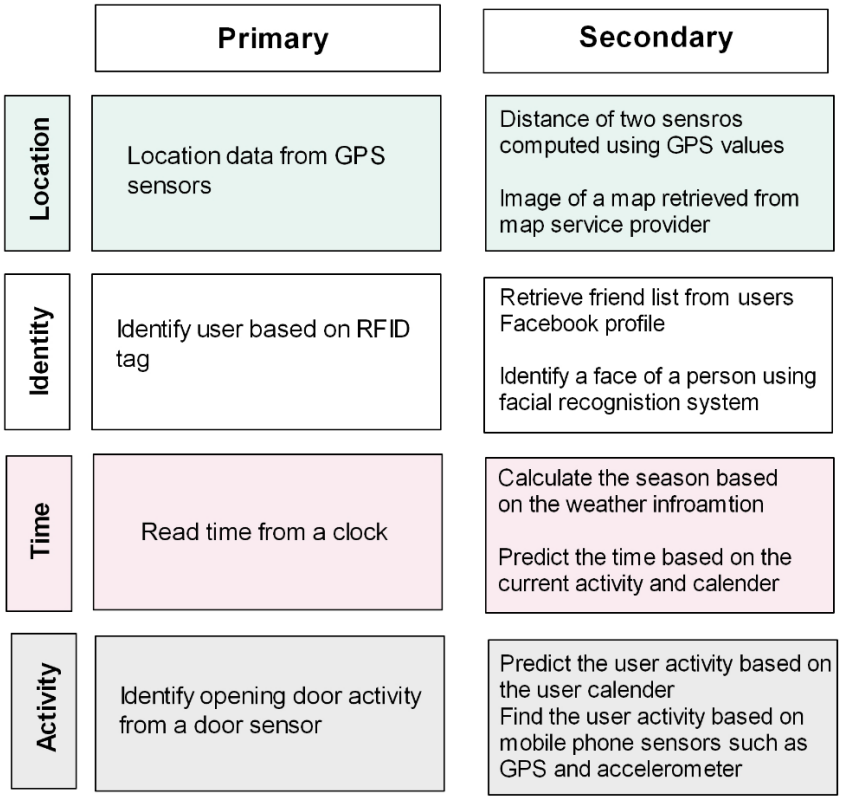


Figure 2.9: Categorization of most common context information (Perera et al., 2014)

2.4.2 Context-aware Features

There are common features of a context-aware environment. Abowd et al. (1999) identified three main features that a context-aware application can support: presentation, execution, and tagging.

- **Presentation**: Presenting the appropriate information regarding a certain context needs to be considered to define and decide what information and services need to be presented to the user. For instance, when a user enters a supermarket and takes their smartphone out. Context-aware mobile application should support the ability to connect to kitchen appliances such as a smart refrigerator in the home to retrieve the shopping list and present it to the user. This supports the idea of presenting information based on context like time, location, etc.

- **Execution**: The IoT system has the capability to use collected and analysed data to make an automatic decision based on context without human intervention. For example, in a smart home environment, when a user starts driving home from their work, the IoT application employed in the house might switch on the air condition system and switch on the coffee machine to be ready to use by the time the user steps into their house. These actions need to be taken automatically based on the context. M2M communication is also a significant part of the IoT that enable automation of IoT services using context information.

- **Tagging**: In the IoT system, there are billions of sensors linked to almost everyday things. These sensors generate a huge volume of data that need to be collected, analysed, and

interpreted (Wan & Alagar, 2013). Data generated by a single sensor will not be able to provide the required information that can be used to fully understand the situation. Instead, data collected through multiple sensors should be fused together. To perform sensor data fusion, context information needs to be tagged together with the sensor data to be processed and understood later. Hence, context tagging plays a significant role in context-aware computing.

2.5 Risk-based Access Control

The risk can be defined as the possibility of loss or injury. Generally, the risk is about some event that may occur in the future and cause losses. According to Elky (2006), the risk is the possible damage that may arise from the existing operation or from some upcoming incident. The risk is found in many aspects of our life and used in different disciplines. From the information technology security perspective, the security risk is defined as the damage that undesirably impacts an operation and its related information. While the process of understanding and mitigating against issues that may result in a breach of confidentiality, integrity or availability of an information system is called risk management (Elky, 2006).

Security risk in the context of access control can be defined as the possibility of information leakage and the value of this information that may occur from accessing system resources (Dos Santos et al., 2014). A risk-based access control model uses security risk as a criterion to make the access decision of an access request. This model permits or denies access requests dynamically based on the estimated risk value (Chen et al., 2007). This model performs risk analysis on each user access request to make the access decision. Mathematically, the most common formula to formalize the risk in quantitative form is (Dos Santos et al., 2014):

$$Quantified\ Risk = Likelihood \times Impact \tag{2.1}$$

Where likelihood represents the probability of an incident to happen, while impact represents the estimation of the value of the damage regarding that incident.

Risk-based access control models are divided into two types: Non-adaptive and adaptive. In the non-adaptive approach, a risk value is estimated for each access request. Then, the estimated risk value is compared against the risk-threshold value to determine the access decision whether granting or denying access. Whereas in the adaptive approach subsequent to granting access, there is an additional activity monitoring process in order to detect any abnormal behaviour during the access session. On the successful detection, the risk-threshold should be automatically lowered to stop certain risky operations. The user can be warned otherwise, access session can be terminated (Abie & Balasingham, 2012). The fundamental distinction between adaptive and non-adaptive is that the adaptive model requires a system monitoring process in which the risk threshold value is adaptively adjusted based on users' activities during access sessions. While the non-adaptive approach only

calculates the risk during the access session creation and does not have run-time monitoring or abnormality detection capability (Bijon et al., 2013).

There are several methods to build a risk-based access control model. These methods share some general characteristics from diverse models. The main elements of a risk-based access control model are shown in Figure 2.10.

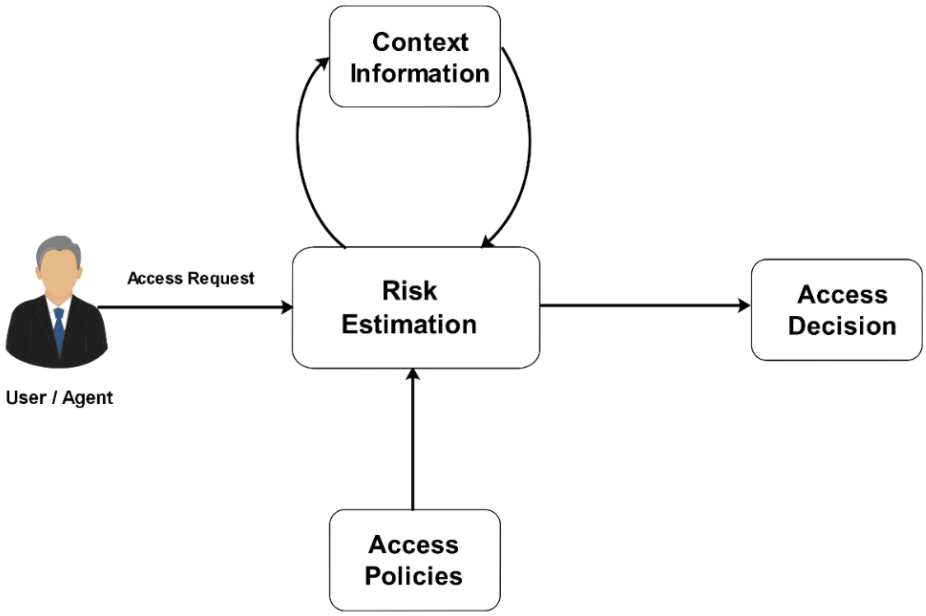


Figure 2.10: Main elements of a risk-based access control model

The risk-based access control model consists of three modules. The risk estimation is the main module. It receives requests from users, analyses them, collects context information and estimates the risk value associated with the access request. Then, the estimated risk value is compared against access policies to make the access decision (Diep et al., 2007).

Several approaches have been recently proposed to address the limitations of static access control models in terms of lack of flexibility, inability to handle contextual information and unexpected situations in managing access control operations. The next section provides a review of the literature regarding context-aware and risk-based access control models.

2.5.1 Context-aware Models

Building a flexible and fine-grained access control model is one of the most important aspects to provide efficient and effective control over access to system resources. This can be achieved by enabling context-aware access control models that not only use access policies but also contextual information to provide the access decision.

Context-aware access control models propose the use of contextual and environmental information to achieve fine-grained access control. Although these models do not evoke an explicit notion of access risk, the request's context and environment can provide relevant information that could be

used to assess the access risk. This research considers contextual information when evaluating the access request, but instead of statically including the contextual condition in the policy, contextual information is used as a risk factor to estimate the security risk value associated with the access request.

There are some context-aware models that extend the RBAC model with context attributes to provide a flexible access control model. Covington et al. (2000) have proposed a Generalized Role-Based Access Control (GRBAC) model. Their model extends the traditional RBAC by applying the roles to all the entities in the system (in RBAC, the role concept is only used for subjects). They defined three main roles; subject role, environment role, and object role. The GRBAC model uses context information as a factor to make access decisions. Also, Zhu and Xu (2008) have utilized context parameters in their dynamic RBAC model with two key ideas: 1) A user's access privileges must change when the user's context changes; 2) A resource must adjust its access permission when its system information (e.g., network bandwidth, CPU usage, memory usage) changes. However, the two papers do not consider security aspects in the decision-making process and the impact of security problems on the system. They also lacked adaptive control to prevent malicious attacks during access sessions.

Context-aware models are also introduced to provide a dynamic access control model for healthcare. Garcia-Morchon and Wehrle (2010) have proposed an access control model for prompt responses to emergency situations in medical environments. This model makes it possible to cope with rapidly changing situations by analysing them step by step in accordance with priorities and by establishing the appropriate policies and permissions for different situations. Also, Peleg et al. (2008) presented a framework for situation-based access control for privacy management using the object-process methodology to structure the scenarios and conceive a situation-based model. Their main objective was to preserve the patient's security and privacy. However, the two papers provided a qualitative method to provide the access decision, which is not applicable to provide a fine-grained access control model. Also, they did not provide an evaluation to prove the accuracy of their qualitative method.

In addition to context awareness, there are other works proposing using operational need to increase the flexibility of access control models. McGraw (2009) proposed a Risk-Adaptable Access Control (RAdAC) model which is based on estimating the security risk and operational needs to grant/deny the access. This model is implemented to first estimate the risk associated with the access request then compares the estimated risk with the access control policy. After that, the system verifies the operational needs, if the associated operational needs and the policy are met, then the access is granted. However, this model does not provide details about how to estimate risk and operational needs quantitatively. In addition, Kandala et al. (2011) utilized the RAdAC model to identify

different risk components with operational needs using their Attribute-Based Access Control (ABAC) model.

Other works proposed integrating trust with risk to provide the access decision. Baracaldo and Joshi (2012) have proposed a framework that extends the RBAC model to incorporate trust with risk to provide the access decision. They argued that their framework adapts to suspicious changes in users' behaviour by removing privileges when users' trust falls below a certain threshold. This threshold is computed based on a risk assessment process that includes the risk due to the inference of unauthorized information. Moreover, Burnett et al. (2014) have proposed trust and risk-aware access control that provide policy coverage and dynamic access decisions. They defined a zone policy model that allows the data owner to have total control over his own data. Trust is used to define verification of whether the requester respected the obligations that are assigned to him/her or not. They utilized a probabilistic computational trust model called subjective logic to formulate their trust assessment. Their risk estimation was done by using a classic method of defining expected loss in term of unwanted disclosure.

2.5.2 Risk-based Access Control Models

Risk-based access control models are used primarily to provide the required flexibility and fine-grained to the access control process. There is a set of risk-based models proposed to address the limitations of static access control models in terms of lack of flexibility and inability to handle unexpected situations.

The journey of implementing a risk-based access control model started when Jason (2004) suggested that there are three main elements to implement a risk-based model. These elements are estimating risk, identifying acceptance levels of risk, and controlling information sharing. This idea has been utilized by Diep et al. (2007) to build a dynamic and flexible risk-based model by collecting useful information from the environment, evaluates it from the security perspective, and make the access decision using a risk assessment. Similarly, Lee et al. (2007) have provided an access control model based on risk assessment and context. This model gathered useful information from the environment and evaluates it from the security perspective. Risk assessment with a MultiFactor Evaluation Process (MFEP) technique is applied to estimate the associated risk value using outcomes of actions in term of availability, confidentiality, and integrity. This model is tested to manage information access in a hospital. However, considering more risk factors from the access control environment will enhance system efficiency but it lacks adaptive features. In addition, Khambhammettu et al. (2013) have constructed a framework based on estimating object sensitivity, subject trustworthiness, and the difference between them using a risk assessment. However, neither model describes how to estimate the risk value for each situation of the environment. Besides, these models require a system administrator with broad experience to give a reasonable value for each input feature in the early

stage of the risk assessment process. Also, these models lack adaptive control to detect malicious users throughout access sessions.

The idea of the risk-based access control model was the same for a long time, but various researchers proposed different risk factors and estimation techniques to produce an efficient and effective model. For example, Bertino and Lobo (2010) used the same elements of the risk-based model proposed by Diep et al. (2007) but with the use of a fuzzy logic system to estimates the risk value associated with the access request. They showed that fuzzy inference is a good approach for estimating access security risks. However, both models neglected the past behaviour of users in the risk estimation process and lacked adaptive features as well. Similarly, Li et al. (2013) have utilized the fuzzy logic system to estimate the risk associated with the access of healthcare information. A risk metric is associated with data sensitivity, action severity, and risk history to determine the appropriate control of healthcare information access. However, this model does not indicate how to estimate the risk quantitatively. In addition, it requires prior knowledge about environment situations outcomes and there is no way to prevent malicious actions during the access session. Additionally, Chen et al. (2007) utilized the fuzzy logic approach to design a Multi-Level Security (MLS) risk-based model. This model measures the risk using the difference between object and subject security levels. So, if the difference is large, the risk value associated with the access will be high. The resultant output risk is represented as a binary number where 0 permits the access and 1 denies the access.

In addition, Wang and Jin (2011) have proposed a quantified risk-based access control model. The risk value is estimated based on the purpose of access to different data sensitivity levels. The risk estimation process is done by employing the concept of Shannon entropy from information theory. A prototype implementation and simulations on real-world medical history records were performed to demonstrate the effectiveness of their proposed approach. However, the purpose of access as a risk factor is not enough to estimate the risk value to make the correct access decision. It also lacked adaptive features and real-time user context features. Also, Rajbhandari and Snekkenes (2011) have presented a risk analysis approach based on preferences or values of benefit that subjects can provide, rather than subjective probability using the game theory. Moreover, a simple privacy scenario between a user and an online bookstore is introduced to provide an initial perception of the concept. However, using only benefits of the subject to determine the access decision is not enough to develop a flexible and scalable access control model. In addition, it lacked adaptive features and contextual features.

Other researchers have suggested mathematical functions to formulate an algorithm to measure security risks of access control operations. For example, Sharma et al. (2012) have presented a task-based access control model that estimates the risk value based on the action to be performed. The risk value is estimated in terms of different actions and corresponding outcomes. However, this model does not provide how to estimate the risk value quantitatively. In addition, it requires prior

knowledge about environment situations outcomes and there is no way to prevent malicious actions during the access session. Also, Namitha et al. (2015) implemented a risk-based access control model based on user features and estimate the risk value using a mathematical function. However, this model does not use any other input features such as resource sensitivity, action severity, and risk history. In addition, no risk prediction technique is involved, and it lacked adaptive features.

In addition, Dos Santos et al. (2014) proposed a risk-based access control model that employs the notion of quantifying risk metrics and aggregating them. This model depends on the idea of risk policies, which allow service providers and resource owners to define their own metrics, allowing greater flexibility to the access control system. Further, a prototype of this model is created using the risk metrics and quantification of Sharma et al. (2012). In addition, they extended the work to develop an ontology-based method to estimate the risk value according to the context and adjusting the weights of each risk metric considering the actual number of risk metrics (Dos Santos et al. , 2016). Although this approach provides greater flexibility by allowing the resource owners to define their own metrics, it requires a security administrator to ensure the minimum security of the system. In addition, it lacked adaptive features. Also, Shaikh et al. (2012) proposed a dynamic risk-based decision method. This method uses the user's past behaviour to identify good and bad users. It depends on granting reward and penalty points to users after the completion of transactions. However, the past user's behaviour (reward/penalty) values are not enough to provide a dynamic access control model and it lacked adaptive features as well.

Britton and Brown (2007) presented a quantification method for their RAdAC model. In their proposed model, 27 metrics divided into 6 categories which are evaluated for every access request and aggregated to achieve a measure of the total security risk. Their risk definition considers both probability and impact as high, medium or low. They employ a triangular probability distribution and a Monte Carlo simulation to find the probability of each event, which is then multiplied by a weight attributed by experts to each metric. This method is built for military applications, so some metrics are not suitable for a general IoT application. Also, Zhang et al. (2006) have suggested a benefit and risk-based access control approach. This approach uses security risk and system benefits to determine the access decision. It assigns a risk and benefit vector for each action. The access to perform a certain action is permitted only if the system benefits are higher than the risk value of the access request. The system creates an action graph to describe permitted actions and methods for users to access system resources. However, this approach uses static and predetermined action graph to determine the access. Also, it is very difficult to update the action state in the action graph and it lacked adaptive features.

In addition, Chen et al., (2016) presented a dynamic risk-based access control model for Cloud Computing. It combines the ABAC with the risk-trust assessment method. The model drives a threshold risk value from historical records to determine the access decision. It utilizes the concept of data stream processing to evaluate risk values. However, this model lacked adaptive features and

real-time features while determining the access decision. Also, Choi et al., (2015) presented a framework for a context-sensitive risk-based model for medical information systems. This framework categorizes information to calculate the risk value and apply the risk through treatment-based permission profiling and specifications. This framework provides the access decision based on the severity of the context and treatment. However, this model does not provide how to estimate the risk quantitatively. Also, the work is limited to medical information systems and lacked adaptive features. Moreover, Abomhara et al. (2018) proposed a work-based access control model that balances between collaboration and safeguarding sensitive patient information. It uses object security level and subject trust to provide the access decision. It decides the risk threshold based on situational conditions. However, this model does not provide how to estimate the risk value quantitatively or how to determine the risk threshold value in various situations. Also, this model is limited to medical information systems and lacked adaptive features. Table 2.1 provides a summary of the contribution and limitations of related risk-based access control models.

Table 2.1: Contribution and limitations of related risk-based access control models

Related model	Summary of Contribution	Limitations
Diep et al. (2007)	Built a dynamic and flexible risk-based model by collecting useful information from the environment and make the access decision using a risk assessment.	Limited risk factors, no risk prediction technique was used, lacked adaptive and real-time features.
Chen et al. (2007)	Built a fuzzy MLS to build a risk-based access control model by measuring the difference between object and subject security level.	The user past behaviour has not been used, lacked adaptive features and time overhead of the fuzzy inference system. Do not include real-time features in the risk estimation process.
Li et al. (2013)	They utilized the fuzzy logic approach to estimate the risk associated with data sensitivity, action severity, and risk history to determine the appropriate control of healthcare information access.	It requires prior knowledge about environment situations outcomes. Do not include real-time features in the risk estimation process and lacked adaptive features.
Bertino and Lobo (2010)	Built a risk-based access control model that uses the fuzzy logic system for the risk estimation process. They showed that the fuzzy inference approach is a good approach for estimating access security risks.	Time overhead of the fuzzy logic system is high, lacked adaptive features, and do not involve real-time features in the risk estimation process.
Khambhammettu et al. (2013)	Conducted a framework based on estimating object sensitivity, subject trustworthiness, and the difference between them using a risk assessment.	lacked adaptive features. No risk prediction was used. Do not involve real-time features in the risk estimation process.
Shaikh et al. (2012)	Proposed a dynamic risk model that utilizes user past behaviour to identify good and bad users. It depends on granting reward and penalty points to users after the completion of transactions.	Past user behaviour (reward/penalty) values are not enough to provide a dynamic access control model and it lacked adaptive features as well.

Table 2.1: Contribution and limitations of related risk-based access control models (Cont.)

Related model	Summary of Contribution	Limitations
Rajbhandari and Snekkenes (2011)	Presented a risk analysis approach based on preferences or values of benefit that subjects can provide, rather than subjective probability, using the game theory.	Limited risk factors as the subject benefit is not enough to generate flexible access. No risk prediction was used and lacked adaptive features.
Sharma et al. (2012)	Presented a task-based access control model that estimates the risk value based on the action to be performed. The risk value is estimated in terms of different actions and corresponding outcomes.	Does not provide how to estimate the risk quantitatively. It requires prior knowledge about environment situations outcomes and lacked adaptive features.
Lee et al. (2007)	Built a risk-based model by gathering all useful information from the environment and evaluate it from the security perspective using the MFEP technique.	No risk prediction technique has been used, limited risk factors, and lacked adaptive and real-time features.
Namitha et al. (2015)	Implemented a risk-based model based on user features to estimate the risk value using a mathematical function.	The model is static in nature. Only user risk factors are used and lacked adaptive features.
Dos Santos et al. (2014) and Dos Santos et al., (2016).	Proposed a risk-based model using risk policies by employing the notion of quantifying risk metrics and aggregating them. In addition, they presented an ontology-based method to estimate the risk value based on the context and adjusting the weights of each risk metric.	Lacked adaptive features and require a security administrator to ensure the minimum security of the system. It also did not provide how to identify risk metrics quantitatively.
Wang and Jin (2011)	Proposed a quantified risk-based access control model for health IT systems based on the purpose of the access to different data sensitivity levels.	Limited risk factors, no risk prediction technique and lacked adaptive and contextual features.
Britton and Brown (2007)	Presented a quantification method for the RAdAC model for military applications. They employed a triangular probability distribution and Monte Carlo simulation to estimate the risk value for each access request.	Lacked adaptive features, no risk prediction technique, do not use real-time and contextual information to make the access and it is not suitable for a general IoT application.
Zhang et al. (2006)	Presented a benefit and risk-based access control approach which uses security risk and system benefits to determine the access decision. It assigns a risk and benefit vector for each action.	Uses static and predetermined action graph to determine access which is very hard to be updated. Also, it lacked adaptive and contextual features.
Choi et al. (2015)	Proposed a context-sensitive risk-based framework for medical information systems that uses the severity of the actions and treatment to provide access decision.	Does not provide how to estimate the risk quantitatively. Also, the work is limited to medical information systems and lacked adaptive features.
Chen et al., (2016)	Presented a dynamic risk-based access control model for Cloud Computing by combining the ABAC with the risk-trust assessment method.	Lacked adaptive features and real-time context features while determining the access decision.
Abomhara et al., (2018)	Proposed a work-based access control model that balances between collaboration and safeguarding sensitive patient information. It uses object security level and subject trust to provide access decision.	Does not provide how to estimate the risk quantitatively or how to determine the risk threshold value in various situations. The model is limited to medical information system and lacked adaptive feature.

In conclusion, building a dynamic access control model for the IoT system is a fundamental priority as the IoT is dynamic and distributed system in nature. However, current risk-based access control models concentrate only on providing access decisions without providing any way to prevent abnormal data access from authorized users. In addition, they lacked real-time and contextual features, which can be extracted from the IoT environment easily to determine the access decision. Although the aforementioned risk approaches offered an important improvement in terms of flexibility compared to traditional systems, there is a need for more research in this area.

2.5.3 Risk Factors

One of the essential parts of a risk-based access control model is to choose the effective risk factors that determine access decisions efficiently. There are many risk factors that can be used to estimate the risk value associated with the access request to make the access decision in a dynamic and effective manner. In this section, an overview of different risk factors used in relation to risk-based access control models is provided. Common risk factors are as follows:

- **Subject Clearance*:*** It represents the subject security level acquired from the system administrator. The most common clearances in the military are Top Secret, Secret, Confidential, and no clearance. Different access permissions are granted according to the subject role in the organization. Each role is associated with certain permissions (McGraw, 2009). The higher the clearance granted, the lower the associated risk value.
- **Object Clearance**: It represents the object classification level. The access is granted to a certain object depending on the classification level. Depending on the subject role, access to a certain object classification level can be granted or denied.
- **Resource Sensitivity***:* It describes the sensitivity level of resources the user wants to access. Different sensitivity levels have different risk values. The higher the resource sensitivity, the higher the risk value if the access is granted to this resource (Li et al., 2013).
- **Action Severity**: It represents the cost of a certain action on a certain resource in terms of confidentiality, integrity, and availability. An action if occurs might lead to a great loss, but another does just a little. So, different actions have different consequences and so have different risk values.
- **Risk History***:* It represents user previous risk values on a certain resource. It can be used to detect the future behaviour of the user toward a certain resource.
- **Trust***:* It is similar to risk history. It represents the subject trust toward a certain resource. Trust is classified into two categories: identity trust and behavioural trust. Identity trust is concerned with verifying the authenticity of an entity and focuses on objective credentials. While behavioural trust deals with the entity's 'trustworthiness, which depends on certain

contexts (Luo et al., 2009). In risk-based access control models, only behavioural trust is used.

- **Education Level:** This risk factor is associated with the amount of security-related training or education the requester has received. Typically, the more security-related training the requester has received, the less likely that the requester is to commit a security violation. Therefore, the security risk would be lower. Conversely, if a requester has not received any security training, there is a higher possibility that a security violation could occur due to negligent action or inaction (Maw et al., 2012).

Different risk factors were utilized in the aforementioned risk-based access control models. Table 2.2 provides a summary of the risk factors used in related risk-based models.

Table 2.2: Security risk factors in different risk-based access control models

Related Model	Subject clearance	Object clearance	Resource Sensitivity	Action Severity	Risk History	Subject Trust
Chen et al. (2007)	✓	✓				
Li, Bai and Zaman (2013)			✓	✓	✓	
Bertino and Lobo (2010)	✓	✓				
Khambhammettu et al. (2013)			✓			✓
Shaikh et al. (2012)					✓	
Rajbhandari and Snekkenes (2011)	✓			✓		
Sharma et al. (2012)			✓	✓	✓	
Lee et al. (2007)				✓		
Namitha et al.(2015)	✓					
Wang and Jin (2011)			✓			
Britton and Brown (2007)	✓	✓			✓	✓
Zhang et al. (2006)				✓		
Chen et al., (2016)					✓	✓
Choi et al. (2015)				✓		
Abomhara et al., (2018)		✓				✓

2.6 Smart Contracts

The notion of smart contracts was first presented by Nick Szabo in 1994 (Szabo, 1994). However, it remained just an idea until the invention of the blockchain. Blockchain is a distributed and decentralized ledger of transactions used to manage a constantly increasing set of records. To store

a transaction in the ledger, the majority of participating nodes in the blockchain network should agree and record their consent. A set of transactions are grouped together and allocate a block in the ledger, which is chained of blocks. To link these blocks together, each block encompasses a timestamp and hash function to the previous block. The hash function validates the integrity and non-repudiation of the data inside the block. Moreover, to keep all participating nodes of the blockchain network updated, each user holds a copy of the original ledger and all nodes are synchronized and updated with newly change (Atlam & Wills, 2019).

Blockchain delivers a high level of transparency by sharing transaction details between all participants' nodes involved in those transactions. Using the blockchain technology, no need for a third party which improve business friendliness and guarantees a trusted workflow. Also, it eliminates the single point of failure which affects the entire system. Moreover, blockchain provides better security since it uses public key infrastructure that protects the system against malicious actions (Atlam & Wills, 2019; Sultan et al., 2018).

A smart contract is defined as an executable code that runs on the blockchain to facilitate, execute and enforce the terms of an agreement. The key objective of a smart contract is to execute the terms of an agreement automatically once the specified conditions are met. Thus, smart contracts require low transaction fees compared to the conventional centralized systems that need a trusted third party to enforce and execute the terms of the contracts (Alharby & Moorsel, 2017).

Typically, there are two types of smart contracts: deterministic and non-deterministic (Morabito, 2017). The major distinction between the two types of smart contracts is the availability of data within the blockchain for the smart contract to run. In other words, a deterministic smart contract does not need any data from an outside entity to run, in contrast to run a non-deterministic smart contract, data from an outside entity is needed. Figure 2.11 shows basic concepts of a smart contract.

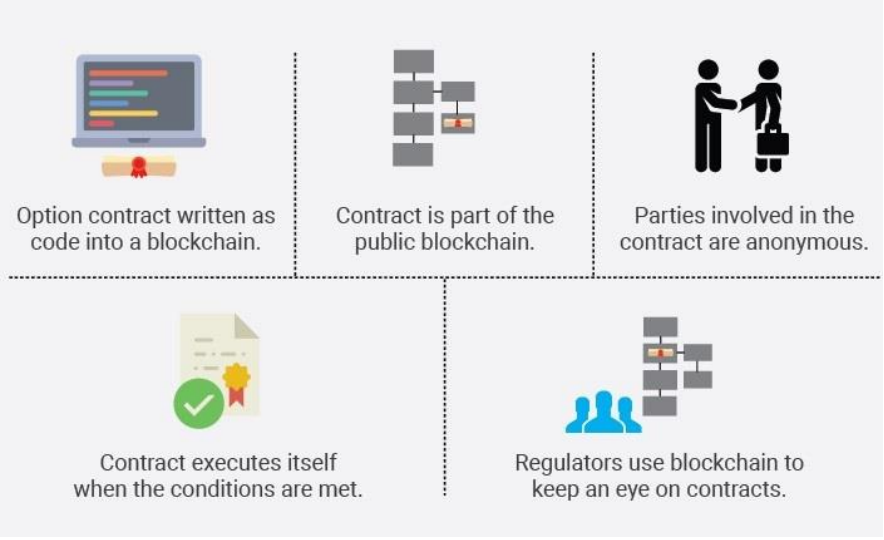


Figure 2.11: Basic concepts of smart contracts (Anand Narayan, 2017)

2.6.1 Structure of Smart Contracts

Smart contracts are nothing, but programming scripts stored on the blockchain. These scripts can be executed automatically when conditions or terms are verified or met. The blockchain technology provides an excellent environment for smart contracts to evolve. It eliminates the need for a trusted third party and secures its contents against any manipulation or attack (Mohanta et al. , 2018).

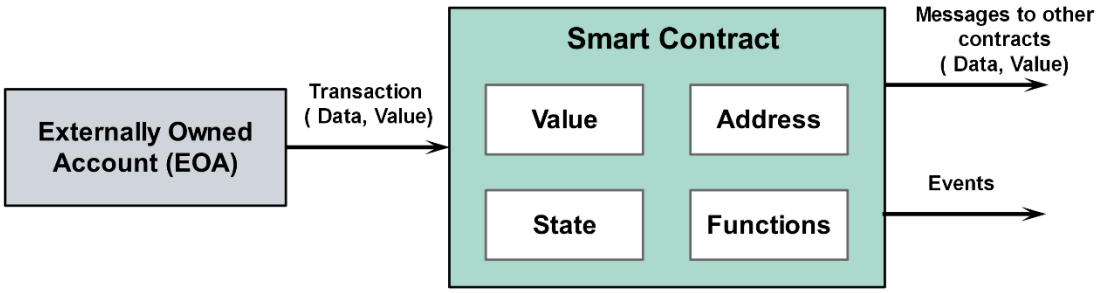


Figure 2.12: Basic structure of a smart contract (Bahga et al., 2016)

Typically, a smart contract consists of four major parts: address, value, functions, and state (Bahga et al., 2016), as depicted in Figure 2.12. Functions are used to represent conditions and terms of the contract. These functions are executed when transactions are made to the smart contract containing these functions. The address is used to identify various smart contracts in the blockchain in which each contract is assigned a unique address of 20 bytes. Once the contract is deployed into the blockchain, the contract code cannot be changed. To run a contract, users can simply send a transaction to the address of the smart contract. This transaction will then be executed by every consensus node (also called miners) in the network to reach a consensus on its output. The contract's state will then be updated accordingly. Then, the contract state will be uploaded to the blockchain network (Alharby & Moorsel, 2017).

As the smart contract is a software code, several programming languages can be utilized to implement it. However, the Solidity is the most common programming language for implementing smart contracts in various blockchain platforms. Solidity is a high-level language that can work with different blockchain platforms such as Ethereum, ErisDB, Bitcoin, and NXT (Mohanta et al., 2018).

2.6.2 Benefits of Smart Contracts

The capability to use computerized contracts that are stored in the blockchain provides multiple benefits over traditional contracts. These benefits involve:

- **Autonomy**: Smart contracts support automation in programming, so when a certain condition is verified, the actions are executed automatically. Although smart contracts can be built on centralized systems, the actions cannot be executed only if they approved by the central system, which can take a long time (Natarajan et al. , 2017). Since there is no third-party in the blockchain, actions are executed automatically in a very short time.

- **Redundancy**: Since each user or node participating in the blockchain network have its own copy of the ledger, data is duplicated many times over on the blockchain. Therefore, the possibility of losing data is zero (Desjardins, 2017).

- **Security**: Blockchain provides better security since it uses public key infrastructure that protects against attackers. The participating users of the blockchain network place their trust in the integrity and security features of the consensus mechanism. In addition, blockchain eliminates the single point of failure which affects the entire system (Sultan et al., 2018). Therefore, data of smart contracts are encrypted and secured against any tamper or manipulation.

- **Cost Reduction**: Blockchain is based on a shared ledger which shares its contents with the participating nodes in the network in which each participating user holds a copy of the original ledger without the need for a central authority. This reduces costs associated with distributing and maintaining the ledger. For smart contracts, this can save the costs of third-parties that are used mainly to maintain the trust between participating users in the agreement (Atlam & Wills, 2019).

- **Accuracy**: Typically, a smart contract is a software code that implements terms and conditions of the contract as programming conditions, when it verified, the corresponding actions are executed automatically. Therefore, building a correct software code to represent conditions of the contract will ensure nearly zero error.

- **Efficiency:** Blockchain reduces the efforts needed to do reconciliation and handle disputes manually. The existing systems with separate ledgers can lead to inconsistent master and transaction data resulting in faulty and duplicated data. Also, identifying and correcting this data will take a long time. By using the distributed and immutability features of the blockchain, smart contracts provide efficient solutions over conventional contracts (Atlam & Wills, 2019).

- **Transparency**: Smart contracts offers a high level of transparency by sharing transactions details between all participants' nodes involved in those transactions. Also, there is no need for a central authority which improves business friendliness and guarantees a trusted workflow (Atlam & Wills, 2019).

- **Trust**: Smart contracts provide complete trust in their execution. The autonomous, transparent and security features in the smart contract eliminate any possibility of manipulation or error. Moreover, smart contracts and its related data are encrypted on the shared ledger and all parties can access them.

2.6.3 How Do Smart Contracts Work?

Smart contracts are typically deployed on blockchain. According to Smart Contracts Alliance (2016), there are six stages to design and verify smart contracts within the blockchain environment, as shown in Figure 2.13.

1. **Identify Agreement:** Smart contracts involve multiple communication nodes, so this phase is used to identify the desired outcomes of the agreement which include business processes, asset swaps, transfer of rights and other tasks.
2. **Set Conditions:** Smart contracts could be initialized by the parties themselves or by satisfying certain conditions like financial market indices, natural disasters, or event via GPS location. In addition, temporal conditions could initiate smart contracts on holidays, birthdays and religious events.
3. **Code the Contract:** A smart contract is written as a computer program in a way that the arrangement will be automatically executed when the conditional parameters are met.
4. **Apply Encryption:** Encryption provides secure authentication and verification of messaging between the parties relating to the smart contract.
5. **Execution and Processing:** In a blockchain iteration, when consensus is reached on authentication and verification, the smart contract is written to a block. The code is then executed, and the outcomes are memorialized for compliance and verification.
6. **Network Updates:** After executing the smart contract, all nodes in the network update their ledgers to reflect the new state. Once the record is verified and posted to the blockchain, it cannot be changed.

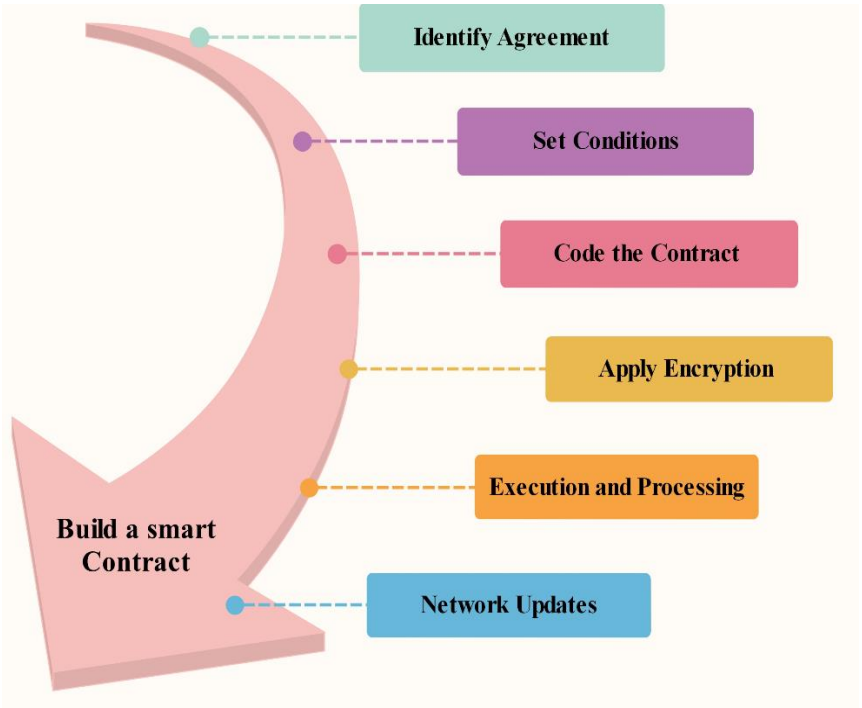


Figure 2.13: Major phases to build a smart contract

2.7 Summary

Chapter 2 presented the background and literature review of access control in the IoT. It started by providing an overview of the IoT and related security challenges. This was followed by providing a discussion of IoT security and access control models. Although the IoT system provides countless benefits, it brings several challenges, especially in security. Building an effective access control model can solve this issue, however, existing access control models are static and give the same outcome in different situations. Hence, dynamic access control models are more appropriate for the IoT as they use not only static policies but also real-time and contextual information to make the access decision. Risk-based access control model is one of the dynamic models that uses the security risk as a criterion to make the access decision. This model provides a flexible way to increase information sharing and at the same time ensures the security of information. Chapter 2 then presented an overview of context-awareness models. This was followed by providing a discussion of existing risk-based access control models by highlighting risk factors utilized in related risk-based models. Chapter 2 also provided an overview of smart contracts by highlighting the structure of smart contracts and how smart contracts work. The next chapter presents an overview of risk estimation techniques and introduces the fuzzy logic system with expert judgment to be the appropriate risk estimation approach.

Chapter 3: Risk Estimation Techniques for IoT

One of the significant stages to implement a risk-based access control model is the risk estimation process. Therefore, this chapter aims to provide an overview of risk estimation techniques that can be utilized to build a risk-based model for the IoT. This chapter starts by providing an overview of the risk estimation process with presenting advantages and disadvantages of both quantitative and qualitative risk estimation methods. Section 3.2 provides a discussion of various risk estimation techniques discussed in related risk-based access control models by highlighting their advantages and weaknesses. Then, section 3.3 introduces the fuzzy logic system with expert judgment to be the suitable risk estimation approach to implement the risk-based access control model for the IoT. Section 3.4 provides a detailed discussion of the main stages required to build a fuzzy logic system. This is followed by providing an overview of expert judgment and different phases needed to obtain an expert judgment in section 3.5. The chapter closes by providing a summary of the main points discussed through the chapter and introduces the next chapter.

3.1 Risk Estimation

Security risk is one of the main features used in access control models. It is the building block of risk-based access control approaches. Using the security risk as a criterion to provide the access decision can increase the security to an appropriate level with ensuring flexibility and increase opportunities of information sharing between different applications (Dos Santos et al., 2014).

The essential stage of implementing a risk-based access control model is the risk estimation process. This process is based on estimating the possibility of information leakage and the value of that information. The main objective of the risk estimation operation is to create a way of arranging risks in the order of importance and use risk numeric values to make access decisions in accordance with a specific context.

The security risk can be estimated either by qualitative or quantitative approaches (Yin et al., 2006). Quantitative risk estimation approaches are concerned with attaching specific numerical values to security risks. These values are used to determine access decisions directly. Although quantitative

risk estimation approaches are ideal as they lead to numeric values for the risk, it is difficult to perform without having a proper dataset describing risk likelihood and its impact on a specific application (Ramona, 2011).

Qualitative risk estimation approaches are used to calculate the risk early in the system. This is effective in categorising which risks should or should not be planned for and what is the appropriate action that should be taken for them. Qualitative risk estimation techniques cannot give accurate values for the risk. However, they are very powerful when we have little time to evaluate risks before they actually happen (Yin et al., 2006). Table 3.1 presents advantages and disadvantages of quantitative and qualitative risk estimation approaches.

Table 3.1: Advantages and disadvantages of quantitative and qualitative risk estimation methods (Yin et al., 2006 ; Ramona, 2011)

Approach	Quantitative Methods	Qualitative Methods
Advantages	• Risks are arranged by their cost • Objective methods are used to evaluate and estimate risk values • Availability, integrity and confidentiality are used to determine the security level • Best-suited measures are selected based on implementing a cost-analysis • With more experience, data accuracy will be increased	• Easy to understand • Easy to detect the risk level • Easy to implement • The risk analysis process is easier as the practical value of information is not used • Quantitative estimation of events probabilities and impact are not required • Estimated cost of the measure that should be implemented is not calculated
Disadvantages	• Calculation methods are complex • Very difficult to implement without an automatic tool • No standards for implementing this method • Need long time to handle the calculation process • The obtained results are introduced in the form of practical values which are hard to understand by the public without experience	• Risk calculation and its results are subjective • The subjective calculation is not enough to generate real and correct values • Because of their subjectivity, the performance of risk management is difficult to follow • A cost-benefit analysis is not implemented, only a subjective calculation • The accuracy of the results depends on the quality of the risk management team

3.2 Risk Estimation Techniques

Risk estimation process faces many challenges for various reasons. For instance, the goal of the risk estimation process is to predict the future possibility of information disclosure that results from the current access. Determining such a possibility is not an easy task (Habib and Leister, 2015). Moreover, if the risk estimation has relied on incomplete or imprecise information and knowledge about relevant risk features, it will result in difficulties to determine the value of the risk (Ni et al., 2010).

This section provides an overview of risk estimation techniques discussed in related risk-based access control models. Since the ultimate target of this research is to obtain a numeric value for the risk to determine the access decision whether granting or denying the access, only quantitative risk estimation techniques will be discussed.

3.2.1 Fuzzy Logic System

A fuzzy logic system is a computational approach which imitates how people think. It describes the world in imprecise terms such as if the temperature is hot, it responds with a precise action. Computers can work only on precise evaluations, while the human brain can provide reasoning with uncertainties and judgments (Bai & Wang, 1982). The fuzzy logic system is considered as a try to combine both techniques. Indeed, the fuzzy logic system is a precise problem-solving approach that has the ability to work with numerical data and linguistic knowledge simultaneously. It simplifies the management of complex systems without the need for its mathematical description (Kose, 2012).

The fuzzy logic system has many advantages. It is flexible, robust, and based on natural language which makes it easy to understand. It is also tolerant to imprecise data in which it can work even when there is a lack of rules. On the other hand, it faces some challenges. For instance, it needs domain experts to determine the fuzzy variables of the system. Also, it requires more tests and simulations which take a long time especially when there is a large number of rules (Shapiro & Koissi, 2015).

The computation process using the fuzzy logic system consists of three main phases:

- **Fuzzification** – The majority of variables are crisp or classical variables. Fuzzification process is used to convert crisp variables of input and output into fuzzy variables to process it and produce the desired output.
- **Fuzzy Inference Process** – Describing relationships between different inputs and output to drive the fuzzy output is done through building IF-THEN fuzzy rules. The fuzzy IF-THEN rule uses linguistic variables to describe the relationship between a certain condition and an output. The IF part is mainly used to represent the condition, and the THEN part is used to provide the output in a linguistic form. The IF-THEN rule is commonly used by the fuzzy logic system to represent how the input data matches the condition of a rule (Bai & Wang, 1982).
- **Defuzzification** – Since the output should be a crisp variable, this phase converts the fuzzy output back to the crisp output (Kose, 2012).

Some researchers utilized the fuzzy logic system to estimate the security risk in access control models. Chen et al. (2007) used the fuzzy logic system to build an MLS access control model to access information of IBM systems. This fuzzy MLS model estimates the risk value associated with the access request based on the difference between the subject security level and the object security

level. For instance, the larger the difference, the higher is the risk. Also, Li et al. (2013) presented a fuzzy modelling-based approach for evaluating the risk associated with the access request to healthcare information. They represented data sensitivity, action severity and risk history as a fuzzy value to determine the access decision. Moreover, Bertino and Lobo (2010) introduced a fuzzy inference technique to estimate the risk. Their fuzzy approach was used to estimate access risks and develop an enforcement mechanism for the risk-based model.

3.2.2 Expert Judgment

When there is insufficient practical data to describe the probability and impact of a certain incident, expert judgment can be used to provide a subjective evaluation based on experience. Expert judgment is commonly utilized to measure uncertain parameters in a probabilistic form to evaluate different elements of a certain model. Expert judgement can be defined as " *the expression of inferential opinions based on knowledge and experience*" (Leung & Verga, 2007).

Expert judgment is a powerful tool in risk analysis. It provides various solutions and decisions in several domains, such as psychology, criminal justice, financial forecasting, political science, and decision analysis. The use of expert judgement has raised many questions regarding the accuracy of the results. However, there are many circumstances where expert judgement is the only source of accurate information (Leung & Verga, 2007). Measuring the probability of an incident in risk analysis with the uncertainty that surrounds it is a difficult task especially for rare and extreme events. This is true when trying to estimate the security risks of access control operations (Turisová et al., 2012).

3.2.3 Risk Assessment

Risk assessment is used to avoid potential damages of a certain scenario. Risk assessment can be defined as the process of investigating possible losses using a combination of known information about the situation and judgment about the information that is not known (Shapiro & Koissi, 2015). The risk assessment is used to identify the risk context and acceptable risk values in each situation. This can be achieved by comparing it to similar risks of similar scenarios. In addition, it aims to provide substitute solutions to reduce the risk and calculate solutions effectiveness (Stoneburner et al., 2002).

Determining an appropriate type of risk analysis depends on the available data that characterize the risk probability and its impact. An effective risk assessment has many benefits. For example, a well-established risk assessment can support a balanced basis to prevent the risk or at least reduce its impact. However, it is a subjective process influenced by experience and it is only valid at a certain point in time (Stoneburner et al., 2002).

Risk assessment has been used in existing risk-based access control models. For instance, Khambhammettu et al. (2013) introduced three different approaches that conduct a risk assessment framework for a risk-based access control model. These approaches are based on the object sensitivity level, the subject trustworthiness level and the difference between them. Moreover, Diep et al. (2007) proposed a risk-based access control model based on risk assessment by using the outcomes of actions in terms of availability, confidentiality and integrity to estimate the risk value for each access request.

3.2.4 Game Theory

Game theory is considered as a division of applied mathematics that has been utilized in several domains like evolutionary biology, economics, artificial intelligence, political science, and information security. Game theory is used to describe multi-person decision scenarios in the form of games where each player selects appropriate actions that lead to the best possible payoff while expecting reasonable actions from opponent players (Binmore & Vulkan, 1999).

Game theory is the main tool for modelling and building automated decision-making operations in interactive environments. This is because it can provide consistent and mathematical platforms. The power of the game theory lies in the methodology it supports for analysing different problems of strategic choice. The process of modelling a condition as a game needs the decision-maker to interact with the players, their strategic decisions, and to observe their preferences and responses (Hamdi & Abie, 2014).

A game theory comprises of four components; the players, their strategies, payoffs and the information they have. The players are the essential part of the game, they are the decision-makers within the game. While the strategy is the plan that the player uses regarding the movement of the opposite player. So, it is critical for the players to select the suitable tactics. The payoff is the rewards of the players in the game. For each player, the payoff is affected by both their own actions and those of the other player (Rajbhandari & Snekkenes, 2011). In the game theory, the risk analysis is done by using user benefits rather than the probability. Moreover, game theory is recommended in conditions where no practical data is available (Hamdi & Abie, 2014). However, it is very complex especially with more than two players. It also leads to random outcomes when using mixed strategies.

Game theory has been utilized in risk-based access control models. For instance, Rajbhandari and Snekkenes (2011) presented a risk analysis approach based on preferences or values of benefit which the subjects can provide using the game theory.

3.2.5 Decision Tree

A decision tree is a common methodology for many operations in machine learning. It is used as a decision support instrument to provide decisions depending on a group of rules presented as a tree

(Shang & Hossen, 2013). Building a decision tree model requires dividing the data into training and validation sets. Training data are utilized to extract appropriate rules for the tree. While validating the tree and making the required modifications are done using the validation data.

The decision tree is represented as a flow diagram where each node, represented by a rectangle, describes the risk probability and its impact. These rectangles are connected by arrows in which each arrow leads to another box representing the percentage probability (Shang & Hossen, 2013).

Decision tree approaches are easy to comprehend. They can operate efficiently with inadequate data if experts provide all the required rules. They can show all possible alternatives and traces in a single view which provides a simple comparison with various alternatives. Whilst the decision tree model provides many advantages, it also has some limitations. For instance, its scalability is questionable such that when the scale of the tree increases, the obtained model will be hard to recognize, and it needs more supplementary data to validate the rules. Also, a decision tree model is based on expectations, so it may be impossible to plan for all contingencies that can arise as a result of a decision (Wang et al., 2016).

Selecting the appropriate risk estimation technique that fits with the requirements of the IoT environment is not an easy task. Table 3.2 provides a summary of advantages and disadvantages of previously discussed risk estimation techniques to help providing a clear picture of each risk estimation approach.

Table 3.2: Advantages and disadvantages of risk estimation techniques

Approach	Advantages	Disadvantages
Fuzzy Logic System	• Easy to understand, test and maintain • Flexible and based on natural language • Robust, it operates even when there is a lack of rules or wrong rules • Tolerant to imprecise data • Can be built on top of the judgment of experts • Ability to work with any set of input-output data using the Neuro-Fuzzy System (NFS) • Use rules that express imprecision of the real world	• Need more tests and simulation • Do not learn easily • Difficult to establish correct rules without using domain experts • Lack of precise mathematical model • Subjective • Time overhead especially when there are a large number of rules • Scalability seems to be questionable when there are a large number of rules
Expert Judgment	• Quick to produce • Requires little resources in terms of time and cost • Can be as accurate as other expensive methods • With experienced experts, accurate results are guaranteed	• Subjective • Not consistent • The estimate depends on the level of experts' experience • Risky and prone to error • Need a large number of knowledgeable experts
Decision Tree	• Consider a large range of consequences • Easy to understand when there are few decisions and outcomes • Results improved by numerical values on decisions • Fast to build and test • Works well with non-linear data • Shows all possible alternatives and traces each alternative in a single view, allowing for easy comparison among various alternatives	• Based on expectations, so it may be impossible to plan for all contingencies that can arise as a result of a decision • More complex and less accurate with large trees • Unstable, a small change in input data can cause large changes in the tree • Storage constraints, re-drawing decision trees manually require large spaces • Need advanced knowledge to create a large decision tree • Do not take into account the dynamic nature of the business
Risk Assessment	• An effective tool used in decision-making • Assess, communicate, organize the risks and expected benefits • Lead to optimal productivity • Enhance transparency	• Subjective process influenced by experience • Valid at a certain point in time, but maybe different later on • Not consistent • Time and cost overhead
Game Theory	• No actuarial data is needed • Risk analysis is based on outcomes, which the subjects can provide rather than subjective probability • Risk analysis is in the form of a game with players and strategies • Ideal for strategic situations with individual behaviour	• Complex and difficult with more than two players • Its application and assumptions are unrealistic • The use of mixed strategies generates random outcomes • Does not consider resource limitations • Assumes both players are smart and rational

In addition, the benefits and limitations of previously discussed risk estimation techniques are represented in a summarized form in Table 3.3.

Table 3.3. Benefits and limitations of risk estimation approaches

Risk Estimation Technique	Benefits					Limitations		
	Usable	Fast	Scalable	Dynamic	Include expert experience	Massive resources needed	Time overhead	Subjective
Fuzzy Logic	✓			✓	✓		✓	✓
Expert Judgment	✓	✓			✓			✓
Risk Assessment		✓	✓		✓		✓	✓
Game Theory	✓			✓		✓		✓
Decision Tree		✓		✓	✓	✓	✓	✓

It is clear that there is no straightforward approach that can be used without limitations. Also, a risk estimation approach without subjectivity will never exist in risk analysis. In addition, scalability seems to be a problem in most approaches. Therefore, choosing the optimal risk estimation approach should depend heavily on the context.

3.3 Proposed Risk Estimation Approach

There is no universal and best method for conducting risk analysis. However, it is important to understand strengths and weaknesses of various approaches to select the most appropriate approach to the context (Boc, 2012). There are many questions about the appropriate risk estimation technique to implement in a risk-based access control model for the IoT system. Understanding different advantages and disadvantages of previously discussed risk estimation approaches, as shown in Table 3.2 and Table 3.3, can provide a good indicator to select an appropriate risk estimation technique for the IoT context.

After investigating the literature regarding risk estimation techniques, the fuzzy logic approach with expert judgment was selected to be the suitable risk estimation technique to implement in a risk-based access control model for the IoT. There are many reasons for this selection. Firstly, there are significant sources of knowledge to provide all the required information to evaluate security risks regarding access control operations. One of the main sources is the past experience. Security administrators generally have some security skills regarding different risk factors and applications of suitable rules and policies regarding each context. This type of knowledge can be converted easily into rules for the fuzzy logic system (Alberts and Dorofee, 2002).

Secondly, one of the major problems in any research, especially in security, is the lack of datasets due to information protection laws. To correctly estimate the risk value associated with a specific

situation, the data describing the situation probability and its impact are required. Once data is available, it can be used to estimate a more precise risk value. Using a fuzzy logic system with expert judgment, there is no need for a dataset since the required data will be provided by the domain security experts. Expert judgment is a significant source of information in risk-based decision-making operations. This is because correct numerical data describing incident probabilities and its impact do not exist in most risk models (Tversky & Kahneman, 1974). In some cases, quantifying the value of the risk using classical approaches is very complicated, but with expert judgment, a more accurate value for a certain situation can be defined especially when appropriate experts are selected (Pluess et al., 2013).

Thirdly, the fuzzy logic system is flexible (Ruan, 2000), so, it will be suitable for the IoT system to adapt to its changing conditions and situations. Fourthly, although expert judgement adds subjectivity to the risk estimation process, the subjectivity can be reduced to an acceptable level in the fuzzy logic system, since subjectivity is moved to the process of creating rules which can be better controlled. Certainly, subjectivity is not completely eliminated. However, as depicted in Table 3.3, it is unlikely that a method with no subjectivity will ever exist for risk analysis (Boc, 2012). Finally, there are many successful applications that used the fuzzy logic system such as decision support, engineering, psychology, medicine, and home appliances (Zimmermann, 2000; Eldabi et al., 2002).

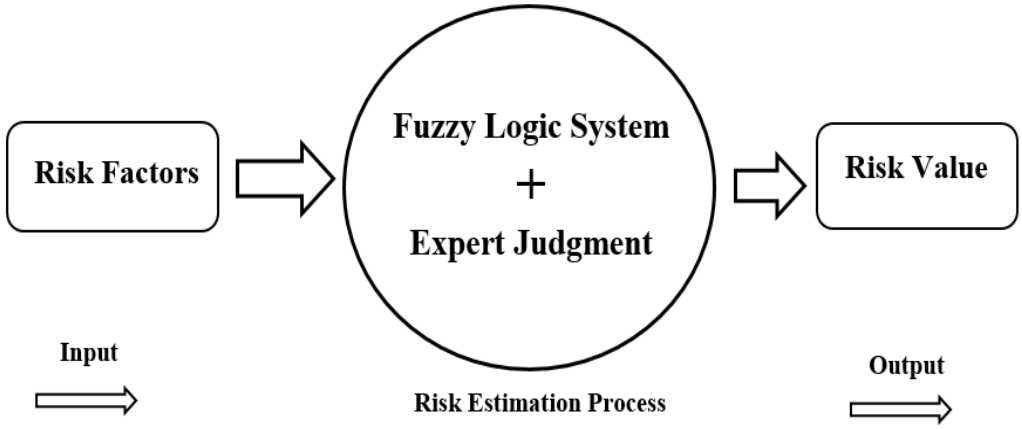


Figure 3.1: Combining the fuzzy logic system with expert judgment for the risk estimation process

Combining the fuzzy logic system with expert judgment could generate realistic risk values regarding certain scenarios. One of the essential steps to implement a fuzzy logic approach to estimate security risks is to set appropriate fuzzy rules. Determining appropriate fuzzy rules is one of the primary goals of combining the expertise of the domain experts with the fuzzy logic system.

The next section provides an overview of the fuzzy logic system by discussing the main stages of building a fuzzy logic system. This is followed by providing an overview of expert judgment and different phases needed to obtain an expert judgment.

3.4 Fuzzy logic System

The past few years have witnessed a rapid growth in the number and variety of fuzzy logic applications such as washing machines, camera autofocus, power supply regulation, aircraft engines, medical diagnosis systems, image processing and others. The fuzzy logic system has the ability to mimic how the human thinks. It employs modes of reasoning that are approximate rather than exact effectively.

Fuzzy Logic is a problem-solving methodology that can be implemented in hardware, software, or a combination of both. It provides a simple method to output a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. Moreover, the fuzzy logic system is based on the idea that all things in our environment are a matter of degrees. Temperature, height, speed, distance, beauty, etc., all can be defined with degrees (Rezaei et al., 2014).

Fuzzy or multi-valued logic was introduced in 1930 by Jan Lukasiewicz, a Polish philosopher (Keller et al., 2016). He introduced the logic that extended the range of truth values to all real numbers in the interval between 0 and 1, while classical logic works only with two values 1 (true) and 0 (false). The difference between classical or Boolean logic and multi-valued logic can be shown in Figure 3.2.

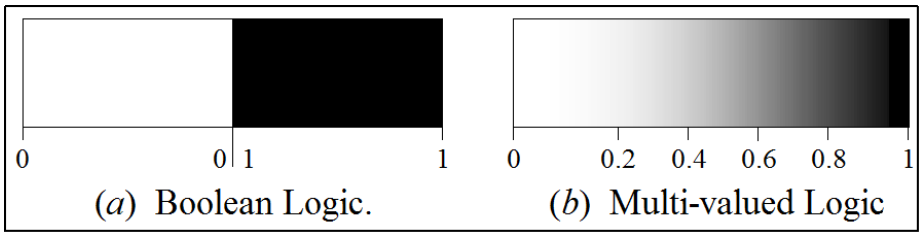


Figure 3.2: Difference between Boolean logic and Multi-valued logic (Keller et al., 2016)

Later in 1937, Max Black argued that a continuum implies degrees and published a paper called "*Vagueness: an exercise in logical analysis*" (Black, 1937). He said if a continuum is discrete, a number could be allocated to each element. He accepted vagueness as a matter of probability (Singhala et al., 2014). In 1965, Lotfi Zadeh published his paper "*Fuzzy sets".* Zadeh extended the work on the possibility theory into a formal system of mathematical logic and introduced a new concept for applying natural language terms (Zadeh, 1965). This new logic for representing and manipulating fuzzy terms was called fuzzy logic, and Zadeh became the master of the fuzzy logic (Boc, 2012; Singhala et al., 2014).

The fuzzy set theory provides a way to utilize imprecise and uncertain information made by a system and human judgments in a precise way. If the available data does not provide a suitable numeric result, the fuzzy logic system can resolve this problem by using linguistic expressions such as low, medium and high (Radionovs & Uzhga-rebrov, 2014).

Generally, implementing a fuzzy logic approach for an application requires five steps (Kose, 2012; Singhala et al., 2014), as depicted in Figure 3.3.

1. **Fuzzification**: It is the process that converts classical data or crisp data into fuzzy variables using linguistic expressions.
2. **Membership Function (MF)**: It involves mapping each variable to a value between 0 and 1. This value is called membership value or degree of membership.
3. **Fuzzy Inference Rules**: It represents the relationship between input and output linguistic expressions using IF-THEN rules to derive the output.
4. **Rule Aggregation**: It combines fuzzy sets that represent the output of each rule into a single fuzzy set.
5. **Defuzzification**: It is the process that converts the fuzzy output back to the crisp or classical output. The fuzzy output is still a linguistic variable, and this linguistic variable needs to be converted to the crisp variable through the defuzzification process.

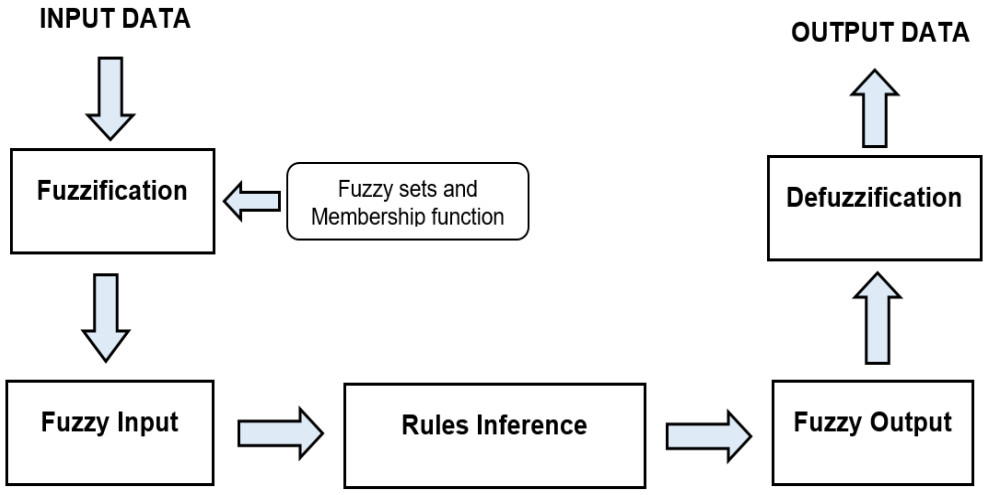


Figure 3.3: Representation of the fuzzy logic approach (Kose, 2012)

3.4.1 Fuzzification

The first step to apply the fuzzy logic approach is fuzzification. Most variables existing in the real world are crisp or classical variables. One needs to convert these crisp variables (both input and output) into fuzzy variables, and then apply fuzzy inference to process these data to obtain the desired output (Singhala et al., 2014). The input and output linguistic variables are divided into fuzzy sets. A fuzzy set is a set containing elements with varying degrees of membership to this set. The idea of fuzzy sets is opposite to classical or crisp sets because members of a crisp set would not be members unless their membership is full or complete in that set. While in a fuzzy set, elements' membership need not to be complete and they can also be members of other fuzzy sets on the same universe (Kose, 2012; Ross, 2010).

The difference between classical logic and fuzzy logic can be shown in Figure 3.4. A sentence in the classical logic universe can have only two possible values; small or large. While with the fuzzy logic theory, the sentence may have a large (maybe infinite) number of values. Therefore, fuzzy sets solve the problem of vague linguistic terms (Korol & Korodi, 2011).

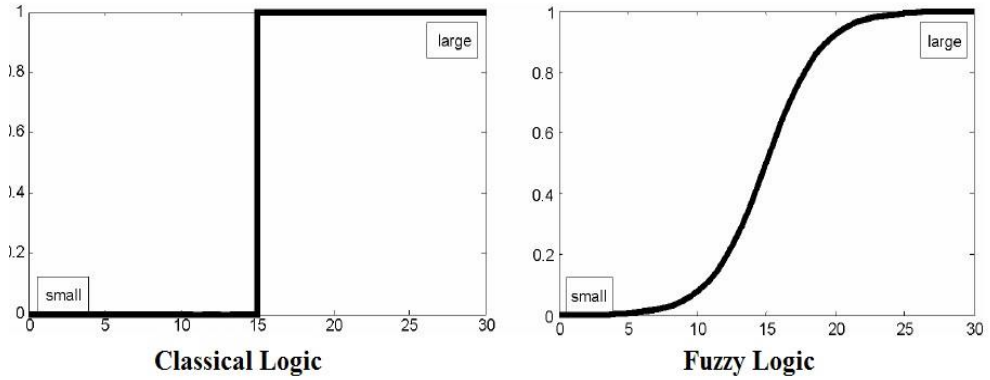


Figure 3.4: The difference between crisp sets and fuzzy sets (Korol & Korodi, 2011)

3.4.2 Membership Functions

Generally, fuzzification involves two processes: derive the MFs for input and output variables and represent them with suitable linguistic variables (Shapiro & Koissi, 2015). Linguistic variables are the building blocks of the fuzzy logic system. They are defined as variables whose values are expressed as words or sentences (Zadeh, 1965). For instance, linguistic variables associated with temperature can be set as cold, hot and very hot.

Fuzzy MF is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes called the universe of discourse. The only condition MF must satisfy is that it must vary between 0 and 1 (Ross, 2010). MFs can have different types such as triangular, trapezoidal, Gaussian, and others. Selecting the appropriate MF depends on the actual application. For those systems that need significant dynamic variation in a short period of time, a triangular or trapezoidal waveform should be utilized. For those systems that need very high control accuracy, a Gaussian or S-curve waveform should be selected (Bai & Wang, 1982; Kose, 2012).

In practice, MF for a fuzzy set "A" on the universe of discourse "X" is defined as $\mu A:X \rightarrow [0,1]$, where each element of X is mapped to a value between 0 and 1. This value, called membership value or degree of membership, quantifies the grade of membership of the element in "X" to the fuzzy set "A". MFs allow us to graphically represent a fuzzy set. The x-axis represents the universe of discourse, whereas the y-axis represents the degrees of membership in the [0,1] interval (Wang, 2015).

The next section provides a brief discussion on various MFs that can be implemented using MATLAB by providing the equation and representation of each MF.

3.4.2.1 Triangular MF

Triangular MF (Trimf) is a triangular-shaped function that used to represent the relationship between fuzzification inputs and fuzzified output. It is represented by a lower limit "a", an upper limit "b", and a value "m", where a < m < b (Wang, 2015). The triangular function and its representation are shown in Figure 3.5.

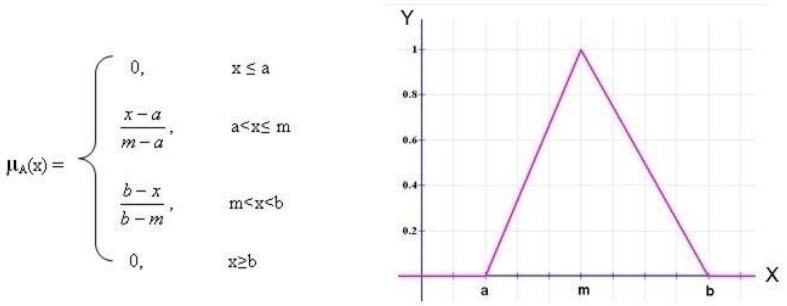


Figure 3.5: Function and representation of triangular MF (Mathworks, 2016)

There are two types of triangular MF; symmetric and asymmetric. The only difference between symmetric and asymmetric MF, as shown in Figure 3.6, is the value of m which divides the MF into two equal halves in the symmetric MF.

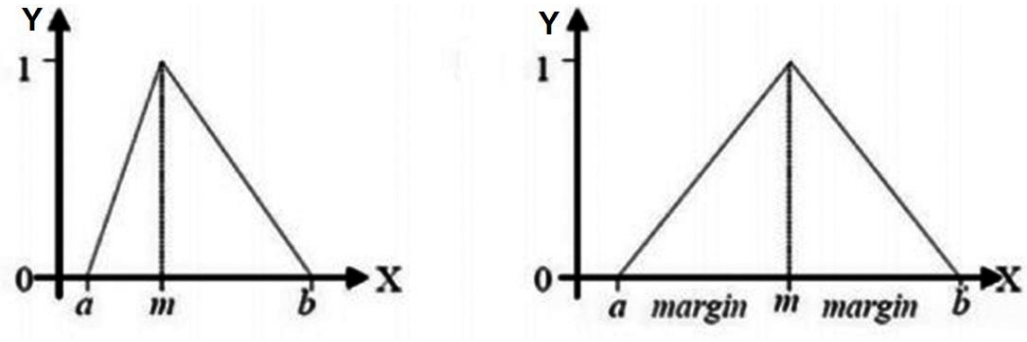


Figure 3.6: Difference between asymmetric and symmetric triangular MF (Mathworks, 2016).

3.4.2.2 Trapezoidal MF

Trapezoidal MF (Trapmf) is a trapezoidal-shaped function that is represented by a lower limit "a", an upper limit "d", a lower support limit "b", and an upper support limit "c", where a < b < c < d (Wang, 2015). Trapezoidal function and its representation are shown in Figure 3.7.

$$\mu_A(x) = \begin{cases} 0, & (x < a) \text{ or } (x > d) \\ \frac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \frac{d-x}{d-c}, & c \le x \le d \end{cases}$$

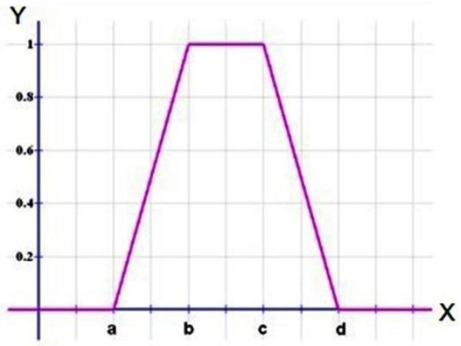


Figure 3.7: Function and representation of trapezoidal MF (Mathworks, 2016)

3.4.2.3 Gaussian MF

Gaussian MF (Gaussmf) is a gaussian curve function that is represented by a central/mean value "m" and a standard deviation k > 0. The smaller "k" is, the narrower the "bell" is (Mathworks, 2016). Gaussian function, where "k" and "m" represent the standard deviation and the mean respectively, and its representation are shown in Figure 3.8.

$$\mu_A(x) = e^{-\frac{(x-m)^2}{2k^2}}$$

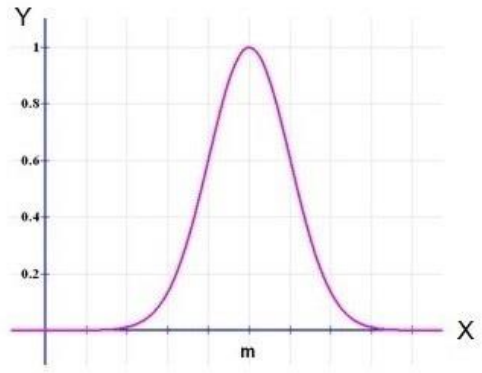


Figure 3.8: Function and representation of gaussian MF (Mathworks, 2016)

3.4.2.4 Gaussian2 MF

Gaussian2 MF (Gauss2mf) is a gaussian combination function that used to represent the relationship between fuzzification inputs and fuzzified output. The normal gaussian MF is represented using two parameters (sig, c), while gaussian2 MF is represented using a combination of two of these two parameters. The first function, specified by sig1 and c1, determines the shape of the left-most curve while the second function specified by sig2 and c2 determines the shape of the right-most curve. Whenever c1 < c2, the gaussian2 MF function reaches a maximum value of 1 (Mathworks, 2016). In Gaussian2 MF σ represents the standard deviation and "c" represents the mean, and its representation can be shown in Figure 3.9.

$$f(x;\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}}$$

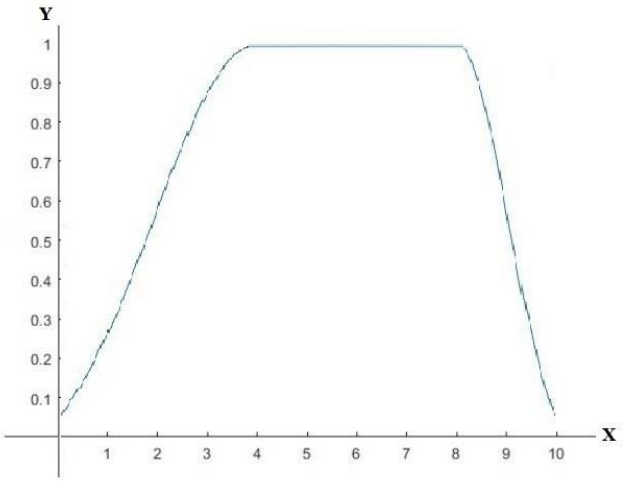


Figure 3.9: Function and representation of gaussian2 MF (Mathworks, 2016)

3.4.2.5 Generalized bell-shaped MF

Generalized bell-shaped MF (Gbellmf) depends on three parameters "a", "b", and "c" where the parameter b is usually positive. The parameter c locates the centre of the curve (Mathworks, 2016). Generalized bell-shaped MF and its representation can be shown in Figure 3.10.

$$f(x;a,b,c) = \frac{1}{1+\left|\frac{x-c}{a}\right|^{2b}}$$

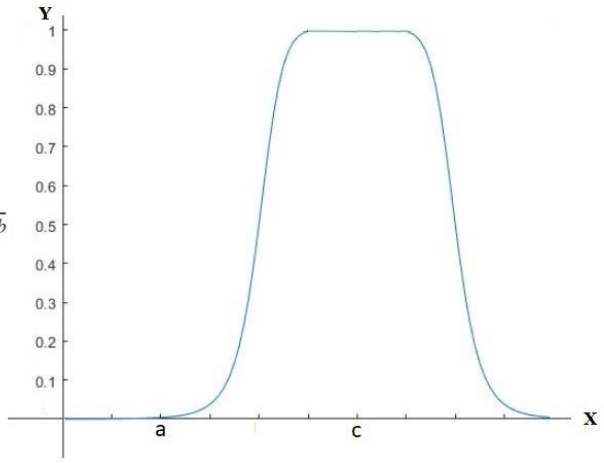


Figure 3.10: Function and representation of generalized bell-shaped MF (Mathworks, 2016)

3.4.2.6 Sigmoid MF

Sigmoid MF (Sigmf) is a sigmoid shaped function that depends on the parameter "a". Generally, the sigmoidal MF is inherently open to the right or to the left, and thus it is appropriate to represent concepts such as "very large" or "very negative" (Mathworks, 2016). Sigmoid MF function and its representation are shown in Figure 3.11.

$$f(x,a,c) = \frac{1}{1+e^{-a(x-c)}}$$

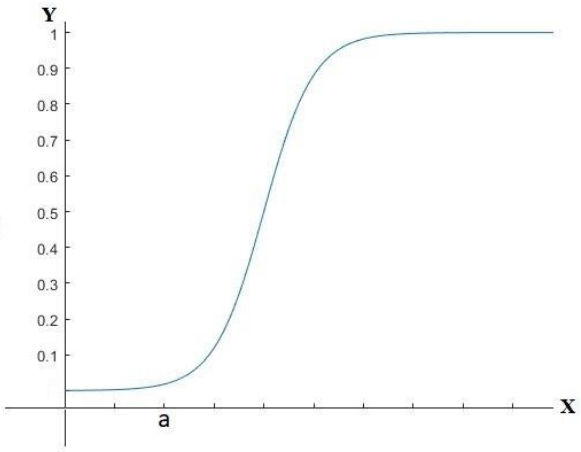
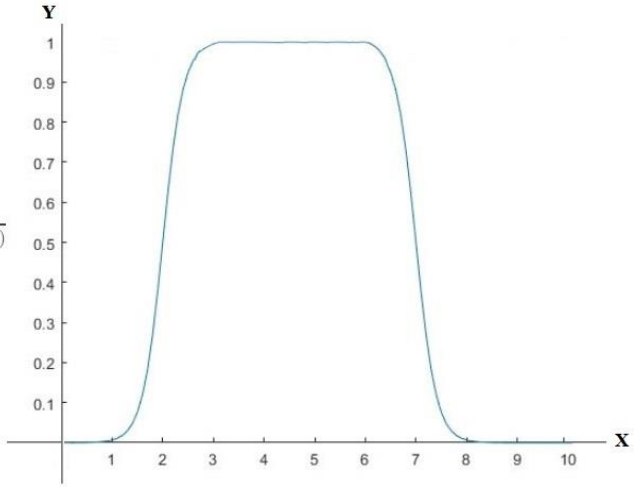


Figure 3.11: Function and representation of sigmoid MF (Mathworks, 2016)

3.4.2.7 Difference between two sigmoidal MF

Dsigmf is a function that used to represent the difference between two sigmoidal functions. It depends on four parameters a1, c1, a2, and c2. It is the difference between two of these sigmoidal functions "F1(x; a1, c1) - F2(x; a2, c2)" (Mathworks, 2016). Dsigmf function and its representation are shown in Figure 3.12.

$$f(x;a,c) = \frac{1}{1+e^{-a(x-c)}}$$



.

Figure 3.12: Function and representation of Dsigmf (Mathworks, 2016)

3.4.2.8 Product of two sigmoidal MF

Psigmf is a function that is used to represent the product of two sigmoidal MFs. Similar to Dsigmf, it depends on four parameters a1, c1, a2, and c2, and it is the product of two of these sigmoidal functions such that F1(x; a1, c1) × F2(x; a2, c2). Psigmf function and its representation are shown in Figure 3.13.

$$f(x;a,c)=\frac{1}{1+e^{-a(x-c)}}$$

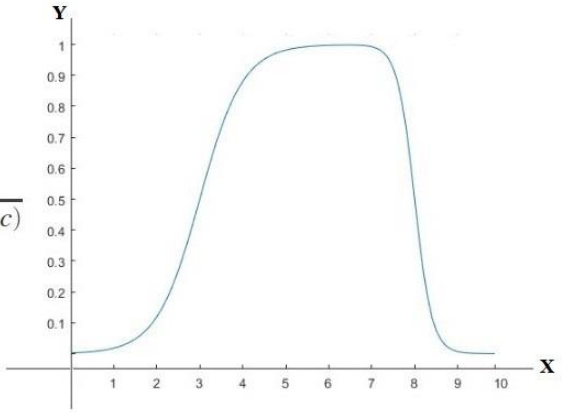


Figure 3.13: Function and representation of Psigmf (Mathworks, 2016)

3.4.2.9 S-shaped MF

S-shaped MF (smf) is an s-shaped function that represents a mapping on the vector x. It depends on two parameters a and b which locate the extremes of the sloped portion of the curve. S-shaped MF function and its representation are shown in Figure 3.14.

$$f(x;a,b)=\begin{cases}0, & x\le a\\ 2\left(\frac{x-a}{b-a}\right)^2, & a\le x\le\frac{a+b}{2}\\ 1-2\left(\frac{x-b}{b-a}\right)^2, & \frac{a+b}{2}\le x\le b\\ 1, & x\ge b\end{cases}$$

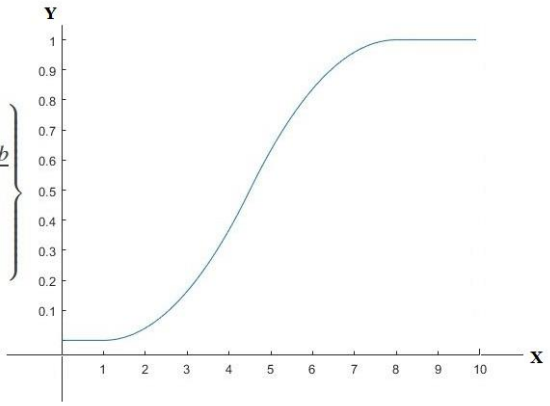


Figure 3.14: Function and representation of S-shaped MF (Mathworks, 2016)

3.4.2.10 Z-shaped MF

Z-shaped MF (zmf) is a z-shaped function that is used to represent the relationship between fuzzification inputs and fuzzified output. This spline-based function of x is so named because of its Z-shape. The parameters "a" and "b" locate the extremes of the sloped portion of the curve. Z-shaped MF function and its representation are shown in Figure 3.15.

$$f(x;a,b)=\begin{cases}1, & x\le a\\ 1-2\left(\frac{x-a}{b-a}\right)^2, & a\le x\le\frac{a+b}{2}\\ 2\left(\frac{x-b}{b-a}\right)^2, & \frac{a+b}{2}\le x\le b\\ 0, & x\ge b\end{cases}$$

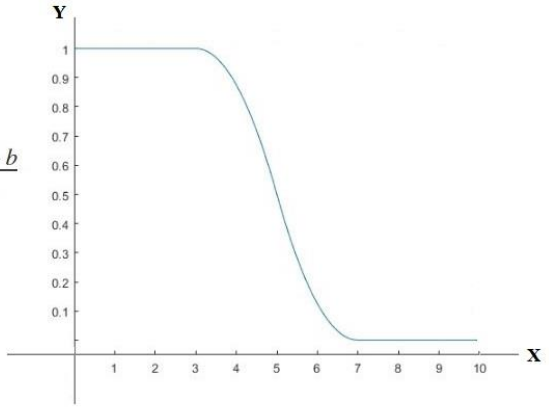


Figure 3.15: Function and representation of Z-shaped MF (Mathworks, 2016)

3.4.2.11 Pi-shaped MF

This spline-based curve is so named because of its Π shape. This MF is evaluated at the points determined by the vector x. Pi-shaped MF is a product of S-shaped MF and Z-shaped MF. Pi-shaped MF function and its representation are shown in Figure 3.16.

$$f(x;a,b,c,d)=\begin{cases}0, & x\le a\\ 2\left(\frac{x-a}{b-a}\right)^2, & a\le x\le\frac{a+b}{2}\\ 1-2\left(\frac{x-b}{b-a}\right)^2, & \frac{a+b}{2}\le x\le b\\ 1, & b\le x\le c\\ 1-2\left(\frac{x-c}{d-c}\right)^2, & c\le x\le\frac{c+d}{2}\\ 2\left(\frac{x-d}{d-c}\right)^2, & \frac{c+d}{2}\le x\le d\\ 0, & x\ge d\end{cases}$$

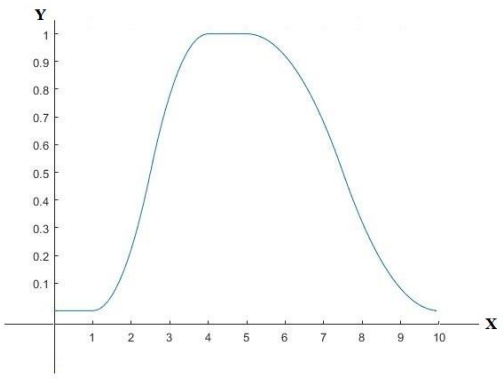


Figure 3.16: Function and representation of Pi-shaped MF (Mathworks, 2016)

3.4.3 Fuzzy Inference Rules

Fuzzy inference rules are considered as the knowledge base that describes the relationship between input and output linguistic expressions. They are represented by a sequence of IF-THEN statements, leading to a set of procedures that define what actions or outputs should be taken in terms of currently observed input combinations. Fuzzy rules are constructed based on knowledge or experience, which is dependent on each application (Bai & Wang, 1982).

The fuzzy IF-THEN rule uses linguistic variables to describe the relationship between a certain condition and an output or a conclusion. The IF part is mainly used to represent the condition, and the THEN part is used to provide the conclusion or output in a linguistic variable form. This IF-THEN

rule is commonly used by the fuzzy logic approach to represent the degree to which the input data match the condition of a rule (Bai & Wang, 1982). A fuzzy rule is represented in the following form:

IF X is A, THEN Y is B (3.1)

Where X and Y are linguistic variables and A and B are linguistic values determined by fuzzy sets on the universe of discourses X and Y respectively.

To build fuzzy inference rules, the type of Fuzzy Inference System (FIS) should be defined. There are two types of FISs: Mamdani and Sugeno. Mamdani FIS is the most commonly used fuzzy inference technique. This approach is introduced by Ebrahim Mamdani in 1975 to control a steam engine and boiler combination (Mamdani & Assilian, 1975). Mamdani FIS applied a set of fuzzy rules supplied by experienced human operators. Mamdani FIS is performed in four steps: fuzzification of the input variables, rule evaluation, aggregation of the rule outputs, and finally defuzzification (Negnevitsky, 2010).

On the other hand, Sugeno FIS was first introduced by Michio Sugeno in 1985 (Takagi & Sugeno, 1985). It is very similar to the Mamdani method. Instead of the fuzzy set; Sugeno used a mathematical function to represent the output variable. Sugeno FIS is based on generating fuzzy rules through a given input-output dataset (Negnevitsky, 2010). The main difference between the two methods lies in the consequent of the fuzzy rules. The format of the Sugeno fuzzy rule is represented as follows:

IF X is A, AND Y is B, THEN Z is F (X, Y) (3.2)

Where X, Y and Z are linguistic variables; A and B are linguistic values determined by fuzzy sets on the universe of discourses X and Y respectively, and F (X, Y) is the mathematical function of the output variable.

One of the important facts to notice about the two FISs is that Sugeno FIS cannot be used unless a given input-output dataset exists. Since most risk estimation processes suffer from the lack of appropriate datasets that characterise the risk, Mamdani FIS is the most commonly adopted fuzzy inference technique in risk estimation operations.

3.4.4 Rule Aggregation

Rule aggregation is the process used to combine fuzzy sets that represent the output of each rule into a single fuzzy set. It is used only once for each output variable, prior to the final step; defuzzification. Rule aggregation is one of the main stages in Mamdani FIS. This is because the fuzzy output depends on the evaluation of all fuzzy rules in the FIS; and hence, all rules should be combined to provide the fuzzy output.

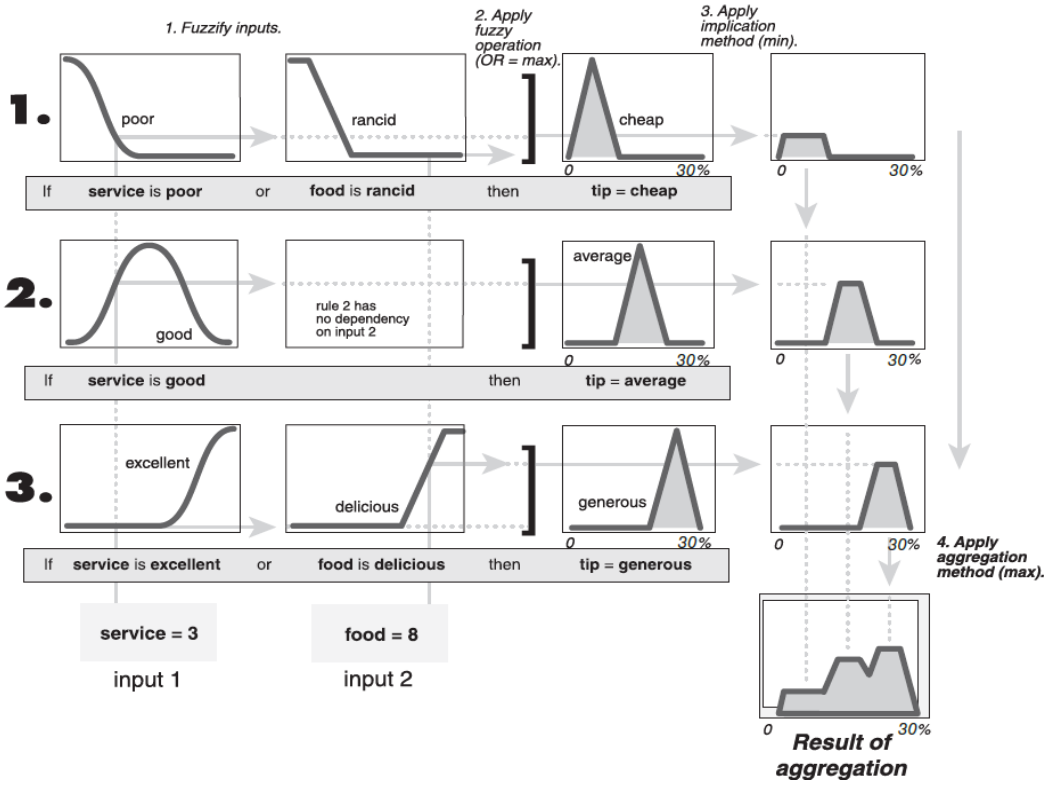


Figure 3.17: Fuzzy rule aggregation with the max method (Mathworks, 2016)

The input of the aggregation process is a list of truncated output functions returned by the inference process for each rule. The output of the aggregation process is one fuzzy set for each output variable (Mathworks, 2016). Most aggregation processes are commutative so, the order of the rule execution is not essential. Figure 3.17 shows the aggregation process by presenting how three fuzzy rules are combined and how the output of each rule is aggregated into a single fuzzy set using the max method.

3.4.5 Defuzzification

The last step for building a fuzzy logic approach is defuzzification. Basically, defuzzification is the process of mapping from a space of a fuzzy logic defined over an output universe of discourse into a space of a crisp logic. In other words, the defuzzification process is used to convert the fuzzy output back to the crisp or classical output. The fuzzy output is still a linguistic variable, and this linguistic variable needs to be converted to a crisp variable (Kose, 2012; Singhala et al., 2014).

There are five common defuzzification methods that can be implemented using MATLAB. These methods include:

1. **Mean of Maximum Method (MOM):** It works by calculating the average of fuzzy outputs that have the highest degrees. This method does not work with the entire shape of the output MF; instead, it only works with points that have the highest degrees in that function. For those MFs that have different shapes but the same highest degrees, this method will provide the same result (Kose, 2012; Ross, 2010).

2. **Centre of Gravity (COG):** It is the most common defuzzification method that is widely used in several applications. It is also called the centroid method. This method is similar to the principle of calculating the centre of gravity in physics. The weighted average of the MF or the centre of the gravity of the area bounded by the MF curve is computed to be the most crisp value of the fuzzy quantity (Kose, 2012; Ross, 2010).

3. **Bisector of Area (BOA):** The bisector is the vertical line that divides the region into two sub-regions of equal area. It is sometimes, but not always coincident with the centroid line (Kose, 2012).

4. **Smallest of Maximum (SOM):** It determines the smallest of the maximum value of the area under the curve of the aggregated MFs (Tóth-laufer & Takács, 2012).

5. **Largest of Maximum (LOM):** It determines the largest of the maximum value of the area under the curve of the aggregated MFs (Tóth-laufer & Takács, 2012).

Figure 3.18 shows an example that uses x = -10:0.1:10, and trapezoidal MF to show different ways to calculate the defuzzified output for each method. There is no superior method, however, the centroid method is more common and recommended to start with in the absence of a dataset (Mathworks, 2016).

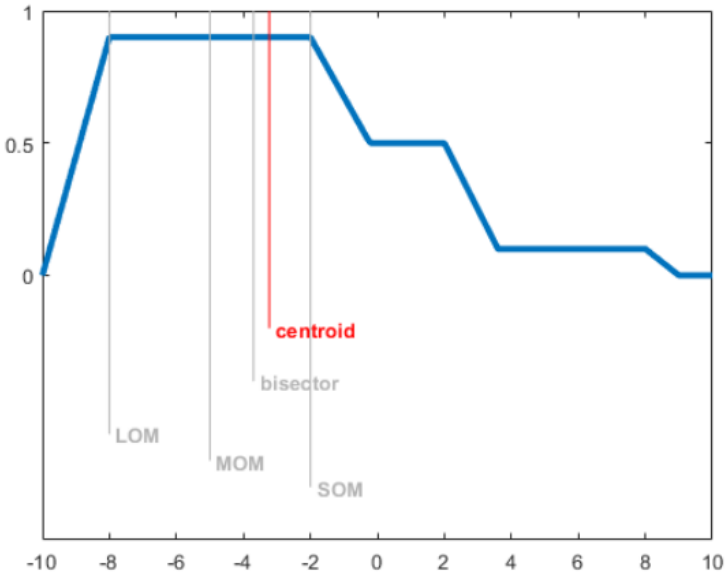


Figure 3.18: An example to show the calculation of defuzzified value using centroid, bisector, LOM, MOM, and SOM defuzzification methods (Mathworks, 2016)

3.4.6 Applications of Fuzzy Logic System

There are numerous applications that utilized the fuzzy logic system in various aspects of industrial production or manufacture. The fuzzy logic system is an effective tool to help decision-making in

manufacturing reengineering, optimize the process parameters for drilling processes, realize a better batch process scheduling, and others (Singh et al., 2013).

There are many applications for the fuzzy logic system. These are:

- **Aerospace** (e.g. altitude control of spacecraft and satellite altitude control)
- **Automotive** (e.g. intelligent highway systems, traffic control, and shift scheduling for automatic transmission)
- **Business** (e.g. decision-making support systems and personnel evaluation in a large company)
- **Defence** (e.g. underwater target recognition, control of a hypervelocity interceptor, and naval decision support aids)
- **Electronics** (e.g. washing machine timing, microwave ovens, vacuum cleaners, and air conditioning systems)
- **Finance** (e.g. stock market predictions, fund management, and banknote transfer control)
- **Marine** (e.g. autopilot for ships, optimal route selection, and ship steering)
- **Healthcare** (e.g. medical diagnostic support system, multivariable control of anaesthesia, and radiology diagnoses)
- **Security** (e.g. decision systems for security trading and various security appliances)
- **Transportation** (e.g. train schedule control, railway acceleration, automatic underground train operation, and braking and stopping)
- **Image processing** (e.g. pattern recognition and classification, handwriting recognition, and command analysis)
- **Psychology** (e.g. analysis of human behaviour and criminal investigation and prevention)

3.5 Expert Judgement

In the absence of sufficient practical data, uncertain variables and models can be computed using expert opinions. An expert is a person who is qualified with special knowledge and skills and with relevant experience in a specific domain (Leung & Verga, 2007). Expert judgements are the expression of inferential opinions, based on knowledge and experience. It is often used to quantify uncertain parameters in a probabilistic form (Otwayl & Winterfeldt, 1992).

Expert judgement can be qualitative or quantitative. Quantitative forms can be expressed as numerical values of probabilities, ratings, odds, uncertainty estimates and weighting factors. While qualitative forms can be represented as textual descriptions to reach an estimate for certain scenarios (Leung & Verga, 2007). Expert judgement is used to support decision-making in many different areas such as financial forecasting and assessing risks of terrorist attacks in the national security domain. The use of expert judgement has induced questions related to the accuracy of the obtained

results. However, there are many situations where expert judgement is the only source of accurate information regarding certain scenarios (Leung & Verga, 2007).

Expert judgment is a powerful tool in risk analysis. The uncertainty that surrounds the measure of probability in risk analysis is particularly hard to compute for rare and extreme events. This is the same when trying to estimate security risks for future and unknown events (Turisová et al., 2012).

3.5.1 Expert's Selection

The identification of experts is a critical part of the expert judgement process. It requires that one develops some criteria by which expertise can be measured (Otwayl & Winterfeldt, 1992). These criteria can be such as:

- Research in the related area as identified by publications and grants
- Citations of work
- Degrees, awards, or other types of recognition
- Availability and willingness to participate
- Recommendations and nominations from respected bodies and persons
- Positions held
- Membership or appointment to review boards, commissions, etc.

3.5.2 Expert Interview

Expert judgment can be realized using different techniques. One of the most common methods is the interview. An interview is carried out as a conversation between two individuals, the researcher and the interviewee. Experts are given a set of predetermined questions, whether using qualitative or quantitative methods. The interview questions may be related to the evaluation of a model, suggestions about some points linked to the study or different aspects of the area of the study (Tessmer, 1993). There are two ways of conducting an interview: structured and semi-structured (Britten, 1995; Rogers et al., 2011). A structured interview is usually used to provide more knowledge about the subject where the interviewees are asked a series of prepared questions. Semi-structured interviews include set of predetermined open and closed questions, with other questions emerging from the dialogue during the interview, by either the interviewer or interviewee, in order to explain an idea in more details (DiCicco-Bloom & Crabtree, 2006). Although the interviewer may face difficulties in finding participants, the interview is a flexible method that used to gain more knowledge in a certain area of study (Britten, 1995).

3.5.3 Phases of Expert Judgment

Regardless of the type of expert judgment, there are basic steps for obtaining an expert judgment (Benini et al., 2017), as depicted in Figure 3.19.

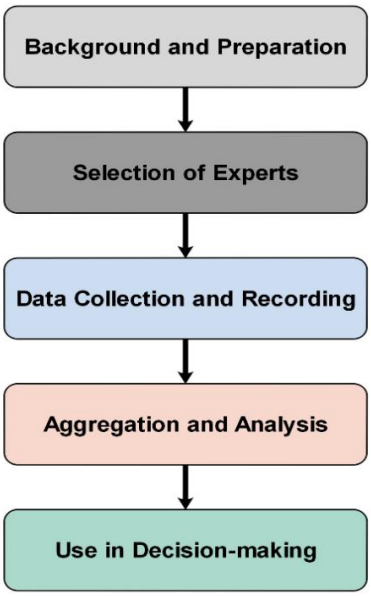


Figure 3.19: Phases of obtaining an expert judgment (Benini et al., 2017)

The first phase is the background and preparation. The need to obtain an expert judgment can vary in different situations, however, the main target of employing an expert judgment should be clearly understood. The researcher should be fully prepared and understand the required background and questions that need to be answered about the study.

Table 3.4: Methods of collecting expert judgment (Benini et al., 2017)

Data collection method	Features
Individual	• Best method for obtaining detailed data • Avoid potential bias from group dynamics • Data are easy to process and analyse • Limited collaboration between experts • Time-consuming
Interactive group	• Generate more accurate data, particularly for predictions • Appropriate for solving problems that require originality and insight • More appropriate for complex response modes, as participants can be collectively trained and guided • Potential for group-think bias • Heavy in preparation, administration and logistics • Strong moderator required, particularly if there are more than seven experts in a group
Delphi	• Experts individually make a prediction on a certain topic • Researcher aggregates all perspectives and shares the results with the contributing experts • Experts are then requested to update their predictions. • Over several rounds, the researcher tries to reach a consensus prediction • Limited bias between experts but time-consuming

The next phase is to select the appropriate experts to conduct the expert judgment according to the criteria discussed in the previous section. The logical next phase after selecting the experts is to record their responses using an audio recorder or by means of manual note-taking. There are three different ways to collect expert judgment: individual, interactive group, or Delphi (Benini et al., 2017), as summarized in Table 3.4. The collected data are then aggregated and analysed to extract the meaningful findings that can be later used in decision-making processes.

3.6 Summary

Chapter 3 has provided a discussion of risk estimation techniques. It started by providing an overview of quantitative risk estimation approaches discussed in related risk-based access control models with presenting their advantages and weaknesses. Then, the fuzzy logic approach with expert judgment was selected to be the suitable risk estimation technique to implement the risk-based access control model for the IoT. There were several reasons for this selection. Firstly, the fuzzy logic system provides a flexible model that is built on top of the experience of experts. Secondly, the fuzzy logic system is tolerant to imprecise data, so the lack of dataset, as in our research, can be resolved. Thirdly, a risk analysis based on the fuzzy logic system with consistent expert knowledge can create an effective method to assess risks in access control operations. Finally, most risk estimation techniques are subjective, but with consistent expert judgment, the subjectivity of the fuzzy logic system can be reduced. In addition, chapter 3 provided a discussion of the main stages of building a fuzzy logic system. This is followed by providing an overview of expert judgment and different phases required to obtain an expert judgment. The next chapter presents the proposed adaptive risk-based access control model for the IoT.

Chapter 4: Adaptive Risk-based Model

This chapter provides a discussion of the proposed adaptive risk-based access control model for the IoT system. It starts by discussing limitations of the existing static access control models and the need for a dynamic risk-based access control model for the IoT system. Section 4.2 provides a discussion of research problems that the literature fails to address. Then, section 4.3 presents the proposed adaptive risk-based access control model by highlighting its main elements and flow process. This is followed by discussing how the proposed model will address the research problems in section 4.4. Section 4.5 presents research methods that was utilized to achieve research targets. The chapter closes by providing a summary of the main points discussed through the chapter and introduces the next chapter.

4.1 Dynamic IoT System

The IoT is a dynamic system in nature where all environment and heterogeneous objects and things can be connected together to share their data and create new applications and services. Although the IoT brought unlimited benefits, it creates several challenges, especially in security. Achieving a higher level of security is a huge challenge due to the heterogeneous and distributed nature of the IoT system. In addition, applying sophisticated security algorithms could affect usability and user satisfaction. Hence, for the IoT system, the ultimate goal is to create an effective security system and at the same time consider the system usability (Habib & Leister, 2015).

One of the significant elements to address security challenges in the IoT is the access control model. This model is used to control access to system resources by allowing only authorized users who have been successfully authenticated (Liu et al., 2016). The major goal of the IoT system is to increase information sharing to maximize organization benefits and at the same time ensure the highest possible security measures are applied to prevent sensitive information disclosure. However, static access control models are built using predefined policies that give the same result in different situations. This binary decision (grant/deny) cannot create an effective level of security in a dynamic, heterogeneous and distrusted environment like the IoT system (Castiglione et al., 2016; Shen et al., 2018).

Static access control models cannot provide the required flexibility to diverse IoT applications. In addition, these models are associated with a system administrator who has access to all system resources. Compromising the administrator account can lead to the breach of almost all system confidential and sensitive data (Ye et al., 2014). Therefore, a dynamic access control model is required for the IoT system.

Risk-based access control model is one of the dynamic models that uses security risk associated with the access request as a criterion to determine the access decision. It estimates a risk value associated with each access request. Then, the estimated risk value is compared against risk policies to make the access decision (Shaikh et al., 2012).

4.2 Research Problems

This research aims to provide a dynamic and adaptive risk-based access control model for the IoT system. The literature has been examined in term of the scope of the research, as presented in chapter 2 and 3. After reviewing existing literature regarding risk-based access control models, the literature failed to:

- Provide a dynamic risk-based access control model for the IoT system. Most presented risk-based models did not focus on the IoT context where billions of sensors can be used to collect real-time and contextual features to determine access decisions in a dynamic manner. Therefore, this issue has been presented as the main research question, as follows:

  ***RQ**: What is the appropriate adaptive risk-based access control model for the IoT system?*

- Present a clear and effective risk estimation technique to estimate a risk value associated with each access request in a dynamic environment quantitatively. Providing a numeric value for the security risk is one of the biggest challenges the literature failed to address. Most presented risk estimation techniques did not provide a clear and precise method to provide a numeric value for the risk associated with each access request. This issue has been represented as one of the sub-research questions, as follows:

  ***SRQ1**: What is the appropriate risk estimation technique to estimate the risk associated with the access request?*

- Provide acceptable risk values that can be used to make access decisions. Most presented risk-based models suggested using a threshold risk value to grant or deny the access without providing any details about how to decide this threshold risk value in different applications. This issue has been represented as one of the sub-research questions, as follows:

  ***SRQ2**: What are acceptable risk values to make the access decision in IoT applications?*

- Provide a plug and play risk-based access control model that works when first used or connected, without reconfiguration or adjustment by the system administrator. In the literature, some risk-based access control models such as Shaikh et al. (2012), Li et al. (2013), Namitha et al. (2015), and Britton and Brown (2007) utilized risk history as a factor to determine access decisions. However, values of risk history will not be available at the start of setting up a new risk-based model, which will make the system unusable until collecting risk history values. This issue has been represented as one of the sub-research questions, as follows:

  SRQ3*: How to provide a plug and play risk-based model that can work when first used or connected to an IoT system?*

- Provide a scalable risk estimation technique that can cope with the constant increase of the number of IoT devices. Providing a clear risk estimation approach was not the only issue the literature failed to resolve, considering the growing rate of IoT devices that need fast and scalable risk estimation approach was not also addressed. There is no proof that the presented risk estimation techniques in the literature were tested to measure its scalability and response time, especially in the IoT context. This issue has been represented as one of the sub-research questions, as follows:

  SRQ4*: How to provide fast and scalable risk estimation technique to handle the constant increase in the number of IoT devices?*

- Consider a way to detect and prevent malicious activity during access sessions. Most existing access control models do not employ a method to detect malicious actions after granting access. In addition, related risk-based access control models lack abnormality detection capabilities that allow the system to detect and prevent abnormal behaviour in a timely manner during access sessions. This issue has been represented as one of the sub-research questions, as follows:

  SRQ5*: How will the user/agent behaviour be monitored during the access session?*

- Provide a way to evaluate related risk-based access control models using real-world scenarios. The ultimate target of any new approach is to guarantee that it is applicable in real-world scenarios. Related risk-based access control models discussed in the literature did not provide a way to validate and evaluate their risk models using real-world scenarios, especially in the IoT context. Therefore, this issue has been represented as one of the sub-research questions, as follows:

  SRQ6*: To what extent is the proposed risk-based model applicable to real-world IoT scenarios?*

4.3 Proposed Adaptive Risk-based Model

This work aims to address research problems discussed in the previous section by proposing a dynamic and adaptive risk-based access control model that uses real-time and contextual information collected from the IoT environment to provide access decisions. The next section provides a detailed discussion of the main components and flow process of the proposed risk-based model.

4.3.1 Model Structure

Unauthorized disclosure of information is one of the critical issues in the IoT system that need to be addressed. Current static access control models cannot resolve this challenge due to three reasons (Lee et al., 2007; Li et al., 2013). First, they are unable to handle exceptional situations in which the access policy itself should be overridden in order not to stop the system. Second, they do not meet the requirements of providing dynamic secure information and permission sharing in collaborative systems. Third, they are not flexible enough to handle the changing behaviour of users, especially in a dynamic environment like the IoT.

A risk-based access control model is one of the dynamic models that performs risk analysis to estimate the security risk value associated with the access request to make the access decision. The main issue solved by this model is the flexibility in accessing system resources (Dos Santos et al., 2014; Shaikh et al., 2012).

An adaptive risk-based access control model for the IoT is proposed, as shown in Figure 4.1. The proposed model has four inputs: user/agent context, resource sensitivity, action severity and risk history. These inputs/risk factors are used to estimate the security risk value associated with the access request. Then, the estimated risk value is compared against risk policies to make the access decisions. In addition, the user behaviour will be monitored to detect and prevent malicious actions from authorized users during their access sessions. The main reason to select only four risk factors is to ensure that the proposed model is generic and can be applied in various IoT applications. In addition, adding more risk factors will add computational complexity on the proposed model. The eventual goal of the proposed risk-based model is to create a system that encourages information sharing to maximize organization benefits while keeping users responsible for their actions and stopping the expected damage that the organization could suffer due to sensitive information disclosure. Moreover, organizations will be able to control insecure information flows dynamically based on its risk tolerance and environment (Chen et al., 2007).

The proposed risk-based model can work well in unexpected situations that often require the violation of security policies. This may occur because policies are incomplete or incoherent, sometimes even conflicting. The most usual examples of such needs are in medical and military

applications, where the need to take actions may save lives and the system immobility may cause serious harm (Dos Santos et al., 2014).

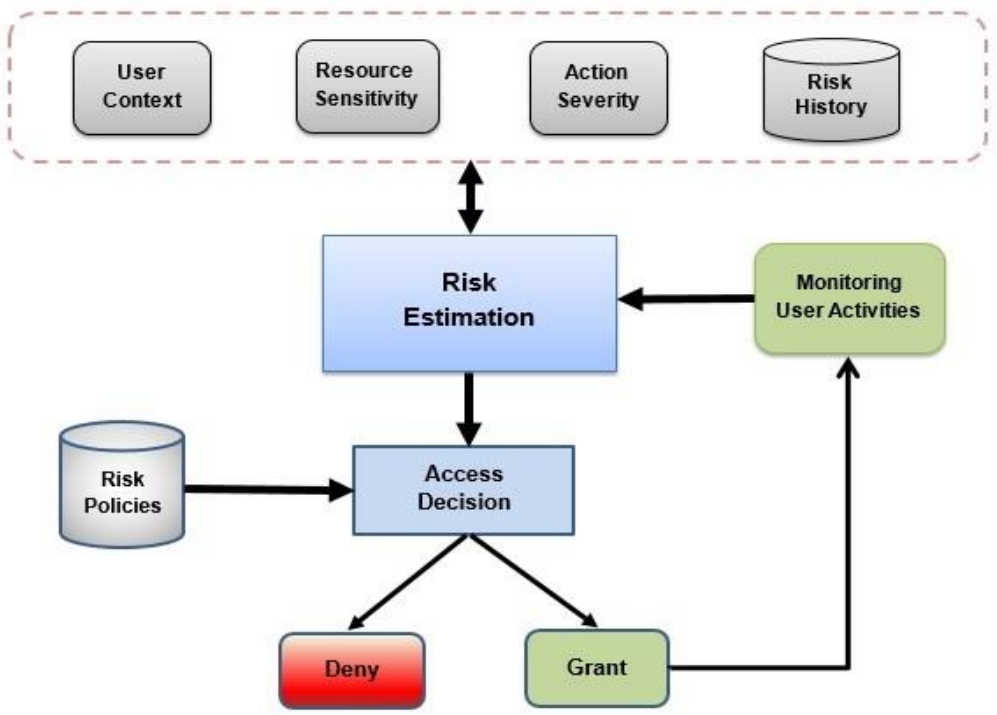


Figure 4.1: Proposed adaptive risk-based access control model

The main components of the proposed adaptive risk-based access control model involve risk factors, risk estimation module, risk policies, access decision and monitoring user activities. The following section provides an overview of each component.

- **User/Agent Context**: It represents environmental and contextual features that embedded with the user/agent at the time of making the access request. These features are collected while making the access request to determine the security risk value associated with the requesting user. Location, identity, time, history events and activity are the most common user/agent contextual features (Perera et al., 2014). The agent is used to express the diversity of applications in the IoT system. An agent represents any system entity that has the ability to make an access request (Feitosa, 2014). For the rest of this thesis, the word user will be used to represent either a user or agent.

- **Resource Sensitivity**: It describes how valuable the resource is to the owner or to the service provider. Data is assigned a level of sensitivity based on who should have access to it and how much damage would be done if it has been disclosed. A risk metric is assigned to each resource in the IoT system depending on how valuable the resource data is to the owner. Therefore, the higher the data sensitivity, the higher the risk metric associated with the resource.

- **Action Severity**: It represents the consequences of a certain action on a particular resource in terms of confidentiality, integrity, and availability. Different operations have different impacts and so have different risk values. For instance, the risk of a "view" operation is lower than the risk of a "delete" operation.

- **Risk History**: It represents the previous risk values of a user regarding a particular resource. This is because the risk history reflects users' behaviour patterns. Moreover, it is used to identify good and bad authorized users and predict their future behaviour (Li et al., 2013).

- **Risk Estimation Module**: It is the heart of the risk-based access control model. It is responsible for taking the input features/ risk factors to quantify the security risk value associated with each access request. There are two ways to estimate the risk: quantitative and qualitative. However, the ultimate goal in the context of access control is to define a numeric value for the security risk associated with each access request to determine the access decision.

- **Risk Policies**: They are mainly used by the risk estimation module to make access decisions. These policies are created by the resource owner or security system administrator to identify terms and conditions of granting or denying access to a particular resource. To determine the access decision, the estimated risk value resulted from the risk estimation module is compared against risk policies to determine the access decision. Defining a threshold risk value is one of the common ways to build a risk policy in risk-based access control models in which the access is granted only if the estimated risk value is lower than the threshold risk value.

- **Monitoring User Activities**: In existing access control models, if the decision is to grant access to a user, then there is no way to detect or prevent any abnormal and unusual data access from the authorized user. So, a monitoring module is needed to adjust the risk value based on the user behaviour adaptively during the access session. The proposed risk-based model utilizes smart contracts to monitor user behaviour or user activities during the access session. Applying smart contracts to accomplish this process is a big challenge especially it will be the first time to use smart contracts in this context. Smart contracts are treated as software code to enforce a functional implementation of particular demands and confirm that certain conditions or terms were met or not (Watanabe et al., 2016). For each user, a smart contract will be built to reflect user permissions. Hence, for each user access session, the behaviour will be compared with the smart contract to ensure the user obeys the terms and conditions of the smart contract so as to prevent any potential security breach during access sessions.

- **Access Decision**: It is the judgment of whether to grant or deny access. The access decision is not associated with permissions. This is because the user will specify the resources to access and the actions to perform in the access request. Therefore, only requested permissions will be granted or denied. The access decision in risk-based access control models is decided based on the estimated risk value of each access request. Then, the estimated risk value is compared with risk policies to determine whether to grant or deny access. Since smart contracts are used as abnormality detection capability to detect and prevent malicious actions during access sessions, three risk decision bands were proposed to determine the access decision, as shown in Figure 4.2.

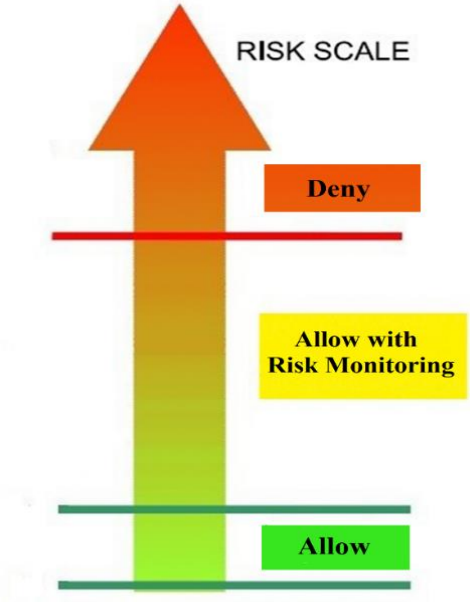


Figure 4.2: Access decision bands

  - **Allow band:** This band is used to grant access without monitoring user's activities during the access session to preserve the user's privacy. This band is very narrow in term of the risk value. It is used mainly to allow access of users associated with very low risk value without being monitored such as device owner or system security administrator.

  - **Allow with Risk Monitoring band:** This band is used to grant access with monitoring all users' behaviours and activities during the access session to detect and prevent any potential malicious activity. The ultimate target of the proposed risk-based access control model is to increase information sharing and at the same time guarantee security of system resources, so smart contracts are used to monitor user's activities during the access session. Therefore, this band is very wide to include most of the access to system resources.

  - **Deny band**: Due to the high-risk value associated with the user requesting access to system resources, the access will be denied through this band.

4.3.2 Process Flow of Proposed Model

The proposed risk-based access control model provides a dynamic method to authorize different types of users in the IoT system by estimating the security risk value associated with each access request. To understand the proposed risk-based model, Figure 4.3 provides a detailed description of the process flow of an access request.

The flow starts when the access control manager receives an access request from a user/agent. The access control manager asks for values of risk factors (user/agent contextual features, resource sensitivity level, action severity and risk history) of the requesting user. The risk estimation module uses these values to estimate the overall risk value associated with the access request. Then, the estimated risk value is compared against risk policies to determine the access decision. At this point, there are two decisions:

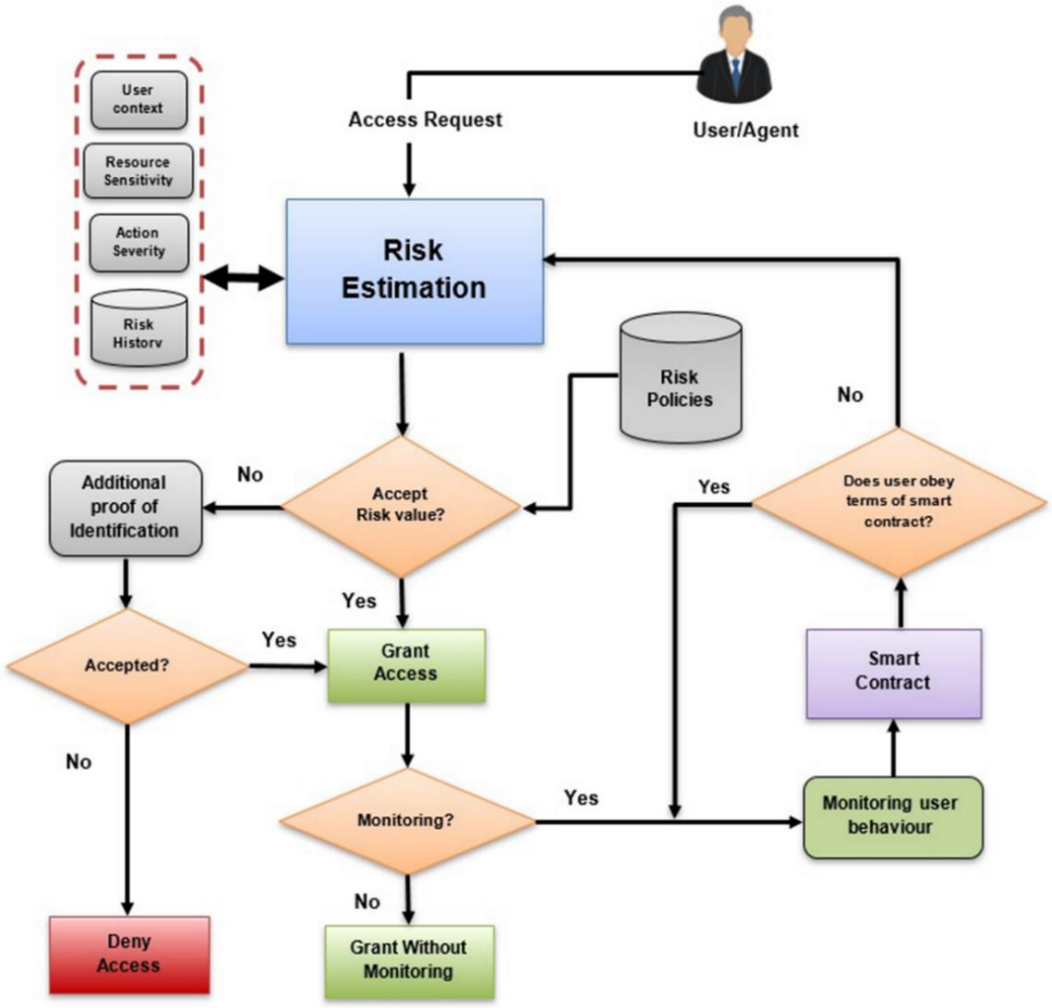


Figure 4.3: The process flow of the proposed adaptive risk-based access control model

a) If the access is granted, then, there are two possible scenarios. The first scenario is if the estimated risk value of the access request lies within the *Allow* band, the access will be granted without monitoring user activities during the access session. The second scenario is

if the estimated risk value of the access request lies within the *Allow with Risk Monitoring* band, the monitoring module will track and record user behaviour and activities to detect and prevent potential malicious actions. The smart contract will use the monitored data to determine if the user follows the terms and conditions of the contract or not. If the user obeys the terms of the contract, the system will keep monitoring the user behaviour, while if not, then it will return to the risk estimation module to reduce user permissions or terminate the access session to stop any potential malicious activity.

b) If the access is denied, the system will ask the user to provide additional proof of identifications so as not to block an authorized user and reduce the false-positive rate. If the user provided the required identifications, the access will be granted, and the flow continues as in decision (a). If the user does not provide the correct identifications, the system will deny access.

4.4 Solutions for Research Problems

There is a set of research questions that the literature failed to resolve regarding building a dynamic risk-based access control model for the IoT system, as discussed in section 4.2, The aim of this research is to provide best solutions to these questions. Table 4.1 provides a summary of the solutions provided by this research to address these questions.

Table 4.1: How this research will address research questions

Research Questions	Proposed Solutions	In Thesis
RQ: What is the appropriate adaptive risk-based access control model for an IoT system?	This research proposed a dynamic and adaptive risk-based access control model that uses contextual and real-time information collected from the IoT environment to make the access decision. This model can be used in various IoT application to adapt to unexpected situations and provide a flexible way to determine access decisions.	Chapter 4
SRQ1: What is the appropriate risk estimation technique to estimate the risk associated with the access request?	Providing a clear and accurate risk estimation technique to provide a quantitative risk value for each access request is one of the main targets of this research. After reviewing existing risk estimation techniques, the fuzzy logic system with expert judgment has been selected to implement the risk estimation process of the proposed model. A clear and detailed implementation of the risk estimation approach has been provided.	Chapter 3 & Chapter 5
SRQ2: What are acceptable risk values to make the access decision in IoT applications?	This research proposed three risk decision bands to grant or deny access. The first band grants access without monitoring, the second band grants access with monitoring, while the third band denies access. In this research, twenty security experts from inside and outside the UK were interviewed to provide acceptable risk values for each band.	Chapter 5

Table 4.1: How this research will address research questions (Cont.)

Research Questions	Proposed Solutions	In Thesis
SRQ3: How to provide a plug and play risk-based model that can work when first used or connected to an IoT system?	One of the issues associated with existing risk-based models that use risk history as one of the factors is that it couldn't operate immediately until data regarding previous risk values have been collected. This research resolved this issue by implementing a solution for the cold start problem that allows the proposed risk-based model to operate immediately when first used or connected.	Chapter 5
SRQ4: How to provide a fast and scalable risk estimation technique to handle the constant increase in the number of IoT devices?	This research provides a risk-based model for the IoT system which is growing in billions. Therefore, the risk estimation technique should be able to cope with the constant increase of the number of IoT devices and provide access decisions in a timely manner. One of the issues associated with the fuzzy logic system is time overhead and scalability. Therefore, this research proposed the NFS and Adaptive Neuro-Fuzzy Inference System (ANFIS) to resolve this issue.	Chapter 6 & Chapter 7
SRQ5: How will the user/agent behaviour be monitored during the access session?	The proposed risk-based model provides abnormality detection capability using smart contracts to track and monitor user behaviour during the access session to detect and prevent potential malicious actions. The operation of the smart contract was simulated using MATLAB Simulink and Stateflow diagrams to test system response for detecting abnormal and malicious activities.	Chapter 8
SRQ6: To what extent is the proposed risk-based model applicable to real-world IoT scenarios?	To evaluate and proof the applicability of the proposed risk-based model in real-world scenarios, three case studies of IoT applications were considered. Various access control scenarios of children hospital, network router and smart home were presented by discussing the access decision in each situation.	Chapter 8

4.5 Research Methodology

Building an adaptive risk-based access control model for the IoT system includes a variety of research methods in order to reach its targets. This section provides a discussion of research methods utilized to reach research targets.

Typically, there are two research methodologies to conduct research; qualitative and quantitative. Quantitative research methodology depends on measuring and analysing data to determine the relationship between one set of data with another to explain a certain phenomenon. The measurement of these variables might produce quantifiable conclusions. Thus, it places emphasis on methodology, procedure and statistical measures of validity (Eldabi et al., 2002). It uses fixed instruments that contain closed questions such as surveys (Creswell, 2003). Quantitative research is evaluated by either descriptive or inferential statistics (Taylor, 2005). Descriptive statistics are used to describe the characteristics of a specific sample of data, while inferential statistics are used to determine the likelihood of generalising the characteristics from small samples to larger ones (Taylor, 2005).

Qualitative research methodology is only concerned with identifying the meaning and understanding of a phenomenon. It is not concerned with the quantification of the phenomenon but providing an understanding of the phenomenon through observation (Berleant & Kuipers, 1997; Eldabi et al., 2002; Pang & Coghill, 2015). Qualitative methods aim to answer the question of "what" and "how" (Taylor, 2005). These methods can be such as interviews, observations, documents, open-ended questions, and audio-visual data. To derive results and answer research questions, the analysis of texts and images could also be used (Creswell, 2003; Taylor, 2005).

Table 4.2: Research methods used in this research for each research question

Research Question	Research Method	Description
SRQ1: What is the appropriate risk estimation technique to estimate the risk associated with the access request?	Expert interview	After reviewing existing risk estimation techniques in related risk-based models, the fuzzy logic with expert judgment was selected as the suitable method. Twenty IoT security experts were interviewed to validate the proposed model and confirm the fuzzy rules.
SRQ2: What are acceptable risk values to make the access decision for IoT applications?	Expert interview	Twenty IoT security experts were interviewed to provide acceptable risk values for the proposed three risk bands.
SRQ3: How to provide a plug and play risk-based model that can work perfectly when first used or connected?	Expert interview	To implement a solution for the cold start problem, ten security research fellows at the University of Southampton were interviewed to validate fuzzy rules.
SRQ4: How to provide a fast and scalable risk estimation technique to handle the constant increase of IoT devices?	Experiments	To resolve issues of time overhead and scalability associated with the fuzzy logic system, the ANFIS and NFS were employed. Several experiments were carried out to implement the risk estimation process using ANFIS and NFS approaches.
SRQ5: How will the user/agent behaviour be monitored during the access session?	Simulation	In this research, smart contracts are used to monitor user behaviour during access sessions. MATLAB Simulink and Stateflow diagrams were adopted to simulate the operation of smart contracts to evaluate its response in detecting malicious actions.
SRQ6: To what extent is the proposed risk-based model applicable to real IoT scenarios?	Access scenarios	To validate the applicability of the proposed risk-based access control model in real-world IoT applications, access control scenarios of three IoT applications including healthcare, smart home and network router were provided.

Pure quantitative models require accurate numerical information about the system structure and its initial state that are represented quantitatively (Rochette et al., 2009). When such data is unavailable, quantitative models face many constraints that restrict the model's value. In contrast, qualitative models display all possible behaviours but only in qualitative terms (Omar et al., 2015). The main

target of this research is to build a dynamic and adaptive risk-based access control model for the IoT that can work in unexpected situations by using not only access policies but also real-time and contextual features while making the access decision. There are multiple research methods utilized in this research to reach its target. Table 4.2 provides a summary of the research methods used to resolve research questions.

A description of the research methods employed in this study is presented as follows:

- **Interviews** are one of the common qualitative data collection methods. They can be structured, semi-structured, and unstructured (Gill et al., 2008). They are considered an informal validation method, as they are based on human subjectivity. The data resulting from these interviews can be both qualitative and quantitative, depending on the material presented in the interview (Balci, 1994). In this research, semi-structured interviews were employed. This type of interview provides the ability for both the interviewer and the interviewee to respond to questions with more detail. The questions within the semi-structured interviews elicit expected information alongside other unanticipated information (Gill et al., 2008).

- **Simulation** allows researchers to assume the inherent complexity of organizational systems as a given. If other methods answer the questions "What happened, how, and why?", simulation helps to answer the question "What if?". Simulation enables studies of complex systems because it creates observations by "moving forward" into the future, whereas other research methods attempt to look back across history to determine what happened, and how (Dooley, 2002). In this research, simulation is used as a method to imitate the operation of smart contracts to monitor user activities during access sessions for the IoT system.

- **Experiments** are a systematic and scientific approach which allows the researcher to manipulate one or more variables and measure any changes in other variables. True experimental research is considered to be successful only when the researcher confirms that a change in the dependent variable is solely due to the manipulation of the independent variable (Moore & McCabe, 1993). The results of experimental research once analysed, can be applied to various other similar aspects. In this research, experiments are used to build the risk estimation process using both ANFIS and NFS approaches.

4.6 Summary

Chapter 4 has presented the proposed adaptive risk-based access control model for the IoT. It started by discussing the need for a dynamic access control model for the IoT system. This is followed by discussing the research problems that the literature failed to address. One of the major issues extracted from the literature was the lack of a dynamic risk-based model that can adapt to different

and unexpected circumstances of the IoT system. Therefore, chapter 4 presented the proposed adaptive risk-based access control model for the IoT to address this issue. This model uses the security risk as a criterion to make the access decision. It estimates the risk value associated with each access request using four inputs: user contextual features, resource sensitivity, action severity and risk history. The estimated risk value is then compared against risk policies to determine the access decision. To prevent and detect abnormal misuse from authorized users during the access session, the proposed model uses smart contracts to monitor user's activities and adjust their risk values adaptively based on their actions. In addition, a discussion of how the proposed adaptive risk-based model will resolve research problems extracted from the literature was also provided. Finally, the research methods used in this research were presented. The next chapter presents the implementation of the risk estimation process using the fuzzy logic system with expert judgment.

Chapter 5: Implementation of Risk Estimation using Fuzzy Logic

This chapter provides a discussion of the implementation of the proposed risk estimation technique using the fuzzy logic system with expert judgment. It starts by discussing the integration of the fuzzy logic system with expert judgment. Then, section 5.2 provides a discussion of the expert interview by highlighting the interview design, sample size and experts' attributes. Section 5.3 presents a discussion of validating the proposed risk-based access control model using IoT security experts. This is followed by providing a step-by-step discussion of the implementation of the risk estimation process using the fuzzy logic system with expert judgment in section 5.4. Section 5.5 provides a discussion of experts' responses to determine acceptable risk values for risk decision bands. Section 5.6 introduces the cold start problem and the proposed solution to address it. Section 5.7 presents a set of experiments to evaluate the efficiency of the proposed risk estimation technique when increasing the number of access requests and determine the most efficient MF, defuzzification method, and rule aggregation operator. The chapter closes by providing a summary of the main points discussed through the chapter and introduces the next chapter.

5.1 Proposed Risk Estimation Approach

Risk-based access control model is one of the dynamic models that provides an efficient way to provide access decisions. It uses the security risk value associated with the access request as a criterion to make the access decision. Building a risk-based access control model to decide whether to grant or deny access for each access request needs providing a quantitative value for the security risk associated with each access request. This process is complicated as it is based on the possibility of information leakage and the value of this information for various incidents that will occur in the future. So, the objective of the risk estimation process in the access control context is to provide an accurate and realistic numeric value for the security risk associated with the access request to determine the access decision.

After reviewing related risk estimation techniques in the literature in section 3.2, the fuzzy logic approach with expert judgement was selected as the appropriate technique to implement the risk estimation process of the proposed risk-based model, as shown in Figure 5.1. The fuzzy logic system has the ability to convert linguistic expressions and human reasoning into quantitative values. Combining expert judgment gives more weights to human reasoning as it comes from experts in the domain. Typically, the fuzzy logic system ensures that we do not neglect human common sense, and experiences. It allows the use of degrees of truth to calculate risk values (Li et al., 2013). Using security experts, fuzzy variables can be identified to build the fuzzy model to estimate security risks of access control operations.

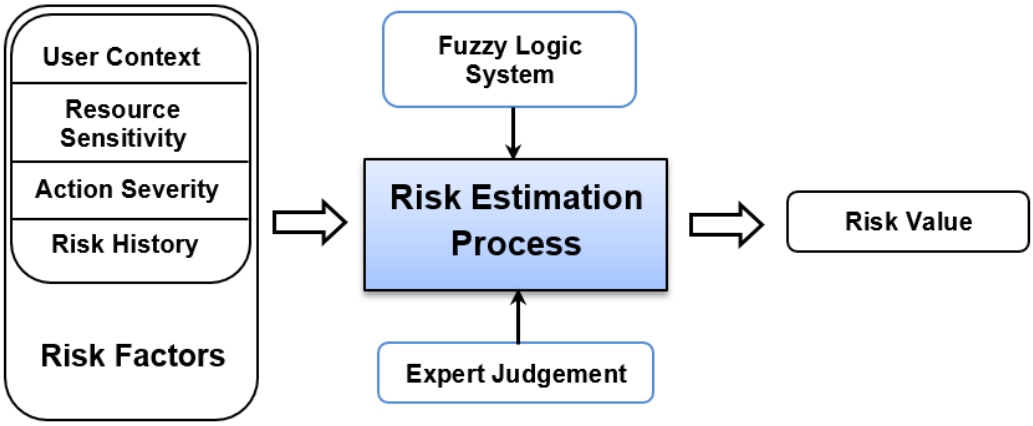


Figure 5.1: Proposed risk estimation approach using the fuzzy logic with expert judgment

One of the most effective ways to collect knowledge of experts and get an expert judgment regarding certain research is through an interview. For this research, the interview was conducted to get more information about the proposed risk-based access control model from highly experienced persons. The next section provides a detailed discussion of the interview including interview design, sample size, expert attributes and ethical approval.

5.2 Expert Interview

For this research, the objective of the interview was to validate fuzzy rules of the proposed risk estimation technique to ensure correct and appropriate fuzzy rules were built. In addition, getting valuable feedback about the proposed risk-based access control model and determine acceptable risk values of risk decision bands. The interview involved four sections, as follows:

- Section 1 was designed to collect background information of participating experts.
- Section 2 was designed to validate the proposed risk-based access control model.
- Section 3 was built to validate fuzzy rules of the proposed risk estimation technique by IoT security experts.
- Section 4 was designed to determine acceptable risk values of risk decision bands.

5.2.1 Interview Design

The interview was built as a semi-structured, which include a set of predetermined open and closed questions with other questions emerging from the dialogue during the interview, by either the interviewer or interviewee (DiCicco-Bloom & Crabtree, 2006).

The interview questions consisted of four sections. The first section has involved open questions to get the background information of the participants. The second section was used to get experts' feedback about the proposed risk-based model. This section was carried out through open questions. The third section was used to validate fuzzy rules that previously created using the information collected from literature with the researcher experience. This section was carried out using closed questions using five choices to be the expected output for each rule. The final section discussed different IoT security experts' view of acceptable risk values of risk decision bands. This section was carried out through open questions.

The interview questions were pilot-tested by seven security research fellows at the University of Southampton. To interact directly with the interviewees and provide further questions based on the interviewees' answerers, face-to-face interviews were used. Some interviews were conducted on the campus of the University of Southampton in the expert's office. Other interviews were conducted online using video conferencing on Skype (Iacono et al., 2016) and were recorded by an audio recorder or taking notes manually.

All interviews were conducted in the English language. Appendix A provides the participant information sheet, consent form and interview questions.

5.2.2 Ethics Approval

Before starting the interview, each expert was asked to sign a consent form after reading the participant information sheet that included all the necessary information, terms and conditions about the study. The University of Southampton Ethics Committee granted approval for this study under their reference number ERGO/FPSE/25091.

5.2.3 Sample Size

When conducting interviews, it is important to find the appropriate number of experts, as this will help to produce accurate results. According to this point, determining the minimum sample size is essential when it comes to producing consistent results (Bhattacherjee, 2012). In terms of the number of experts, according to Guest et al. (2006), there is no agreed-upon number of experts for an interview in a content validity study. However, most researchers recommend a panel consisting of 3 to 15 experts (Bhattacherjee, 2012). The main criterion the researcher used to determine the number

of experts to validate the research is reaching saturation in which more interviews will not add new ideas for the research.

Table 5.1: Attributes of IoT security experts who have interviewed in this study

Expert No	Job Description	Experience (Years)	Know Fuzzy logic	Knowledge of IoT Applications	UK/Non-UK
E 1	IoT Security researcher	6 – 10	Yes	Connected industry, smart city, connected car, and connected healthcare	UK
E 2	Senior Cybersecurity Engineer	More than 10	Yes	Connected industry, smart city, smart energy, connected car, and connected healthcare	UK
E 3	IoT Security researcher	More than 10	No	Smart city, smart energy, and smart home	UK
E 4	IoT Security researcher	6 – 10	Yes	Smart energy, connected car, and smart home	India
E 5	Security Administrator	2 – 5	No	Connected industry, smart energy, smart home, and connected healthcare	Egypt
E 6	IoT Security researcher	2 – 5	Yes	Smart city	UK
E 7	Risk analysis professor	2 – 5	Yes	Smart city, smart energy, and smart home	UK
E 8	IoT Security researcher	2 – 5	No	Smart energy and connected healthcare	India
E 9	Security Administrator	2 – 5	No	Smart city and smart home	Egypt
E 10	Senior Cybersecurity Engineer	2 – 5	Yes	Smart city, smart home, and connected healthcare	UK
E 11	Security Specialist	6 – 10	Yes	Connected healthcare	Italy
E 12	Security Administrator	6 – 10	Yes	Smart home	Egypt
E 13	Security Specialist	6 – 10	No	Connected industry and smart home	UK
E 14	IoT Security researcher	2 – 5	Yes	Connected industry, smart city, smart energy, and connected car	UK
E 15	Security Specialist	2 – 5	Yes	Smart city, smart energy, and smart home	UK
E 16	Security Administrator	2 – 5	Yes	Smart city	KSA
E 17	IoT Security researcher	2 – 5	No	Smart city and Smart energy	Romania
E 18	Security Administrator	6 – 10	Yes	Smart city, smart energy, and smart home	Egypt
E 19	Security Administrator	6-10	Yes	Connected industry, smart city, connected car, and connected healthcare	Egypt
E 20	IoT Security researcher	2 – 5	Yes	Smart city, smart energy, and smart home	UK

The interviews were conducted with twenty IoT security experts from inside and outside the UK. The criteria used to select experts were years of experience in security and familiarity with IoT applications. The IoT security researchers who have been interviewed in this study were selected after investigating and reading their works and making sure that there is a relevancy between their

work and this study. While other experts were selected depending on their current jobs that require a great experience in security and IoT applications. Information of experts who have involved in this study can be shown in Table 5.1.

Most interviewed experts had large experiences in security and IoT applications. Most experts had at least 2- 5 years of experience. In addition, although validating fuzzy rules of the proposed risk estimation technique does not require extensive knowledge about the fuzzy logic approach and how it works as it only requires human reasoning, 70% of interviewed experts had adequate knowledge about the fuzzy logic approach. Typically, fuzzy rules are constructed using linguistic expressions of the English language, which are easy to understand and interpret. For experts who did not have knowledge about the fuzzy logic approach, the researcher spent about 10 minutes to make sure the participant understands essential information about the fuzzy logic approach and how a fuzzy rule can be built using linguistic expressions.

5.3 Implementation of Fuzzy Logic Technique

The proposed risk-based access control model has four risk factors: user context, resource sensitivity, action severity and risk history which are used as input to determine the security risk value associated with the access request to make the access decision for various IoT applications. MATLAB fuzzy logic toolbox was used to implement the risk estimation process using the fuzzy logic system with expert judgment. MATLAB provides an efficient framework and easy-to-use graphical user interfaces that can generate surfaces and plots to analyse the system performance (Mathworks, 2016).

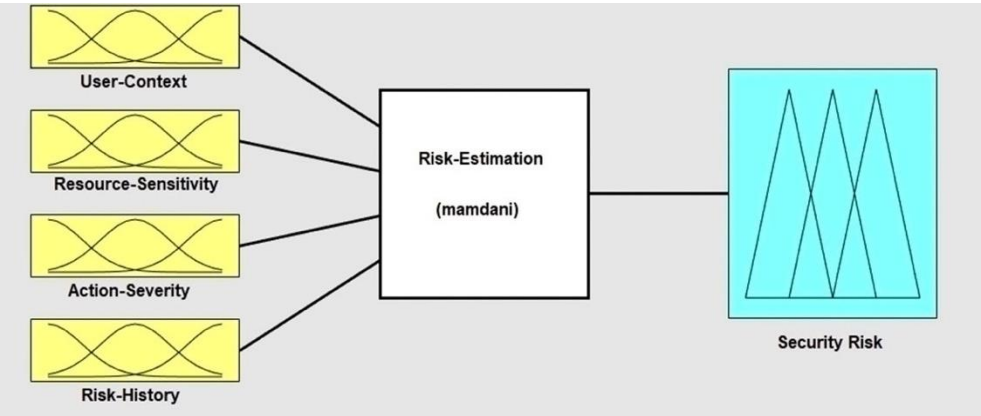


Figure 5.2: Risk estimation implementation in MATLAB fuzzy logic toolbox

To implement the proposed risk estimation technique using the fuzzy logic toolbox, there are two built-in FISs; Mamdani and Sugeno, as discussed earlier in section 3.4.3. Since no available dataset exists in this research, Mamdani FIS will be used to implement the risk estimation process. It is intuitive, has widespread acceptance, and is well suited to human input. The structure of input and output of Mamdani FIS to implement the proposed risk estimation technique can be shown in Figure 5.2.

Implementing the Mamdani FIS to estimate security risks of access control operations requires five stages; fuzzification, MF, fuzzy inference rules, rule aggregation, and defuzzification. In the next section, each stage will be discussed in detail by showing corresponding MATLAB images.

5.3.1 Fuzzification

The first step to implement the proposed fuzzy risk estimation technique is to define input and output variables and their corresponding linguistic expressions. These linguistic expressions are called fuzzy sets. The proposed risk-based model has four inputs: user context, resource sensitivity, action severity and risk history, which will be used to produce the output risk. Input risk factors and output risk are divided into fuzzy sets. The user context, action severity, and risk history are represented using three fuzzy sets; "Low", "Moderate", and "High". Also, the resource sensitivity is represented using three fuzzy sets; "Not Sensitive", "Sensitive", and "Highly Sensitive". While the output risk is represented using five fuzzy sets; "Negligible", "Low", "Moderate", "High", and "Unacceptable High". Since the degree to which a value is a member of a certain fuzzy set can be any value between 0 and 1 (Li et al., 2013), the range of each fuzzy set should be determined accurately. Using related fuzzy logic models in the literature (Li et al., 2013; Ni et al., 2010), ranges of fuzzy sets of both input and output were determined. Table 5.2 shows linguistics variables for both input and output and their fuzzy ranges.

Table 5.2: Input and output linguistic variables and their range

Linguistic Expression	Notation	Range
Input Variable: User Context		
Low	L	0.0 – 0.4
Moderate	M	0.3 – 0.7
High	H	0.6 –1.0
Input Variable: Resource Sensitivity		
Not Sensitive	NS	0.0 – 0.35
Sensitive	S	0.2 – 0.5
Highly Sensitive	HS	0.45 – 1.0
Input Variable: Action Severity		
Low	L	0.0 – 0.4
Moderate	M	0.35 – 0.7
High	H	0.6 – 1.0
Input Variable: Risk History		
Low	L	0.0 – 0.4
Moderate	M	0.3 – 0.7
High	H	0.6 – 1.0
Output Variable: Risk		
Negligible	N	0.0 – 0.3
Low	L	0.1 – 0.4
Moderate	M	0.2 – 0.6
High	H	0.4 – 0.8
Unacceptable High	UH	0.7 – 1.0

5.3.2 Membership Function

The main purpose of this step is to choose the appropriate MF for input and output fuzzy sets. Each fuzzy set should have a corresponding MF that returns the degree of membership for a given value within the fuzzy set. Fuzzy sets can be represented using a variety of MFs, as discussed earlier in section 3.4.2. Choosing the appropriate MF depends on the available dataset. Comparing the results of the training data with the real data and calculate error values using Mean Average Percentage Error (MAPE) guarantee choosing the appropriate MF. However, when there is no available dataset, it is recommended to select the triangular MF (Li et al., 2013). This is because it provides an adequate representation of the expert knowledge, and at the same time simplifies the process of computation. Since there is no available dataset to show different combinations of input and their output for a set of scenarios, the triangular MF is used to represent input and output fuzzy sets of the proposed the risk estimation technique. Figures 5.3 – 5.7 show the representation of triangular MF for the input risk factors and the output risk.

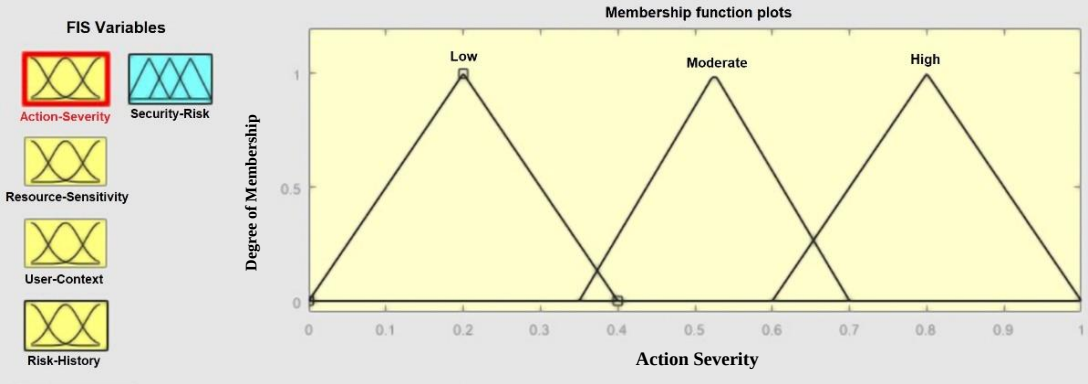


Figure 5.3: Triangular MF of the action severity

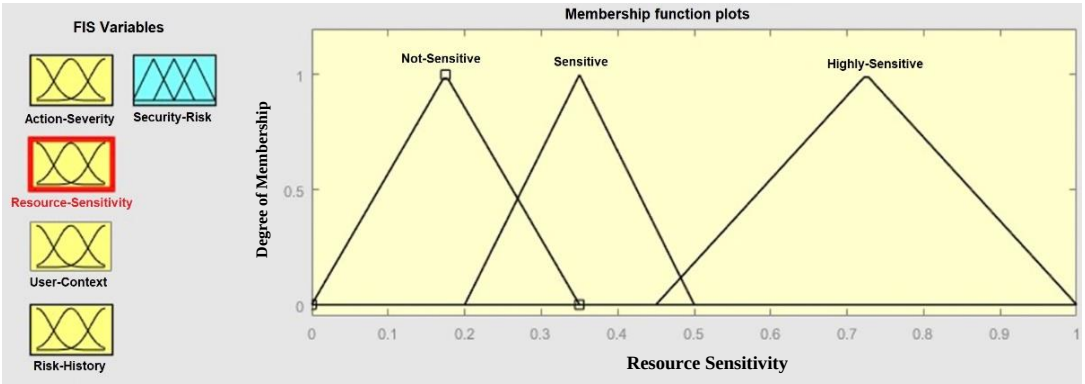


Figure 5.4: Triangular MF of the resource sensitivity

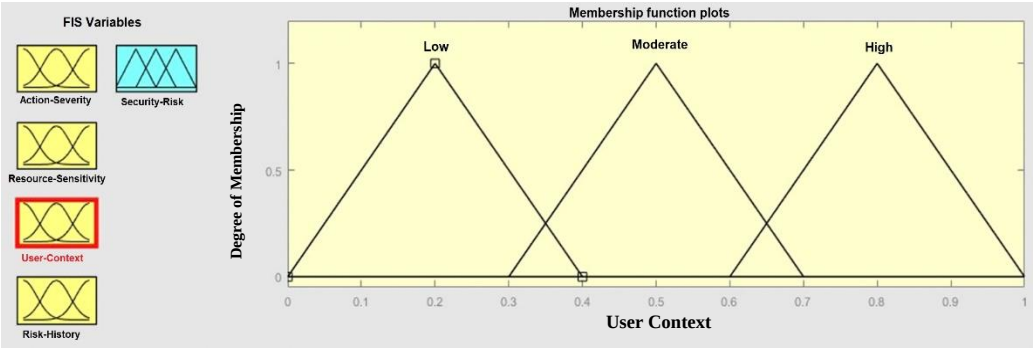


Figure 5.5: Triangular MF of the user context

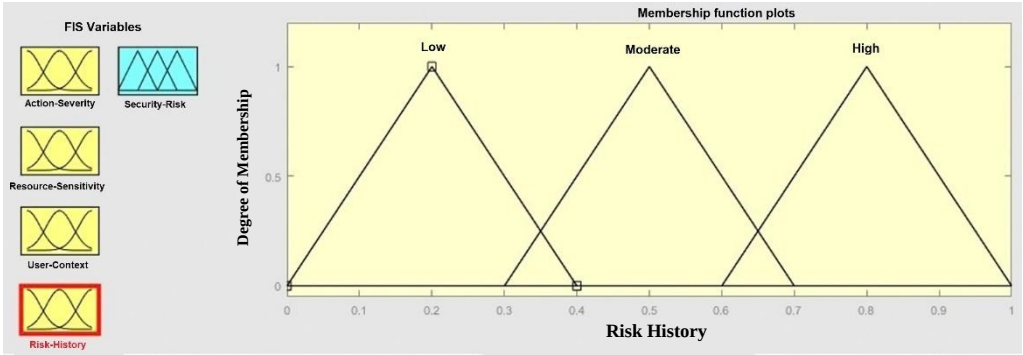


Figure 5.6: Triangular MF of the risk history

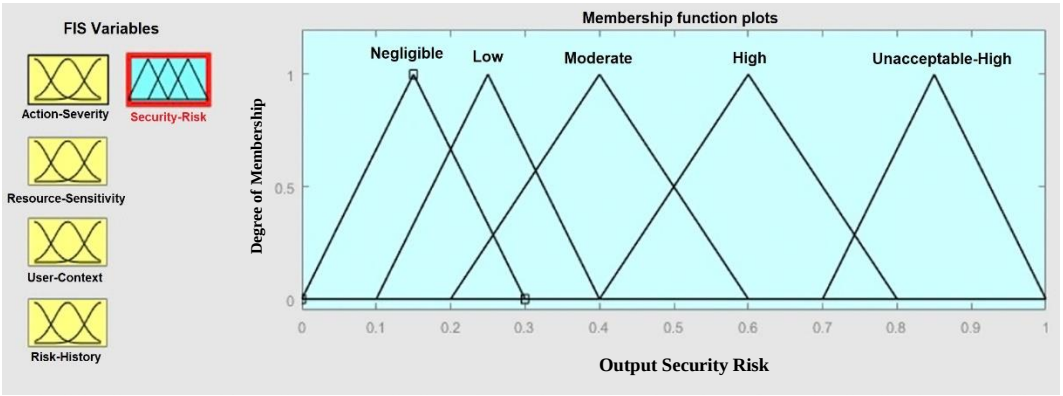


Figure 5.7: Triangular MF of the output risk

5.3.3 Fuzzy Inference Rules

One of the most significant stages to implement the proposed fuzzy risk estimation technique is to build appropriate and correct fuzzy inference rules that represent relationships between input and output linguistic expressions (Li et al., 2013). Having specified the risk and its factors, the logical next step is to specify how the output risk varies as a function of the four risk factors. Fuzzy rules are the brain of the fuzzy logic system that need to be specified accurately.

Fuzzy inference rules are built using IF-THEN statements, which are used to specify how the output risk varies as a function of the input. The IF-THEN rule uses linguistic expressions to describe the

relationship between input conditions and output. The IF part is used to represent the condition while the THEN part is used to provide the output in a linguistic form (Bai & Wang, 1982). For example, IF (the action severity is low) THEN (the output risk will be negligible). So, in this example, if the condition (action severity is low) is verified, then the output will be negligible. The IF part can involve multiple conditions but the THEN part includes only one output. For example, IF (action severity is Low & resource sensitivity is Not Sensitive & user context is Moderate & risk history is High) THEN (the output risk will be Moderate).

For this research, building fuzzy rules was the most intensive aspect that took a long time to complete. Fuzzy rules were created in two stages; the first stage involved building fuzzy rules using the information collected from related fuzzy models that have been reviewed in the literature with the researcher experience. The second stage was utilized to validate fuzzy rules through IoT security experts. Therefore, fuzzy rules were created by the researcher and then IoT security experts were interviewed to validate these rules either by accepting it or by suggesting different output.

5.3.3.1 Building Fuzzy Rules

Fuzzy rules are used to define the relationship between the output risk and input risk factors. It builds input combinations with the corresponding output in the form of IF-THEN statements. Since the proposed model has four inputs/risk factors, each input has three fuzzy sets, as depicted in Table 5.2. Therefore, the total number of input combinations will be 3*3*3*3=81. So, the total number of fuzzy rules is 81. All input combinations were built, and the output was decided using the information collected from the literature with the researcher experience. Some of the important information used to create fuzzy rules was the relation between action severity and resource sensitivity (Li et al., 2013), as shown in Figure 5.8.

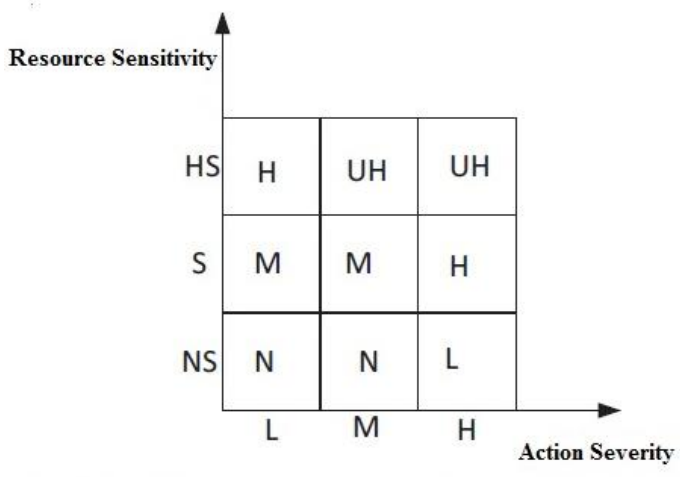


Figure 5.8: Fuzzy matrix of resource sensitivity with action severity (Li et al., 2013).

In addition, a set of logical rules between input and output were utilized to facilitate extracting appropriate output for fuzzy rules. These logical rules include:

- If the risk history increased, the output risk will not decrease.
- If the resource sensitivity increased, the output risk will not decrease.
- If the action severity increased, the output risk will not decrease.
- If any two risk factors are high, the lowest output will be Moderate.
- If the resource sensitivity is Highly Sensitive (HS) or Sensitive (S), the output risk cannot be Negligible (N).

Fuzzy rules were divided into five groups based on the output to facilitate analysing each group. The rules are given numbers to easily refer to them especially at the time of making comparisons after being validated by IoT security experts. Table 5.3 shows fuzzy rules when the output risk was N. Notations of input and output risk linguistic expressions can be shown in Table 5.2.

Table 5.3: Fuzzy rules when the output was N

Rule No	Risk Factors				Output Risk
	Action Severity	Resource Sensitivity	User context	Risk History	
1	L	NS	L	L	N
2	M	NS	L	L	N
3	H	NS	L	L	N
4	L	NS	M	L	N
5	M	NS	M	L	N
6	H	NS	M	L	N
7	L	NS	H	L	N
8	L	S	L	L	N
9	M	S	L	L	N
10	L	NS	L	M	N
11	M	NS	L	M	N
12	L	NS	M	M	N
13	M	NS	M	M	N

As illustrated in Table 5.3, the resource sensitivity was the dominant risk factor to determine the output of these fuzzy rules in which the lower the resource sensitivity, the lower the output risk. For instance, when the resource sensitivity was NS, the rule output became N regardless of the values of other risk factors. This was the same scenario for all these set of fuzzy rules except rule 8 and 9 where resource sensitivity was S. Based on the logical rule that stated, “If the resource sensitivity increased, the output risk will not decrease”, the output risk should be at least L, however, the output risk is assumed to be N, since both user context and risk history were L. This implies that the access comes from a trusted user with low risk regarding contextual features.

Table 5.4 shows fuzzy rules when the output risk was L. There is no dominant risk factor affecting the output risk directly, but the combination of the four risk factors led to L in the output risk. For the first four rules; 14, 15, 16 and 17, although the resource sensitivity was S and HS, both user context and risk history were L which led to L in the output risk.

Table 5.4: Fuzzy rules when the output was L

Rule No	Risk Factors				Output Risk
	Action Severity	Resource Sensitivity	User context	Risk History	
14	H	S	L	L	L
15	L	HS	L	L	L
16	M	HS	L	L	L
17	H	HS	L	L	L
18	L	S	M	L	L
19	M	S	M	L	L
20	M	NS	H	L	L
21	H	NS	L	M	L
22	L	S	L	M	L
23	H	NS	M	M	L
24	L	NS	H	M	L
25	L	NS	L	H	L
26	M	NS	L	H	L
27	L	NS	M	H	L
28	M	NS	M	H	L

The same scenario was in rules 18, 19, and 22. Although the resource sensitivity was S which lead to increasing the output risk, having L in two of the risk factors led to L in the output risk. For rules 25, 26, 27 and 28, although the risk history was H, having NS in the resource sensitivity and either L or M in both action severity and user context led to L in the output risk. For rules 20, 21, 23, and 24, the resource sensitivity was the dominant risk factors in which having NS led to L in the output risk.

Table 5.5: Fuzzy rules when the output was M

Rule No	Risk Factors				Output Risk
	Action Severity	Resource Sensitivity	User context	Risk History	
29	H	S	M	L	M
30	L	HS	M	L	M
31	M	HS	M	L	M
32	H	HS	M	L	M
33	H	NS	H	L	M
34	L	S	H	L	M
35	M	S	H	L	M
36	M	S	L	M	M
37	H	S	L	M	M
38	L	HS	L	M	M
39	M	HS	L	M	M
40	L	S	M	M	M
41	M	S	M	M	M
42	M	NS	H	M	M
43	H	NS	H	M	M
44	H	NS	L	H	M
45	L	S	L	H	M
46	H	NS	M	H	M
47	L	NS	H	H	M

Table 5.5 shows fuzzy rules when the output was M. There is no dominant risk factor in this set of fuzzy rules. For rules 29, 30, 31, and 32, although the resource sensitivity was S and HS, both user

context and risk history were M and L respectively which led to M in the output risk. The same scenario was for rules 33, 34, and 35 where having H in the user context and S in the resource sensitivity led to M in the output risk. The output of eleven of these set of fuzzy rules was derived based on the logical rule that stated, "if two risk factors were H, the lowest output will be M". These rules include 29, 32, 33, 34, 35, 37, 43, 44, 45, 46, and 47. For rules 36, 38, 39, 40, 41, and 42, the output risk was decided to be M since risk history and resource sensitivity were M and S/HS respectively.

Table 5.6 shows fuzzy rules when the output was H. The resource sensitivity was the dominant risk factor for these set of rules in which the output risk was decided to be H if the resource sensitivity was S or HS. In addition, if the resource sensitivity was S and the risk history was H; the output risk became H regardless of values of other risk factors. This is because the user who wants to access confidential data and had a bad risk history score should be characterized as a malicious user. The output of most of these set of fuzzy rules was derived based on the logical rule that stated, "If the resource sensitivity increased, the output risk will not decrease".

Table 5.6: Fuzzy rules when the output was H

Rule No	**Risk Factors**				**Output Risk**
	Action Severity	**Resource Sensitivity**	**User context**	**Risk History**	
48	H	S	H	L	H
49	L	HS	H	L	H
50	H	HS	L	M	H
51	H	S	M	M	H
52	L	HS	M	M	H
53	M	HS	M	M	H
54	H	HS	M	M	H
55	L	S	H	M	H
56	M	S	H	M	H
57	L	HS	H	M	H
58	M	S	L	H	H
59	L	S	M	H	H
60	M	S	M	H	H

Table 5.7 shows fuzzy rules when the output was UH. These rules represent the highest output risk that used particularly to deny access requests to protect system resources. For these set of rules, at least two risk factors should be H with S or HS in resource sensitivity to have UH in the output risk. For instance, if the user context was H, the resource sensitivity was S or HS, and the risk history was H, the output risk will be UH. In addition, as the risk history indicates the past behaviour, having H in the risk history and S or HS in the resource sensitivity demonstrates a malicious activity that needs to be denied.

Table 5.7: Fuzzy rules when the output was UH

Rule No	Risk Factors				Output Risk
	Action Severity	Resource Sensitivity	User context	Risk History	
61	M	HS	H	L	UH
62	H	HS	H	L	UH
63	H	S	H	M	UH
64	M	HS	H	M	UH
65	H	HS	H	M	UH
66	H	S	L	H	UH
67	L	HS	L	H	UH
68	M	HS	L	H	UH
69	H	HS	L	H	UH
70	H	S	M	H	UH
71	L	HS	M	H	UH
72	M	HS	M	H	UH
73	H	HS	M	H	UH
74	M	NS	H	H	UH
75	H	NS	H	H	UH
76	L	S	H	H	UH
77	M	S	H	H	UH
78	H	S	H	H	UH
79	L	HS	H	H	UH
80	M	HS	H	H	UH
81	H	HS	H	H	UH

5.3.3.2 Validation of Fuzzy Rules by Experts

One of the problems associated with fuzzy logic models is the lack of appropriate data to create correct and appropriate fuzzy rules. If a dataset is available, fuzzy rules can be built dynamically and efficiently. In this research, there is no dataset, so there is no way to ensure correct and precise fuzzy rules were created. One solution to resolve this issue is to create fuzzy rules based on the knowledge and expertise of experts. Therefore, twenty IoT security experts were interviewed to validate fuzzy rules that are previously created using the information collected from the literature with the researcher experience to increase the accuracy of the fuzzy model.

Twenty IoT security experts from inside and outside the UK were interviewed to ensure suitable fuzzy rules were created. Experts' responses were analysed using the SPSS software program. The mean function was utilized to determine the final decision regarding the output risk of each rule. The mean, also called average, is the most common function used to measure the spread of values in statistics. It is used to ensure all responses of experts involved in the study were considered and have the same weight (Phinyomark et al., 2012).

To implement experts' responses in the SPSS program, responses were given ratings in which Negligible =1, Low =2, Moderate =3, High =4, and Unacceptable High =5. Therefore, the output of each fuzzy rule should be mapped to one of these five categories. An assumption was made in which any mean value lower than 0.5 will be mapped to the lower category and any mean value higher than or equal 0.5 will be mapped to the higher category. For instance, if the mean value is 1.25, the fuzzy

rule output will be mapped to 1 (Negligible), while if the mean value is 1.6, the fuzzy rule output will be mapped to 2 (Low). So, any mean value lies between 0 to 1.49, the rule output will be mapped to Negligible, while if the mean value lies between 1.5 to 2.49, the rule will be mapped to Low, and so on. Mapping the mean value to the linguistic expression of output risk can be illustrated in Figure 5.9.

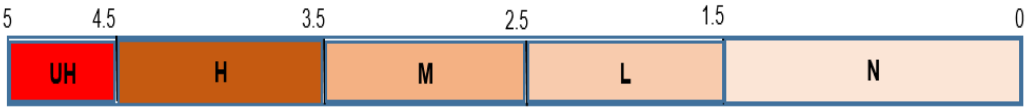


Figure 5.9: Mapping the mean value to output risk linguistic expression

Table 5.8 shows validation of fuzzy rules by IoT security experts when the output was N. Expert's responses were different from the output derived using the information collected from the literature with the researcher experience in which the output of only five rules was classified as N. Examining these rules indicates that experts decided to classify the output of a certain rule as N either if there were three risk factors with L or two risk factors with L and one NS and the fourth risk factor was M or L. On the other hand, experts decided to classify the output of six rules as L. These rules involve rule 3, 5, 6, 8, 11, and 12. Investigating these rules demonstrates that experts decided to classify the output of a fuzzy rule as L when one risk factor was H or S even when the other three risk factors were L or NS. They decided to be more careful when one of the risk factors was H or S and at least the fuzzy output should be L. For instance, in rule 3, although user context, risk history and resource sensitivity were L, L and NS respectively, having H in the action severity made experts to classify the output of this rule as L.

Table 5.8: Validation of fuzzy rules when the output was N

Rule No	Number of Experts		Mean	Experts Responses					Mapped Category	Rule Output
	Valid	Missing		N	L	M	H	UH		
1	20	0	1	20	0	0	0	0	1	N
2	20	0	1.25	15	5	0	0	0	1	N
3	20	0	2.15	6	5	9	0	0	2	L
4	20	0	1.10	18	2	0	0	0	1	N
5	20	0	1.75	11	3	6	0	0	2	L
6	20	0	2.25	7	1	12	0	0	2	L
7	20	0	1.25	17	1	2	0	0	1	N
8	20	0	1.70	11	6	1	2	0	2	L
9	20	0	2.75	1	5	12	2	0	3	M
10	20	0	1.4	15	3	1	1	0	1	N
11	20	0	1.9	7	8	5	0	0	2	L
12	20	0	1.6	12	5	2	1	0	2	L
13	20	0	2.5	2	9	6	3	0	3	M

Expert's responses were distributed between N, L, and M in rule 5. Although eleven experts have decided that the output should be N, the mean value of all experts mapped the output to be L. Examining this rule demonstrates that having M in both action severity and user context should at least make the output as L. This scenario was the same for both rule 8 and rule 12. Although the

majority of experts decided to classify the rule as N, the mean value of all experts mapped the output to be L. however, examining this rule indicates that having S in the resource sensitivity should make the output to be L even if all other risk factors were L. For rule 12, since values of user context and risk history were M, so the appropriate output should be L. Rule 11 with similar to rule 12 in which values of risk history and user context were M, so experts decided to classify the output to be L.

In addition, experts decided to classify both rule 9 and rule 13 as M. For rule 9, although both user context and risk history were L, having S in the resource sensitivity and M in the action severity made experts to classify the output of this rule as M. Moreover, for rule 13, having M in three risk factors made experts classify the output of this rule as M.

Table 5.9 represents validation of fuzzy rules when the output was L. The output of only six rules were the same before and after expert validation. These rules involve rule 15, 20, 22, 24, 25, and 27. Examining these rules indicates that experts decided to classify the output of these rules as L due to having L in two risk factors regardless of values of other risk factors. On the other hand, the majority of experts have classified the output of nine rules as M. These rules include rule 14, 16, 17, 18, 19, 21, 23, 26, and 28. For rule 14, ten experts have decided to classify the output of this rule as M, while the responses of other ten experts were distributed equally between L and H. Investigating this rule demonstrates that M will be the appropriate output due to having H in the action severity and S in the resource sensitivity.

Table 5.9: Validation of fuzzy rules when the output was L

Rule No	Number of Experts		Mean	Experts Responses					Mapped Category	Rule Output
	Valid	Missing		N	L	M	H	UH		
14	20	0	3	0	5	10	5	0	3	M
15	20	0	2.45	1	11	6	2	0	2	L
16	20	0	3	0	4	13	2	1	3	M
17	20	0	3.25	0	5	6	8	1	3	M
18	20	0	2.5	1	9	9	1	0	3	M
19	20	0	2.75	0	7	11	2	0	3	M
20	20	0	2.35	1	13	4	2	0	2	L
21	20	0	2.6	0	11	6	3	0	3	M
22	20	0	2.3	1	13	5	1	0	2	L
23	20	0	2.75	0	10	5	5	0	3	M
24	20	0	2.25	1	16	0	3	0	2	L
25	20	0	2.1	1	17	1	1	0	2	L
26	20	0	2.7	0	8	10	2	0	3	M
27	20	0	2.25	0	17	1	2	0	2	L
28	20	0	2.75	0	7	11	2	0	3	M

For rule 16 and 17, having HS in the resource sensitivity made most experts to classify the output of these rules as M especially with M and H in the action severity respectively, so the access will be to highly sensitive data with a sever action which makes M is appropriate output specifically with having L in both risk history and user context. This scenario was the same for rule 18 and 19 in which the resource sensitivity was S and the user context was M which demonstrates why M is the appropriate output in this situation. For rule 21 and 23, although the majority of experts have decided

to classify the output of these rules to be L due to having NS in the resource sensitivity, the mean value of all expert mapped the output to be M. For rules 26 and 28, most experts have decided that M is the appropriate output for these rules. Examining these rules demonstrates that even if the data is insensitive, having H in the risk history should be considered since the malicious user always comes with malicious actions.

Table 5.10 represents validation of fuzzy rules when the output was M. The output of thirteen rules was identical before and after expert validation. The majority of experts have decided that M is the appropriate output for these set of rules as well as the mean value of all experts mapped the output to be M. Investigating these rules demonstrates that the resource sensitivity and risk history were the dominant risk factors for most experts to make their decision for these set of rules. Having HS or S in the resource sensitivity and H or M in the risk history made most experts to classify the output of these rules to be M.

On the other hand, security experts classified the output of six rules to be H. These rules involve rule 29, 31, 32, 37, 39 and 43. For rule 29, fourteen experts have decided that H is the appropriate output for this rule. This is due to having H in the action severity and S in the resource sensitivity which should be characterized as high risk especially with having M in the user context. For rule 31 and 32, most experts have decided that the suitable output for these rules is H. This is due to having HS in the resource sensitivity and M and H in the action severity respectively.

Table 5.10: Validation of fuzzy rules when the output was M

Rule No	Number of Experts		Mean	Experts Responses					Mapped Category	Rule Output
	Valid	Missing		N	L	M	H	UH		
29	20	0	3.7	0	0	6	14	0	4	H
30	20	0	3.3	0	0	14	6	0	3	M
31	20	0	3.75	0	0	6	13	1	4	H
32	20	0	3.8	0	0	6	12	2	4	H
33	20	0	3.25	0	0	15	5	0	3	M
34	20	0	3.05	0	0	19	1	0	3	M
35	20	0	3.4	0	0	12	8	0	3	M
36	20	0	3.1	0	0	18	2	0	3	M
37	20	0	3.65	0	0	8	11	1	4	H
38	20	0	3.25	0	1	13	6	0	3	M
39	20	0	3.8	0	0	4	16	0	4	H
40	20	0	2.9	0	2	18	0	0	3	M
41	20	0	3.25	0	0	15	5	0	3	M
42	20	0	3.2	0	0	16	4	0	3	M
43	20	0	3.5	0	0	10	10	0	4	H
44	20	0	3.4	0	1	10	9	0	3	M
45	20	0	3.05	0	2	15	3	0	3	M
46	20	0	3.45	0	1	11	6	2	3	M
47	20	0	3	0	1	18	1	0	3	M

Similarly, in rule 37 and 39, the majority of experts have classified the output of these rules to be H. This is because of having S and HS in the resource sensitivity, and M and H in the action severity respectively. In rule 43, half of the IoT security experts have classified the output of this rule to be

M, while the other half have decided that H is the appropriate output for this rule. However, the mean value of all experts mapped the output to be H.

Table 5.11 shows validation of fuzzy rules when the output was H. The output of twelve rules was identical before and after expert validation. Almost all experts have decided that H is the appropriate output for these twelve rules as well as the mean value of all experts mapped the output to the same result. In some rules, all twenty experts have decided that H is the appropriate output as in rule 48 and 56. Examining these rules demonstrates that experts have decided to classify the output of a rule to be H only if two risk factors were H or H and S and with at least M in other risk factors.

On the other hand, there was only one rule (rule 54) that most experts have decided that UH is the appropriate output for this rule. This is due to having H in the action severity and HS in the resource sensitivity. So, the system will be in real danger especially the values of the user context and risk history associated with this rule were M which indicates that the requesting user intends to perform malicious actions on highly sensitive data.

Table 5.11: Validation of fuzzy rules when the output was H

Rule No	Number of Experts		Mean	Experts Responses					Mapped Category	Rule Output
	Valid	Missing		N	L	M	H	UH		
48	20	0	4	0	0	0	20	0	4	H
49	20	0	3.65	0	0	7	13	0	4	H
50	20	0	4.25	0	0	0	15	5	4	H
51	20	0	3.9	0	0	2	18	0	4	H
52	20	0	3.95	0	0	1	19	0	4	H
53	20	0	4.05	0	0	0	19	1	4	H
54	20	0	4.6	0	0	0	8	12	5	UH
55	20	0	3.75	0	0	6	13	1	4	H
56	20	0	4	0	0	0	20	0	4	H
57	20	0	4.15	0	0	1	15	4	4	H
58	20	0	3.6	0	0	8	12	0	4	H
59	20	0	3.8	0	0	4	16	0	4	H
60	20	0	3.95	0	0	1	19	0	4	H

Table 5.12 represents validation of fuzzy rules when the output was UH. The output of seventeen rules was identical before and after expert validation. Also, the mean value of all experts mapped the output to the same result. In rules 63, 64, 65, 72, 73, 78, 80 and 81, all twenty experts have decided that UH is the appropriate output. Examining these set of rules demonstrates that experts classified the output to be UH if three risk factors were H regardless of the value of the fourth risk factor.

Table 5.12: Validation of fuzzy rules when the output was UH

Rule No	Number of Experts		Mean	Experts Responses					Mapped Category	Rule Output
	Valid	Missing		N	L	M	H	UH		
61	20	0	4.8	0	0	0	4	16	5	UH
62	20	0	4.7	0	0	0	6	14	5	UH
63	20	0	5	0	0	0	0	20	5	UH
64	20	0	5	0	0	0	0	20	5	UH
65	20	0	5	0	0	0	0	20	5	UH
66	20	0	4.7	0	0	1	4	15	5	UH
67	20	0	4.25	0	0	3	9	8	4	H
68	20	0	4.75	0	0	0	5	15	5	UH
69	20	0	4.9	0	0	0	2	18	5	UH
70	20	0	4.9	0	0	0	2	18	5	UH
71	20	0	4.75	0	0	0	5	15	5	UH
72	20	0	5	0	0	0	0	20	5	UH
73	20	0	5	0	0	0	0	20	5	UH
74	20	0	4.3	0	0	2	10	8	4	H
75	20	0	4.3	0	0	1	12	7	4	H
76	20	0	4.45	0	0	1	9	10	4	H
77	20	0	4.9	0	0	0	2	18	5	UH
78	20	0	5	0	0	0	0	20	5	UH
79	20	0	4.85	0	0	0	3	17	5	UH
80	20	0	5	0	0	0	0	20	5	UH
81	20	0	5	0	0	0	0	20	5	UH

On the other hand, experts decided to classify the output of four rules to be H. These rules involve rule 67, 74, 75, and 76. For rule 67, although only nine experts have decided to classify the output of this rule to be H, the mean value of all experts mapped the output to be H. Experts have decided that H is the appropriate output due to having L in both action severity and user context. For rule 74 and 75, the majority of experts have decided that H is the suitable output for these rules. This is because the resource sensitivity value associated with these rules was NS. So, if the data is not sensitive, then the appropriate risk output should be H even if all the other three risk factors were H. For rule 76, although ten experts have decided that UH is the appropriate output for this rule, the mean value for all experts mapped the output to H. Having L in the action severity should make H to be the appropriate output for this rule.

5.3.3.3 Implementation of Fuzzy Rules Using MATLAB

After all fuzzy rules were validated by twenty IoT security experts, fuzzy rules were implemented using the rule editor of the MATLAB fuzzy logic toolbox, as shown in Figure 5.10. The rule editor is used to construct fuzzy rules statements automatically. It is flexible, in which a rule can be added, edited, or deleted easily (MathWorks, 2016). All the rules had the same weight and the connection type was logical AND. Some studies have suggested deleting the rules that are covered by other rules to improve performance. However, the main target of the proposed risk estimation technique is to provide a precise and accurate risk value for each access request. Indeed, deleting some rules can improve the performance but the accuracy is the main concern in which removing some fuzzy rules will decrease the accuracy of resultant risk values. There are many studies referring that reducing the

number of rules will lower the system accuracy (Maksimovic et al., 2013; Seguí et al., 2013). In addition, the proposed risk estimation technique uses 81 rules which is not large compared to large systems involving multiple inputs with hundreds of fuzzy rules. Therefore, all 81 rules are used to implement the proposed fuzzy risk estimation technique.

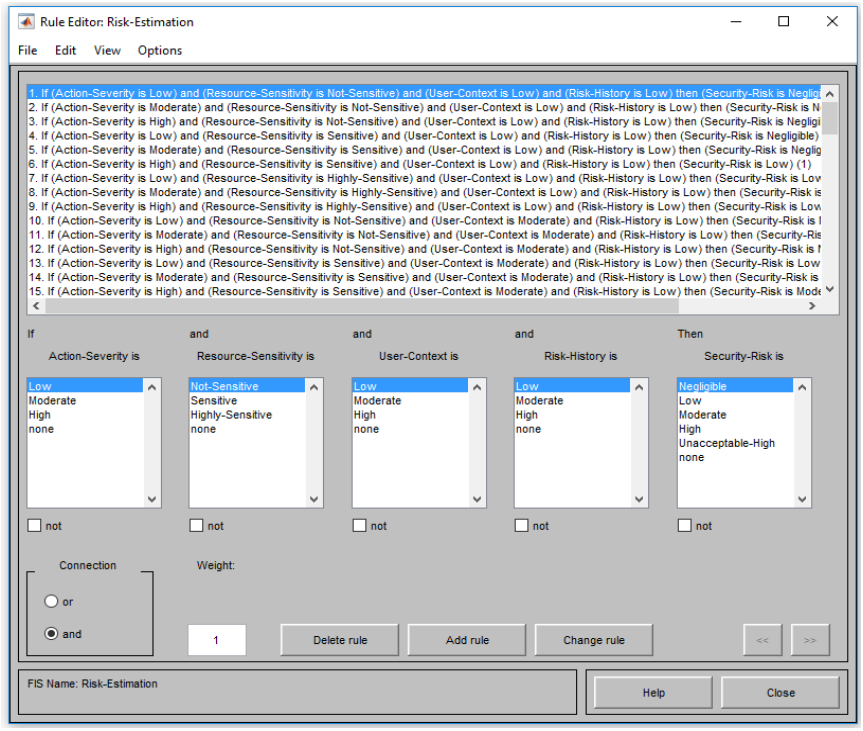


Figure 5.10: MATLAB rule editor to build fuzzy rules

In addition, the rule viewer shows a roadmap of the fuzzy inference process, as shown in Figure 5.11. It uses fuzzy rules that have been implemented using the rule editor. The first four columns of the rule viewer show MFs referenced by the IF-part of each rule of the four risk factors. The fifth column shows the MF of the output risk referenced by the THEN-part of each rule (Mathworks, 2016). Each rule is a row and each column represents risk input factors and output risk in the rule viewer. Rule numbers are displayed on the left of each row. In addition, when values of action severity, resource sensitivity, user context, and risk history were 0.5, the value of the output risk was 0.6 as shown in Figure 5.11.

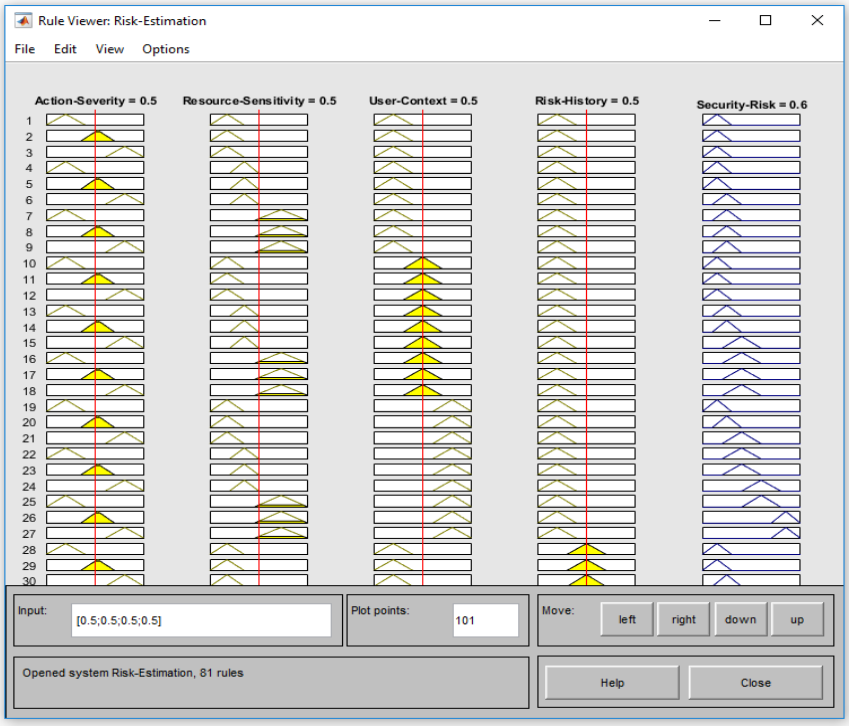


Figure 5.11: MATLAB rule viewer to show the fuzzy inference process

5.3.4 Aggregation of Output Rules

Rule aggregation is the process of combining outputs of all fuzzy rules. In other words, MFs of all fuzzy rules are combined into a single fuzzy set via rule aggregation (Li et al., 2013). MATLAB has three built-in rule aggregation operators; max, probor, and sum (Mathworks, 2016).

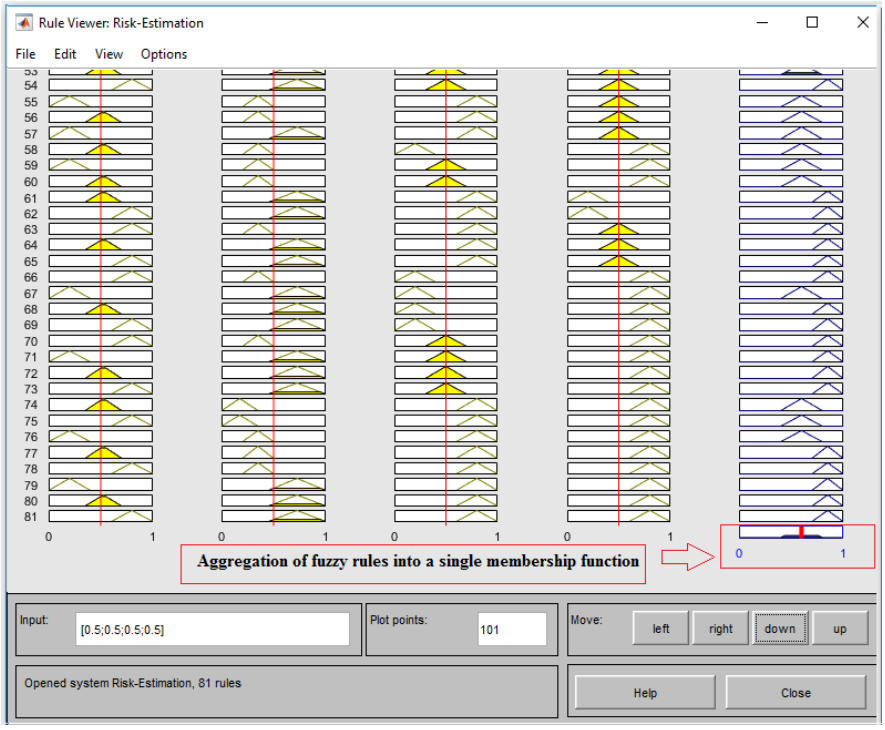


Figure 5.12: MATLAB rule viewer to show aggregation of rules using the max operator

In this research, the max (maximum) aggregation operator is used to combine the output of 81 rules into one fuzzy set. There is no superior operator than others. However, with the availability of a dataset, the try-and-error method can be used to select the appropriate aggregation operator. In MATLAB rule viewer, as shown in Figure 5.12, the aggregation occurs down the fifth column, and the resultant aggregate plot is shown in the single plot appearing in the lower right corner of the plot field.

5.3.5 Defuzzification

The final step of implementing the proposed fuzzy risk estimation technique is the defuzzification. The risk output has to be a crisp number. The most popular defuzzification method; the centroid method, was selected to be the appropriate defuzzification method since it provides a unique decision value between zero and one (Fernández et al., 2014). The defuzzified output value can be shown in Figure 5.12. It is represented by the thick red line passing through the aggregated fuzzy set, which indicate the defuzzied value that resulted from the aggregation of fuzzy sets.

5.3.6 GUI for Risk Estimation Process

The proposed risk estimation technique was implemented using MATLAB fuzzy logic toolbox. However, to provide the output risk value with an easy-to-use user interface, a Graphical User Interface (GUI) was created to show the estimated risk value and the access decision for a certain input combination. Therefore, when the values of risk input factors are known, the output risk value can be estimated, and the access decision can be determined.

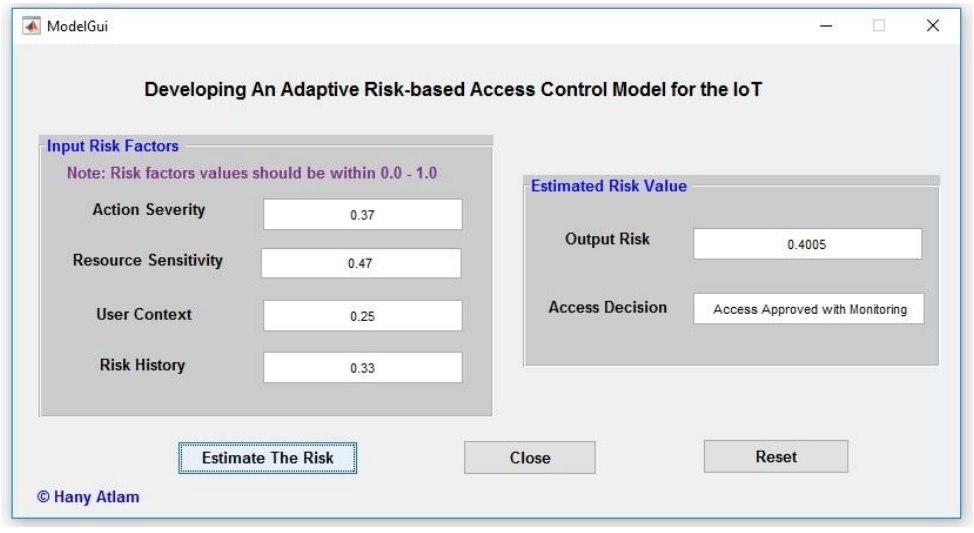


Figure 5.13: Using GUI to show input and output of the proposed risk estimation technique

For example, if there is an access request that involves an action severity rated as 37%, data sensitivity rated as 47%, the risk from user context features rated as 25%, and the requester risk history rated as 33%. Then, the estimated risk value associated with this access request is rated as

about 40%, which will grant access with monitoring user activities during the access session. The input and output risk values of this scenario can be shown in Figure 5.13.

5.4 Validation of Acceptable Risk by Experts

After estimating the security risk value associated with the access request, the next step is to compare the estimated risk value with acceptable risk values to determine whether to grant or deny access. Providing acceptable risk values for each application is very difficult to determine. Although most related risk-based access control models suggested using a threshold risk value to determine the access decision, they did not provide any details about how to determine this threshold risk value for different applications, especially in the IoT context.

As discussed earlier in section 4.3.1, this research proposed three risk decision bands to provide the access decision for each access request: allow, allow with risk monitoring and deny. However, determining the appropriate values for each band is very difficult. Therefore, twenty IoT security experts from inside and outside the UK were asked to determine the best values for each risk decision band using four open-ended questions. The researcher suggested certain values for each risk decision band and asked IoT security experts either to confirm these values or suggest new values for each risk decision band. The researcher suggested to use values 0.0 – 0.25 for the allow band, 0.26 – 0.7 for the allow with risk monitoring band, and 0.71 – 1.0 for the deny band. In addition, experts were asked to suggest any other decision bands regarding their expertise. Experts' responses were summarized as shown in Table 5.13.

Many of the experts have acknowledged that three bands are applicable for the IoT system. While others have recommended using a fourth risk decision band. Eight experts (E3, E5, E11, E12, E14, E15, E16, and E20) have confirmed values suggested by the researcher for each risk decision band and decided that no other decision bands are required. They decided that these values reflect the fact that most access to system resources will be done through the allow with risk monitoring band, which will be realized by having such large range of risk values from 0.26 – 0.7.

The other experts, specifically E2 and E13, recommended using different values for risk decision bands. Although expert E2 confirmed values of the deny band, he/she suggested to use values 0.0 – 0.1 and 0.11 – 0.7 for the allow and allow with risk monitoring band respectively. He/she added that values of allow band should be narrow to grant only the device owner or the user with very small risk value through this band. For expert E13, although he/she has confirmed values of the allow band that are suggested by the researcher, he/she recommend extending the allow with risk monitoring band to involves values from 0.26 – 0.8.

Table 5.13: Experts' responses to determine the best values for risk decision bands

Expert No	Allow Band	Allow with Risk Monitoring Band	Deny Band	Other Suggested Bands
E1	0.0 – 0.25	0.26 – 0.5	0.76 – 1.0	Use a fourth band from 0.51 – 0.75
E2	0.0 – 0.1	0.11 – 0.7	0.71 – 1.0	No other bands are required
E3	0.0 – 0.25	0.26 – 0.7	0.71 – 1.0	No other bands required
E4	0.0 – 0.25	0.26 – 0.5	0.71 – 1.0	Use a fourth band from 0.51 – 0.7
E5	0.0 – 0.25	0.26 – 0.7	0.71 – 1.0	No other bands required
E6	0.0 – 0.25	0.26 – 0.5	0.71 – 1.0	Use a fourth band from 0.51 – 0.7
E7	0.0 – 0.15	0.16 – 0.5	0.71– 1.0	Use a fourth band from 0.5 – 0.7
E8	0.0 – 0.25	0.26 – 0.5	0.76 – 1.0	Use a fourth band from 0.51– 0.75
E9	0.0 – 0.25	0.26 – 0.5	0.71 – 1.0	Use a fourth band from 0.51– 0.7
E10	0.0 – 0.25	0.26 – 0.5	0.71 – 1.0	Use a fourth band from 0.5 – 0.7
E11	0.0 – 0.25	0.26 – 0.7	0.71 – 1.0	No other bands are required
E12	0.0 – 0.25	0.26 – 0.7	0.71 – 1.0	No other bands are required
E13	0.0 – 0.25	0.26 – 0.8	0.8 – 1.0	No other bands are required
E14	0.0 – 0.25	0.26 – 0.7	0.71 – 1.0	No other bands are required
E15	0.0 – 0.25	0.26 – 0.7	0.71 – 1.0	No other bands are required
E16	0.0 – 0.25	0.26 – 0.7	0.71 – 1.0	No other bands are required
E17	0.0 – 0.25	0.26 – 0.5	0.71 – 1.0	Use a fourth band from 0.51 – 0.7
E18	0.0 – 0.25	0.26 – 0.5	0.71 – 1.0	Use a fourth band from 0.51 – 0.7
E19	0.0 – 0.25	0.26 – 0.5	0.81– 1.0	Use a fourth band from 0.51 – 0.8
E20	0.0 – 0.25	0.26 – 0.7	0.71 – 1.0	No other bands are required

On the other hand, ten security experts (E1, E4, E6, E7, E8, E9, E10, E17, E18, and E19) recommended new changes to the suggested values made by the researcher. They confirmed values of the allow band suggested by the researcher except for expert E7 who suggested to use values 0.0 - 0.15 for the allow band. For the deny band, the majority of ten experts confirmed values suggested by the researcher. There were suggestions from experts E1, E8, and E19 in which experts E1 and E8 have suggested assigning values 0.76 – 1.0 for the deny band, whereas expert E19 suggested using values from 0.81–1.0 for the deny band. For the allow with risk monitoring band, ten experts recommended using values 0.26 – 0.5 for this band. They added although most access will be through this band; this band should be divided into two bands with different monitoring measures. They suggested using values 0.51 – 0.7 as the fourth band with more restrictions on the access.

5.5 Validation of Proposed Risk Model by Experts

The need for access control models that provide more flexibility than static approaches has been pointed out repeatedly in recent years especially after the IoT appearance. The risk-based access control model provides a dynamic way to make the access decision. It uses the risk associated with the access request as a criterion to determine the access decision. This research provided a dynamic

and adaptive risk-based access control model that uses real-time and contextual information to provide the access decision.

Twenty IoT security experts from inside and outside the UK were asked to validate the proposed risk-based access control model using open questions through an interview. The interview questions can be shown in Appendix A. The first question was about experts' feedback regarding the proposed risk-based access control model. Most experts have shown their interest regarding the proposed risk-based model. They confirmed it will be valuable as various IoT applications require such dynamic access control models. For example, Expert E2 stated, "using contextual data from the IoT environment will be a good way to provide dynamic access". Also, Expert E5 said, "the idea of making a dynamic access control model is very interesting". Moreover, Expert E14 stated, "it will be very attractive for commercial". Similarly, Expert E7 added "being able to determine risk value of contextual features will be interesting and difficult as well".

In addition, experts have confirmed that the proposed risk factors are appropriate for the IoT context. For instance, Expert E5 stated "risk factors depend on the environment, I think these factors are appropriate for the IoT, but they are more generalized". Also, Expert E17 stated "they are appropriate but what if I do not have a risk history. You need to take this in your mind". Similarly, Expert E9 and Expert E12 have mentioned the issue associated with the risk history. In addition, Expert E16 added, "they appropriate but how contextual features will be collected from the environment, as it depends on which features you are able to collect". Although the majority of experts confirmed that the proposed risk factors are appropriate for various IoT applications, some experts suggested that more risk factors can be extracted depending on the IoT application context. In other words, they advised to work on a single IoT application and identify different features and factors for this application. However, the main objective of this research is to provide a dynamic risk-based access control model that can work with various IoT applications.

When experts were asked about the ranking of four risk factors in term of the importance in the IoT system, the majority of experts expressed that all risk factors used in the proposed model are important. They considered the resource sensitivity and the risk history are the most effective risk factors, then the action severity and the user context. For example, Expert E7 stated, "definitely, resource sensitivity is the most important, then user context, action severity and risk history". Also, Expert E1 considered action severity is most important, then resource sensitivity, user context and risk history. On the other hand, some experts such as Expert E3, E4, E9, E19, and E20 suggested that the ranking of risk factors should be regarding a specific application. For instance, for sensitive applications, the resource sensitivity and action severity would be the most effective. After implementing the proposed risk estimation approach using the fuzzy logic system with expert judgment, results demonstrated that resource sensitivity is the dominant risk factor that decides most access in which the higher the resource sensitivity, the higher the output risk.

5.6 Cold Start Problem

Cold start is one of the major issues in computer-based information systems. It refers to the problem of obtaining sufficient information about users or items to draw inferences or results (Quijano-Sánche et al., 2012). For this research, the proposed risk-based access control model has four risk factors; user context, resource sensitivity, action severity and risk history. Each risk factor is used to estimate the overall risk value associated with the access request to make the access decision. The cold start problem appears when a user makes an access request for the first time, so there is no risk history to estimate the security risk value associated with the access request. Therefore, the proposed risk-based model cannot operate until collecting sufficient risk history values for system users, which will be very difficult in a dynamic environment such as IoT systems which accept new users almost every day.

Fuzzy rules of the proposed fuzzy risk estimation technique were implemented using four inputs/risk factors to estimate the risk value for each access request, so how the risk estimation technique can provide a result without having a risk history value. To resolve this issue, one can say, if there is no risk history value, then suppose the risk history value is the minimum fuzzy value, which is zero. However, if the risk history value is assumed to be zero, this means that the requesting user is a trusted one with very low-risk history value. Indeed, the malicious user might be the first time to access the system, and if the risk history value is assumed to be zero, the system will consider the malicious user as a trusted one. Therefore, this solution will not work.

Another one can say, if there is no risk history value, then suppose that the risk history value to be the maximum fuzzy value, which is one. However, if the risk history is assumed to be one, this means that the requesting user is definitely a malicious user with very high-risk history. Therefore, the owner of an IoT device who will access the system for the first time will be considered as a malicious user. Consequently, this solution will not work either.

The proposed solution to overcome the cold start problem is to allow the risk estimation approach to estimate the security risk value when there is no risk history. In other words, when there is no risk history value associated with the requesting user, the proposed risk estimation approach should use only three risk factors (user context, resource sensitivity, and action severity) to estimate the overall risk value associated with the access request. However, fuzzy rules are built using four risk factors. Therefore, the proposed fuzzy risk estimation technique should be modified by adding another 27 fuzzy rules to include input combinations of only three risk factors. These fuzzy rules are created in the same way as discussed earlier in which fuzzy rules are first created using the information collected from the literature with the researcher experience. Then, fuzzy rules are validated using IoT security experts.

5.6.1 Building Fuzzy Rules of Cold Start

When there is no risk history value, new fuzzy rules need to be added to represent relationships between only three risk factors (user context, resource sensitivity, and action severity) to resolve the cold start issue and provide an output risk value for each access request. Since three risk factors have three MFs each, the number of fuzzy rules that need to be added will be 3*3*3=27 rules. Hence, the total number of fuzzy rules of the proposed fuzzy risk estimation technique will be 81+27=108. The output of 27 fuzzy rules has been created using the information collected from the literature with the researcher experience, as shown in Table 5.14. Notations of input and output risk linguistic expressions can be shown in Table 5.2. As discussed earlier in section 5.3.3.1, there were some logical rules and information from the literature that helped the researcher to build fuzzy rules.

Table 5.14: Fuzzy rules of cold start

Rule No	Risk Factors			Output Risk
	Action Severity	Resource Sensitivity	User context	
82	L	NS	L	N
83	M	NS	L	N
84	H	NS	L	N
85	L	S	L	L
86	M	S	L	L
87	H	S	L	L
88	L	HS	L	M
89	M	HS	L	M
90	H	HS	L	M
91	L	NS	M	N
92	M	NS	M	L
93	H	NS	M	L
94	L	S	M	M
95	M	S	M	M
96	H	S	M	H
97	L	HS	M	H
98	M	HS	M	UH
99	H	HS	M	UH
100	L	NS	H	L
101	M	NS	H	M
102	H	NS	H	H
103	L	S	H	UH
104	M	S	H	UH
105	H	S	H	UH
106	L	HS	H	UH
107	M	HS	H	UH
108	H	HS	H	UH

The output of four rules was classified to be N. These rules include rule 82, 83, 84, and 91. The resource sensitivity was the dominant risk factor in which having NS in resource sensitivity helped to classify the output as N especially with having L in user context regardless of values of action severity. In addition, six fuzzy rules were classified to be L. These rules include rule 85, 86, 87, 92,93, and 100.

Further, the output of six rules was classified to be M. These rules include rule 88, 89, 90, 94, 95, and 101. For this set of rules, the resource sensitivity was the dominant risk factor in which when its value increased to S and HS, the output became M regardless of values of both user context and action severity. This scenario changed for rule 101 as the resource sensitivity was NS. However, the output became M due to having H and M in the user context and action severity respectively. In addition, the output of three fuzzy rules including rule 96, 97, and 102 was classified to be H. Having H or H and S in two risk factors were the main reason to classify the output of these rules to be H. While the output of eight fuzzy rules was classified to be UH. These rules involve rule 98, 99, 103, 104, 105, 106, 107, and 108. The output of a fuzzy rule was classified to be UH if the resource sensitivity was S or HS and the other risk factor was H.

5.6.2 Validation of Cold Start Fuzzy Rules by Experts

As discussed earlier, there is no dataset in this research, so the best way to obtain accurate and precise fuzzy rules is by IoT security experts. Hence, experts were interviewed to validate fuzzy rules of cold start which were previously built using the information collected from the literature with the researcher experience. In this section, validating fuzzy rules by experts through the interview will be discussed. Initially, the interview design, experts sample size, and expert's information will be provided, then the results of the interview will be discussed.

5.6.2.1 Expert Interview Design

Since the main target of the interview is to validate fuzzy rules of the cold start problem, the interview was structured using closed questions in which the participant was given five choices to be the expected output for each rule as the following: Negligible; Low; Moderate; High and Unacceptable High. All the interviews were conducted on the campus of the University of Southampton in the expert's office and others in cafes. All interviews were conducted in the English language. Appendix B provides the interview questions. Before starting the interview, each participant was asked to sign a consent form after reading the participant information sheet that included all the necessary information, terms and conditions about the study. The University of Southampton Ethics Committee granted approval for this study under their reference number ERGO/FPSE/25091.

5.6.2.2 Interview Sample Size

Due to the problem of reaching a large number of experts, interviews have conducted with ten security experts at the University of Southampton. Attributes of experts who have involved in this study can be shown in Table 5.15. Most experts had at least 2- 5 years of experience. For experts who did not have knowledge about the fuzzy logic approach, the researcher has spent about 10 minutes to make sure the participant understood essential information about the fuzzy logic system.

Table 5.15: Attributes of security experts used to validate fuzzy rules of the cold start problem

Expert No	Job Description	Experience (Years)	Knowledge of Fuzzy logic	Knowledge of IoT Applications
E 1	Cybersecurity Lecturer	6 – 10	No	Smart city, smart home, and connected healthcare
E 2	Security Research Follow	2– 5	Yes	Smart city, connected car, and connected healthcare
E 3	Cybersecurity Lecturer	6 – 10	No	Connected industry, smart city, smart home, and smart agriculture
E 4	Security Research Follow	2– 5	Yes	Smart city and connected car
E 5	Security Research Follow	2 – 5	No	Smart city and connected healthcare
E 6	Security Research Follow	2 – 5	Yes	Smart city, connected car, smart home, and connected healthcare
E 7	Security Research Follow	Less than 2	Yes	Smart home and smart supply chain
E 8	Security Research Follow	6 – 10	Yes	Smart energy, smart city, and smart home
E 9	Security Research Follow	2 – 5	No	Connected car and smart home
E 10	Security Research Follow	2 – 5	Yes	Smart city, smart energy and smart home

5.6.2.3 Results of the Interview

As discussed earlier in section 5.3.3.2, the mean function was utilized to determine the output of each rule. To use expert's responses in the SPSS program, expert's responses have given ratings such that Negligible =1, Low=2, Moderate=3, High=4, and Unacceptable High=5. Therefore, the output of each fuzzy rule should be mapped to one of these five categories. An assumption was made such that any mean value lower than 0.5 will be mapped to the lower category and any mean value higher than or equal 0.5 will be mapped to the higher category.

As shown in Table 5.16, experts have confirmed the output of fuzzy rules that were built using the information collected from the literature with the researcher experience. The output of seventeen fuzzy rules was identical in both expert validation and literature review with the researcher experience. In these rules, all experts confirmed the same output for ten fuzzy rules. These fuzzy rules include rule 82, 88, 94, 95, 96, 97, 99, 105, 106, 107, and 108. On the other hand, the output of ten fuzzy rules was different from the one derived using the literature review with the researcher experience. These rules involve rule 83, 84, 87, 89, 90, 92, 93, 98, 100, and 103. For rule 83, the majority of experts decided to classify the output of this rule to be L especially with having M in the action severity. The same scenario was for rule 84 in which having H in the action severity made experts' responses to distribute between N, L and M, however, the mean value of all experts made the output to be L.

For rule 87, some experts suggested that having L in the user context should make M to be the appropriate output for this rule, while other experts decided to classify the output of this rule to be H due to having H in the action severity and S in the resource sensitivity. However, the mean value of

all experts made the output to be M. Similarly, in rule 89 and 90, most experts decided that H is the appropriate output for these rules. Examining these rules demonstrates that experts decided to classify the output to be H due to having HS in the resource sensitivity.

Table 5.16: Validation of fuzzy rules of the cold start by security experts

Rule No	Number of Experts		Mean	Experts Responses					Mapped Category	Rule Output
	Valid	Missing		N	L	M	H	UH		
82	10	0	1	10	0	0	0	0	1	N
83	10	0	1.6	4	6	0	0	0	2	L
84	10	0	2	4	2	4	0	0	2	L
85	10	0	2.2	0	8	2	0	0	2	L
86	10	0	2.3	0	7	3	0	0	2	L
87	10	0	3.2	0	3	2	5	0	3	M
88	10	0	3	0	0	10	0	0	3	M
89	10	0	3.6	0	0	4	6	0	4	H
90	10	0	3.7	0	0	4	5	1	4	H
91	10	0	1.4	6	4	0	0	0	1	N
92	10	0	2.5	0	5	5	0	0	3	M
93	10	0	2.9	0	1	9	0	0	3	M
94	10	0	3	0	0	10	0	0	3	M
95	10	0	3	0	0	10	0	0	3	M
96	10	0	4	0	0	0	10	0	4	H
97	10	0	4	0	0	0	10	0	4	H
98	10	0	4.4	0	0	0	6	4	4	H
99	10	0	5	0	0	0	0	10	5	UH
100	10	0	2.6	0	4	6	0	0	3	M
101	10	0	3.3	0	0	7	3	0	3	M
102	10	0	4.2	0	0	0	8	2	4	H
103	10	0	4.4	0	0	0	6	4	4	H
104	10	0	4.9	0	0	0	1	9	5	UH
105	10	0	5	0	0	0	0	10	5	UH
106	10	0	5	0	0	0	0	10	5	UH
107	10	0	5	0	0	0	0	10	5	UH
108	10	0	5	0	0	0	0	10	5	UH

For rule 92, half of the experts decided to classify the output of this rule to be L, while the other half decided to classify it to be M. However, the mean value of all experts made the output to be M. While for rule 93 and 100, the majority of experts decided that M is the appropriate output for these rules. For rule 98 and 103, most experts decided that H is the appropriate output for these rules due to having HS and S in the resource sensitivity respectively.

5.6.3 Implementing Fuzzy Rules of Cold Start

After validating fuzzy rules of the cold start problem by IoT security experts, the rule editor of MATLAB fuzzy logic toolbox was utilized to add these rules to the fuzzy model that implemented earlier, as shown in Figure 5.14. All the rules had the same weight and the connection type was logical AND.

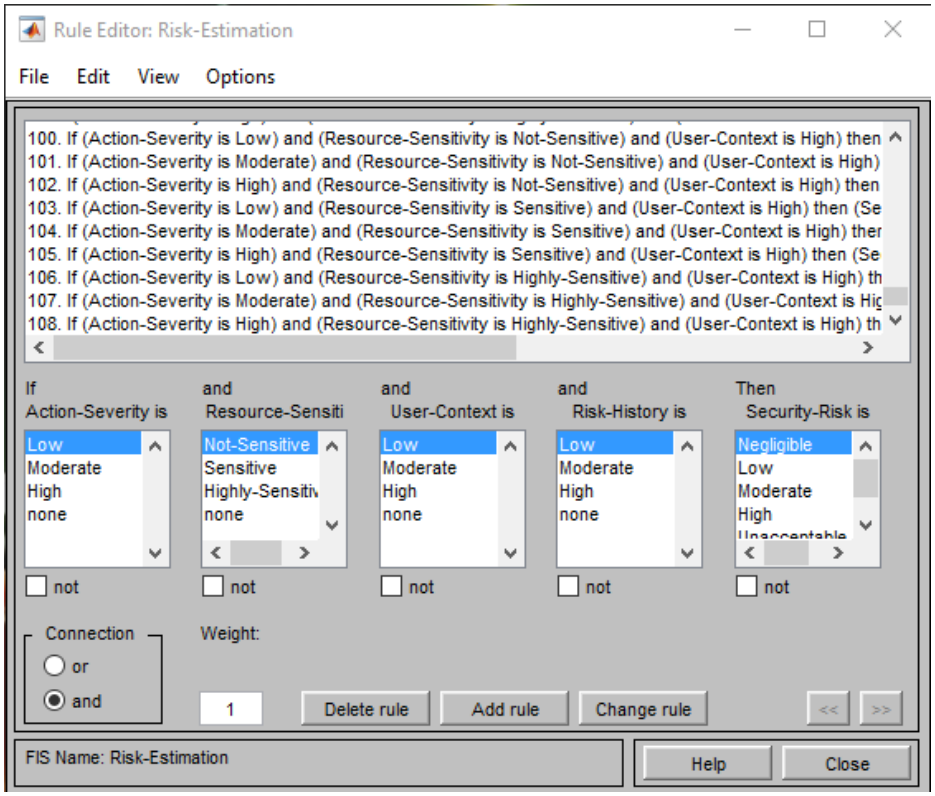


Figure 5.14: Adding validated fuzzy rules using MATLAB rule editor

After adding new fuzzy rules, the proposed risk estimation technique can work well now when there is no risk history associated with the requesting user. So, the output risk will be estimated using three risk factors: user context, resource sensitivity, and action severity. Then, the estimated risk value will be compared against risk decision bands to determine the access decision whether to grant or deny the access. For example, if there is an access request that involves an action severity rated as 25%, data sensitivity rated as 56%, the risk from user context features rated as 77%, and no risk history value. As shown in Figure 5.15, the proposed risk estimation approach provides an output risk of 0.85, which will reject access based on proposed risk decision bands. Therefore, the proposed risk estimation technique has overcome the cold start problem associated with the proposed risk-based access control model. This makes the proposed risk-based model productive and ready to provide the required functionality immediately and effectively without any prior adjustments.

Figure 5.15: Providing access decision without having a risk history value

5.7 Efficiency of The Fuzzy Model

The fuzzy logic system has demonstrated it can generate accurate and realistic values in assessing security risks in access control operations. As discussed in section 3.3, there are many advantages to use the fuzzy logic system with expert judgment to conduct the risk estimation process of the proposed risk-based access control model. However, it is not straight forward, it raises some issues.

First, there are multiple methods in each fuzzy stage such as MFs, defuzzification methods, and rule aggregation operators (Dubois & Yager, 1992). So, determining the most appropriate method regarding the IoT context is a major issue especially when there is no available dataset. Second, the scalability of the fuzzy logic system seems to be questionable. Fuzzy logic systems need a long period of time to estimate security risks especially when there is a large number of input parameters and hundreds of fuzzy rules. In addition, an access control system may need to serve hundreds or thousands of users. Therefore, a fuzzy inference-based access control system might be too computationally expensive (Ni et al., 2010). In the research due to the lack of datasets, fuzzy logic is the appropriate approach.

In this section, several experiments were carried out to evaluate the efficiency of the proposed fuzzy risk estimation technique to demonstrate the effect of changing fuzzy parameters such as MF, defuzzification method, and rule aggregation. In addition, the efficiency will be evaluated when increasing the number of access requests.

5.7.1 Experiment Setting

The objective of this set of experiments is to evaluate the efficiency of the proposed fuzzy risk estimation technique with different numbers of access requests. In addition, the efficiency of the proposed risk estimation technique will be evaluated while changing fuzzy parameters to decide the

best and efficient parameters. All experiments and measurements are coded using MATLAB on Intel(R) Core (TM) i7-2600, 3.40 GHz CPU with 16 GB RAM running Windows 10.

5.7.2 Experimental Results

This section provides experimental results of different experiments carried out to evaluate the efficiency of the proposed fuzzy risk estimation approach with different number of access requests and when changing fuzzy parameters including MF, defuzzification method, and rule aggregation operator.

5.7.2.1 Scale of Access Requests

The first experiment evaluates the response time of the proposed fuzzy risk estimation approach when changing the number of access requests. This experiment was carried out using the triangular MF as the appropriate MF and the centroid method as the suitable defuzzification method. The response time of the proposed risk estimation technique was estimated when increasing the number of access requests from 1000 to 250000, as shown in Figure 5.16. To provide a consistent response time, this experiment was carried out five times and the mean value was utilized to represent the mean response time with each number of access requests.

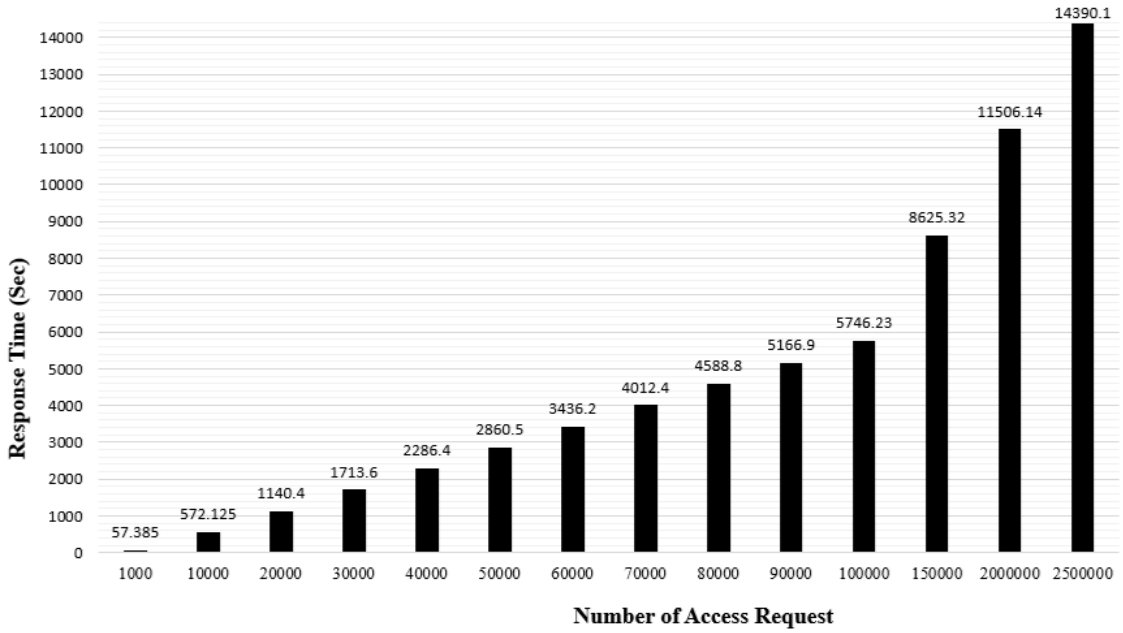


Figure 5.16: Response time when increasing number of access requests

This chart represents a linear relationship between the response time and the number of access requests in which the larger the number of access requests, the higher the response time. It is a privilege for the proposed fuzzy risk estimation technique to estimate the security risk value for 1000 access requests in only 57.385 seconds. This demonstrates that the proposed risk estimation approach provides an efficient way to estimate security risks of access control operations in a timely manner.

In addition, the response time per access request of the proposed risk estimation technique was estimated with different number of access requests, as shown in Figure 5.17. Generally, the response time per access request for the proposed risk estimation technique is about 0.057 second, which demonstrates it can provide the estimated risk value for each access request in a very short time. As depicted in Figure 5.17, the time per access request at the start was quite large, then it started to decrease until reaching its lowest value at 20000 access requests. This decrease occurred due to the fact that the system became familiar with the process until a certain stage. Then, it follows a straight line in which increasing number of access requests lead to increasing the response time per request.

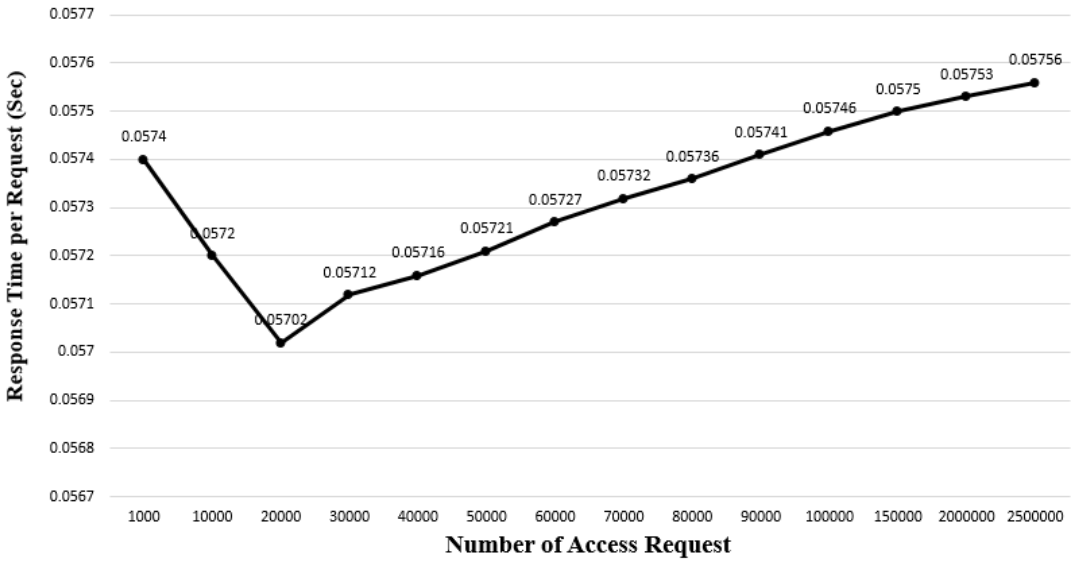


Figure 5.17: Response time per request when increasing number of access requests

5.7.2.2 Complexity of MFs

One of the important parameters to build an effective fuzzy logic system is to select the appropriate MF to represent the degree of membership of a certain value in a fuzzy set. When there is no available dataset to determine the best MF, the alternative way is to use the try-and-error method to determine the most efficient MF in term of the response time. The aim of this experiment is to verify the impact of different MFs. As discussed earlier in section 3.4.2, there are eleven MFs that can be used to define how each value is mapped to a membership value for the proposed fuzzy risk estimation technique.

In this experiment, the proposed risk estimation technique was implemented using eleven MFs to test the response time for each MF. All remaining fuzzy parameters of the proposed risk estimation technique held fixed such as defuzzification method, aggregation method, and fuzzy rules. To test the response time of different MFs, a fixed number of access requests (1000) was used. To provide a consistent response time for each MF, this experiment was carried out five times and the mean value was utilized to represent the mean response time for each MF.

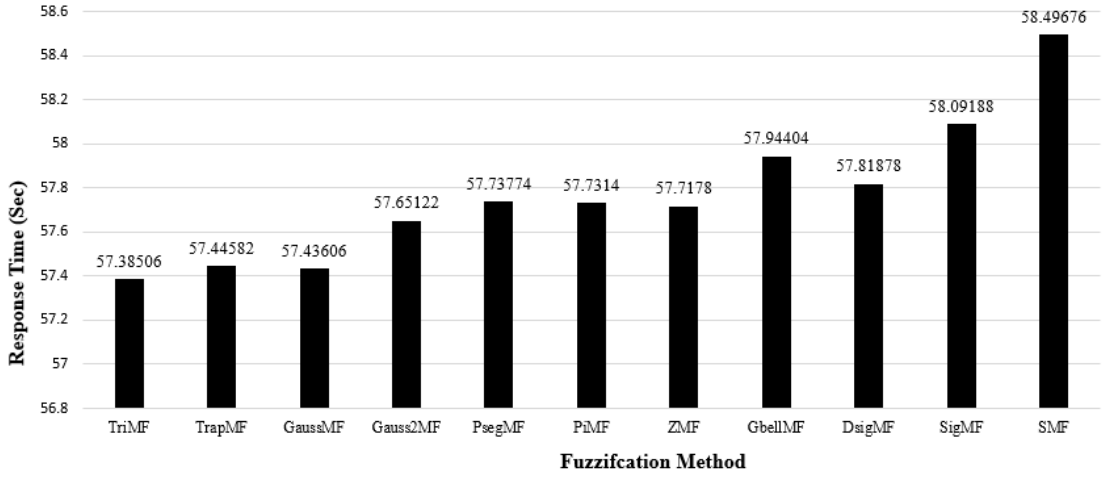


Figure 5.18: Response time of different fuzzification methods when applying 1000 access requests

The results of this experiment can be shown in Figure 5.18. As expected, there were different response time for each MF. TriMF produced the lowest response time, while SMF produced the highest response time. The difference in response time between various MFs was quite small, but with increasing number of access requests, this difference will be much higher.

To compare and determine if the mean values of various MFs are different, one-way repeated measure ANOVA test was carried out. Analysis of Variance (ANOVA) is a common and robust statistical test that is used to compare the mean scores collected from different conditions or groups in an experiment (Singh et al., 2013). Since the main target is to examine processing time (one group) of different MFs, one-way repeated measure ANOVA test is suitable in this situation.

The mean values of different MFs were tested to determine if there is a difference between them. The response time for each MF was measured five times. Mauchly's Test of Sphericity tests the null hypothesis that the variances of the differences are equal. Hence, if Mauchly's Test of Sphericity is statistically significant ($p < 0.05$), the null hypothesis will be rejected, and the alternative hypothesis will be accepted in which the variances of the differences are not equal (Haverkamp & Beauducel, 2017).

Table 5.17: Mauchly's Test of Sphericity of fuzzification method

Within-Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[b]		
					Greenhouse-Geisser	Huynh-Feldt	Lower-bound
MF	.014	10.353	9	.432	.444	.775	.250

As the main objective is to determine if mean values of various MFs are different, five MFs including TriMF, PsegMF, GbellMf, SigMF, and SMF were used to analyse their mean using one-way repeated measure ANOVA test. These MFs were selected based on changes in mean values, as depicted in Figure 5.18. Five measurements for each MF were entered in the SPSS software and the result was

as shown in Table 5.17. Mauchly's test indicated that the assumption of sphericity had not been violated, $X^2(9) = 10.353$, $p = 0.432 > 0.05$.

The results demonstrated that using one-way repeated measure ANOVA, the mean scores of different MFs were statistically significant different ($F(4, 16) = 27.401$, $p = 0.0001$), as depicted in Table 5.18. In addition, looking at the pairwise comparisons between five MFs shows that there was a statistical significance difference in the response time between TriMF and PsegMF ($p=0.04<0.05$), TriMF and SigMF ($p=0.017<0.05$), TriMF and SMF ($p=0.002<0.05$), PsegMF and SMF ($p=0.004<0.05$), and GbellMF and SMF ($p=0.037<0.05$). In addition, there was no statistical significance difference between SigMF and SMF ($p=0.732>0.05$).

Table 5.18: Tests of within-subjects' effects of fuzzification method

		Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
MF	Sphericity Assumed	3.408	4	.852	27.401	.000	0.873
Error (MF)	Sphericity Assumed	.497	16	.031			

The results demonstrated that the TriMF produced the lowest processing time among other MFs, so it can provide better system performance. Therefore, it has been selected as the appropriate MF to implement the proposed fuzzy risk estimation technique.

5.7.2.3 Complexity of Defuzzification Methods

Defuzzification is a mapping from a space of fuzzy control actions defined over an output universe of discourse into a space of non-fuzzy control actions. The defuzzification technique is aimed to produce a non-fuzzy control action that best represents the possibility distribution of an inferred fuzzy control action (Liaw, 1994). As discussed earlier in section 3.4.5, MATLAB has five built-in defuzzification methods: centroid, bisector, MOM, LOM, and SOM.

The purpose of this experiment is to verify the impact of various defuzzification methods by estimating the response time. The response time of 1000 access request was estimated for each defuzzification method while other fuzzy parameters such as fuzzy rules, rule aggregation operator, and fuzzification method were kept fixed. This experiment was performed five times and the mean value was utilized to represent the mean response time for each defuzzification method to provide more consistent results, as depicted in Figure 5.19.

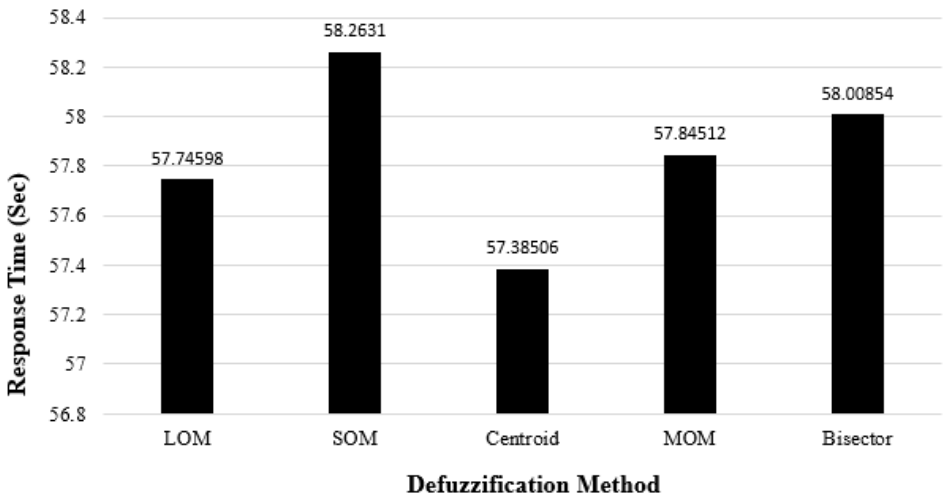


Figure 5.19: Response time of different defuzzification methods when applying 1000 access requests

In addition, a one-way repeated measure ANOVA test was utilized to determine if the mean values of various defuzzification methods are different. Mauchly's test indicated that the assumption of sphericity had not been violated, $X^2(9) = 9.641$, $p = 0.489 > 0.05$, as depicted in Table 5.19.

Table 5.19: Mauchly's Test of Sphericity of defuzzification methods

Within-Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[b]		
					Greenhouse-Geisser	Huynh-Feldt	Lower-bound
Defuzzification	.019	9.641	9	.489	.538	1.000	.250

Using one-way repeated measure ANOVA, the results demonstrated that mean scores of different defuzzification methods were statistically significant different ($F(4, 8.608) = 15.923$, $p = 0.00002$), as depicted in Table 5.20. In addition, looking at the pairwise comparisons between different defuzzification methods shows that there was a statistical significance difference in the response time between SOM and Centroid ($p = 0.044 < 0.05$), Centroid and MOM ($p = 0.008 < 0.05$), and Centroid and Bisector ($p = 0.048 < 0.05$). In addition, there was no statistical significance between Centroid and LOM ($p = 0.216 > 0.05$).

Table 5.20: Tests of within-subjects' effects of defuzzification methods

		Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Defuzzification	Sphericity Assumed	1.880	4	.470	15.923	.000	.799
Error (Defuzzification)	Sphericity Assumed	.472	16	.030			

The results demonstrated that the centroid is the most efficient defuzzification method as it produced the lowest processing time among other defuzzification methods, so it can provide better system performance. Therefore, it has been selected as the appropriate defuzzification method to implement the proposed fuzzy risk estimation technique.

5.7.2.4 Complexity of Rule Aggregation Operator

Rule aggregation is used to combine outputs of all fuzzy rules. In other words, MFs of all fuzzy rules are combined into a single fuzzy set via rule aggregation (Li et al., 2013). As discussed earlier in section 3.4.4, there are three main aggregation operators in MATLAB: max, probor, and sum. This experiment aims to determine the most efficient rule aggregation operator that can be utilized to implement the proposed fuzzy risk estimation technique.

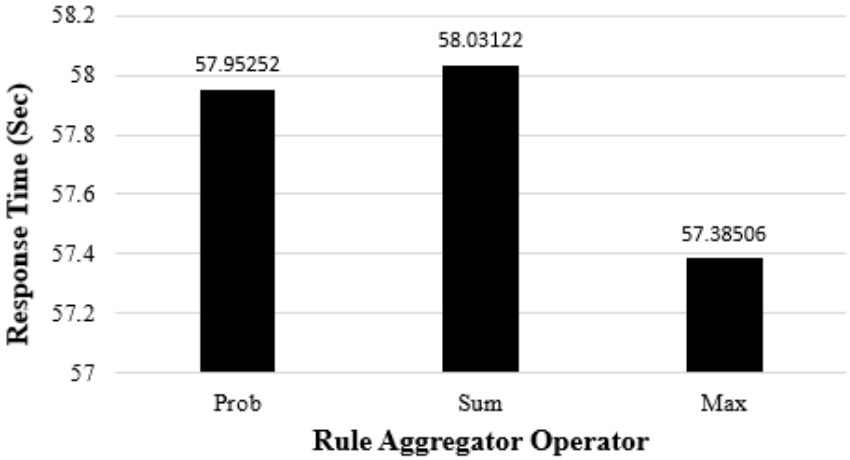


Figure 5.20: Response time of different rule aggregation operators when applying 1000 access requests

The response time of 1000 access request was estimated for each rule aggregation operator while other fuzzy parameters such as fuzzy rules, defuzzification method, and fuzzification method were kept fixed. This experiment was performed five times and the mean value was utilized to represent the mean response time for each rule aggregation operator to provide more consistent results, as shown in Figure 5.20. The results of this experiment demonstrated that the max operator produced the lowest response time, while the sum operator produced the highest response time.

In addition, a one-way repeated measure ANOVA test was utilized to determine if the mean values of various rule aggregation operators are different. Mauchly's test indicated that the assumption of sphericity had not been violated, $X^2(2) = 5.824$, $p = 0.054 > 0.05$. The results of one-way repeated measures ANOVA demonstrated that there was a statistically significant difference between mean scores of rule aggregator operators, ($F(2,8) = 31.151$, $p = .054$), as depicted in Table 5.21.

Table 5.21: Mauchly's Test of Sphericity of rule aggregator operators

Within-Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon[b]		
					Greenhouse-Geisser	Huynh-Feldt	Lower-bound
Aggregator	.143	5.824	2	.054	.539	.579	.500

In addition, looking at the pairwise comparisons between different rule aggregator operators shows that there was a statistically significant difference in the response time between Max and Prob ($p=0.0005<0.05$) and Max and Sum ($p=0.011<0.05$). Also, there was no statistical significance

between Prob and Sum (p=1.00>0.05). Using one-way repeated measure ANOVA, the results demonstrated that mean scores of different defuzzification methods were statistically significant different (F (4, 8.608) = 15.923, p =0.00002), as depicted in Table 5.22.

Table 5. 22: Tests of within-subjects' effects of rule aggregator operators

		Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Aggregator	Sphericity Assumed	1.243	2	.621	31.151	.000	.886
Error (Aggregator)	Sphericity Assumed	.160	8	.020			

The results demonstrated that the max operator produced the lowest processing time among other rule aggregator operators, so it can provide better performance. Therefore, it has been selected as the appropriate rule aggregator operator to implement the proposed fuzzy risk estimation technique.

5.7.3 Scalability Challenge in IoT

The number of IoT devices is growing rapidly. Predictions are made that by 2020, the number of IoT devices will reach or even exceed 50 billion (Evans, 2011). One of the major issues of the IoT system is scalability. It means the ability of the system to handle needs as they arise. It helps the system to work efficiently without performance issues due to system expansion. The main purpose of ensuring the scalability of the IoT system is to meet changing demands as the interest of people changes with time as well as environmental conditions (Gupta et al., 2017).

For this research, a risk-based access control model was proposed to handle flexibility issues in current access control models for the IoT system. To implement the proposed risk-based model, the fuzzy logic system with expert judgment was selected as the suitable risk estimation technique. However, the scalability of the fuzzy logic system seems to be doubtful since it requires a long period of time to estimate security risks of access control operations. An access control model for the IoT system is intended to serve hundreds or thousands of users. Based on the experimental results discussed earlier, the proposed risk estimation technique requires 57.385 seconds to estimate security risks of 1000 access request. This response time is very efficient for a small network of devices, but with the IoT system, there are thousands of devices per network. This number of IoT devices is constantly increasing which require to take the scalability of the proposed risk estimation technique into accounts and provide the required solution to resolve this issue. In addition, the proposed risk estimation technique lacks the ability to learn and cannot adjust themselves to a new environment.

Providing a scalable and able to learn risk estimation technique is one of the main objectives of this research. To achieve this target, the Artificial Neural Network (ANN) is proposed to be integrated with the fuzzy logic system. ANN is a low-level computational structure that performs well when dealing with raw data (Rezaei et al., 2014). It can learn to produce output even with incomplete

information, after being trained. In addition, it provides parallel processing capabilities that improve overall system efficiency (Cheng et al., 2016).

One of the solutions that integrate ANN with the fuzzy logic system is the NFS and ANFIS. NFS is an ANN technique, which is functionally equivalent to the fuzzy logic system. It combines the parallel computation and learning capabilities of ANN with the human-like knowledge representation and explanation abilities of the fuzzy logic system. As a result, ANN becomes more transparent, while fuzzy systems become capable of learning. In addition, the NFS can be trained to develop IF-THEN fuzzy rules and determine MFs for input and output variables of the system (Asogbon et al., 2016; Iranmanesh et al., 2009). In addition, the ANFIS is similar to neuro-fuzzy technique but works only with Sugeno FIS (Asogbon et al., 2016). For this research, to apply the neuro-fuzzy and ANFIS techniques, a dataset representing different risk factors values with corresponding output is required. This dataset can be created using the proposed fuzzy risk estimation technique that was implemented earlier.

The next two chapters will discuss the implementation of the proposed risk estimation technique using ANFIS and NFS to show how these solutions can solve learning and scalability issues associated with the proposed fuzzy model.

5.8 Summary

Chapter 5 has presented the implementation of the risk estimation process using the fuzzy logic system with expert judgment. It started by discussing the integration of the fuzzy logic system with expert judgment as the appropriate solution to provide accurate risk values of access control operations in the IoT system. One of the most effective ways to collect knowledge and expertise of experts is through the interview. Therefore, twenty IoT security experts were interviewed to validate the proposed risk-based access control model, validate fuzzy rules and determine acceptable risk values for proposed risk decision bands. This was followed by providing a step-by-step discussion of the implementation of the proposed risk estimation technique and how security experts have validated fuzzy rules and decided acceptable risk values of risk decision bands. In addition, one of the problems that may face the proposed risk-based model is the lack of information about the risk history of system users. This problem is called cold start. Therefore, a solution for the cold start problem was introduced by adding another 27 fuzzy rules using only three risk factors. To validate these rules, ten IoT security experts were interviewed. This was followed by providing a set of experiments to evaluate the efficiency of the proposed fuzzy risk estimation technique. These experiments were utilized to measure the response time with different number of access requests and to determine the most efficient MF, defuzzification method, and rule aggregation operator. This was followed by discussing scalability and learning issues of the proposed fuzzy risk estimation technique

and how adopting ANFIS and NFS can provide a good solution to these issues. The next chapter presents the implementation of the risk estimation process using the ANFIS technique.

Chapter 6: Implementation of Risk Estimation using ANFIS

This chapter provides a discussion of the implementation of the proposed risk estimation technique using the ANFIS. It starts by providing an overview of the ANFIS by highlighting the main objectives of the ANFIS in risk estimation techniques, ANFIS architecture, and ANFIS learning methods. Then, section 6.2 presents the implementation of the risk estimation technique using the ANFIS by showing different experimental results of training the ANFIS model using both hybrid and backpropagation learning methods at different number of epochs. Section 6.3 shows the effect of the training process on the MFs of the fuzzy logic system. The chapter closes by providing a summary of the main points discussed through the chapter and introduces the next chapter.

6.1 An Overview of ANFIS

ANFIS is a multilayer feed-forward network which utilizes ANN techniques and fuzzy reasoning to map inputs into an output. It is a FIS implemented in a framework of adaptive neural networks (Wang & Elhag, 2008). The ANFIS is considered the first integrated hybrid neuro-fuzzy model that uses the decomposition approach to extract rules at individual nodes within the ANN (Zanchettin et al., 2010). Then, the extracted rules are combined to construct global behaviour descriptions. According to Jang (1993), ANFIS is a type of adaptive networks that is equivalent to a FIS functionally. It uses training data to produce fuzzy rules and MF automatically. Typically, the ANFIS network comprises of connected nodes that depend on parameters that change constantly using the learning techniques to minimize possible errors. The most common learning techniques in the ANFIS are the backpropagation and hybrid learning methods (Jang, 1993).

The main objective of the ANFIS is to optimize parameters of the fuzzy logic system by applying a learning algorithm using input-output datasets. The parameter optimization is done in such a way that the error measured between the target and the actual output is minimized (Guney, 2008). The ANFIS has a high capability to adapt to its environment in the learning process. Therefore, it can be

used to adjust the MFs and reduce the error rate automatically to determine fuzzy rules of the fuzzy logic system.

The ANFIS combines the benefits of the fuzzy logic system and ANN into a single technique (Jang, 1993). It provides better results for applications where performance is more important than interpretation since the learning results may be difficult to interpret (Wu et al., 2011). According to Al-Hmouz et al., (2012), there are multiple advantages for the ANFIS, which include:

- Optimizes fuzzy rules to describe the behaviour of a complex system;
- Does not require prior human expertise;
- Easy to implement;
- Enables fast and accurate learning;
- Generates greater choice of MFs to use;
- Strong generalization abilities;
- Excellent explanation facilities through fuzzy rules; and
- Easy to incorporate both linguistic and numeric knowledge for problem-solving.

6.1.1 Architecture of ANFIS

The ANFIS consists of five layers: fuzzy layer, product layer, normalized layer, defuzzification layer, and summation layer (Wu et al., 2011), as shown in Figure 6.1. Layer 1 is the input layer. The crisp input values are transformed into fuzzy values by the MFs in this layer. The output from each node is a degree of membership value that is given by the input of MFs (Suparta & Alhasa, 2016).

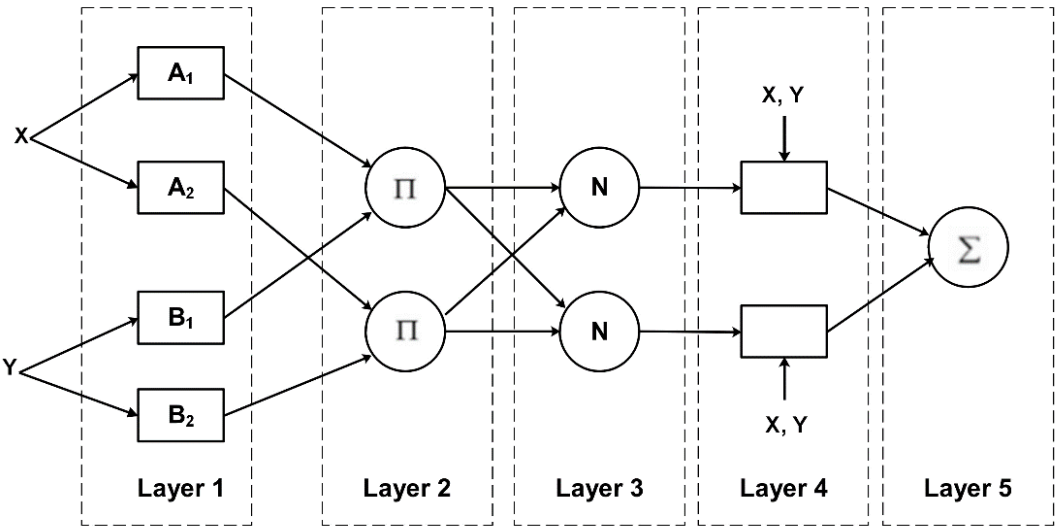


Figure 6.1: Architecture of ANFIS (Wu et al., 2011)

Layer 2 is the fuzzification layer. Neurons in this layer represent fuzzy sets used in the antecedents of the fuzzy rules. A fuzzification neuron receives a crisp input and determines the degree to which this input belongs to the neuron's fuzzy set. Every node in this layer is fixed and the node is labelled

as ∏. The output node is the result of multiplying the value coming into the node and delivered to the next node. Each node in this layer determines the weighting factor of each rule (Gao, Xue, Lu, & Dong, 2015).

Layer 3 is the fuzzy rule layer. Each fuzzy rule is represented by a neuron in this layer. This neuron receives inputs from the fuzzification neurons that represent fuzzy sets in the rule antecedents. Every node in this layer is fixed and the node is labelled as N. Layer 4 is the output membership layer. Neurons in this layer represent the fuzzy sets used in the consequent of fuzzy rules. An output membership neuron combines all its inputs by using the fuzzy operation union (Vieira, Dias, & Mota, 2004).

Layer 5 is the defuzzification layer. Each neuron in this layer represents a single output of the ANFIS. It takes the output fuzzy sets with different weights of fuzzy rules and combines them into a single fuzzy set. The single node in this layer provides the overall output as the summation of all incoming values from the previous node. In this layer, the node is labelled as ∑ (Wu et al., 2011).

6.1.2 ANFIS Learning Methods

Learning is one of the significant features provided by the ANFIS to modify the parameters and decrease the error rate to adapt to new environments. The ANFIS has two common learning methods; hybrid and backpropagation. This section provides an overview of these learning methods.

6.1.2.1 Hybrid Learning Method

The main purpose of the learning process is to update the system parameters to adapt to its environment. In the ANFIS architecture, the first layer and the fourth layer contain parameters that are updated using the learning method. The hybrid learning method is one of the common ANFIS learning methods proposed by Jang (1993). It consists of two main parts, namely forward and backward pass. In the forward pass, the parameters of the premises in the first layer should be in a steady-state. A Recursive Least Square Estimator (RLSE) method is applied to repair the consequent parameter in the fourth layer. Then, after the consequent parameters are obtained, input data are passed back to the adaptive network input, and the produced output is compared against the actual output (Suparta & Alhasa, 2016).

While in the backward pass, the consequent parameters should be in a steady-state. The error occurred during the comparison between the produced output and the actual output is propagated back to the first layer. At the same time, the parameter premises in the first layer are updated using gradient descent or backpropagation learning methods. With the use of the hybrid learning method that combines RSLE and gradient descent methods, it can ensure the convergence rate is faster because it reduces the dimensional search space in the original method of backpropagation (Pramanik & Panda, 2009).

The RLSE method is used to optimize the consequent parameters in the forward pass, while the gradient descent method is used to optimize the premise parameters in the backward pass, as depicted in Figure 6.2. The output of the ANFIS is calculated by employing the consequent parameters found in the forward pass. The output error is used to adapt the premise parameters by means of a standard backpropagation method. Several studies demonstrated that the hybrid method is highly efficient in training ANFIS systems (Jang et al., 1997).

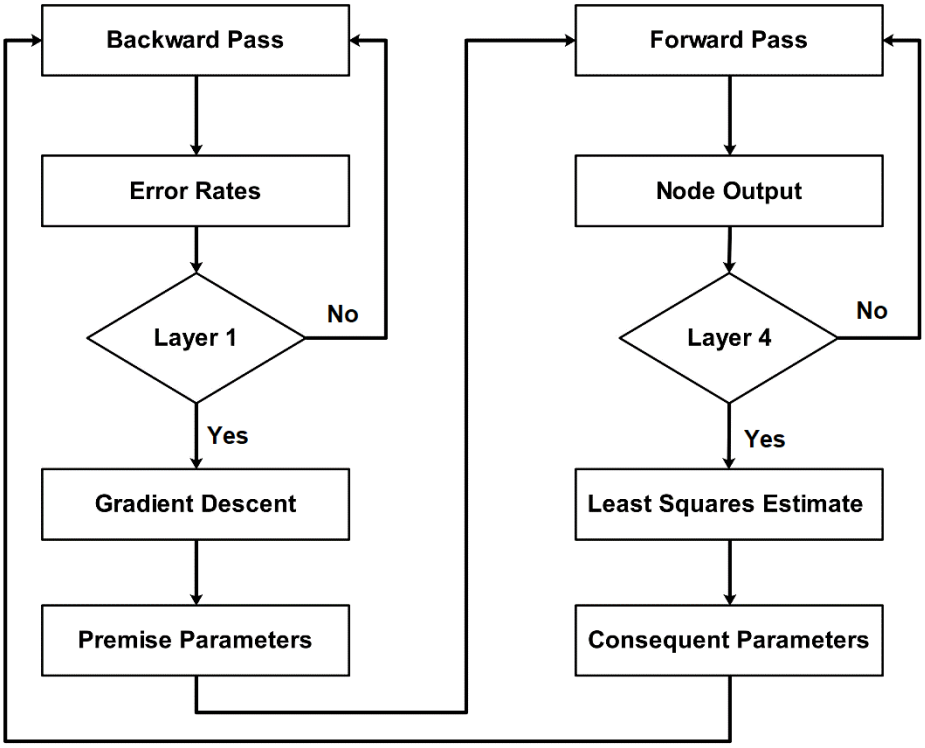


Figure 6.2: Flow chart of the hybrid learning algorithm (Ramesh et al., 2013)

6.1.2.2 Backpropagation Learning Method

Backpropagation is a common learning method in the ANN. It is a method of training multilayer ANNs by using the process of supervised learning. Supervised algorithms are based on errors in which the external reference signal is used to produce an error signal by comparing the produced output with the reference signal. Using the generated error signal, the ANFIS updates its parameters to improve the system performance (Saduf & Wani, 2013). The backpropagation method learns by evaluating the output layer to extract errors in the hidden layers. Due to its flexibility and learning capabilities, it has been implemented successfully in multiple applications (Haykin, 2004).

The backpropagation learning process can be described as follows:

- **Forward propagation of operating signal**: The input signal is propagated from the input layer to the output layer via the hidden layer. During the forward propagation of the operating signal, the weight and offset values of the network are maintained constant and the status of each layer of the neuron will only extend an effect on the next layer of the neuron. In case

that the expected output cannot be achieved in the output layer, it can be switched into the backpropagation of the error signal (Jing et al., 2012).

- **Backpropagation of error signal:** The difference between the desired output and obtained output of the network is defined as the error signal. In the backpropagation learning method, the error signal is propagated from the output layer to the input layer in a layer-by-layer manner. During the backpropagation of error signals, the weight value of the network is regulated by the error feedback. The continuous modification of weight and offset values are applied to make the obtained output of the network more closer to the desired output (Jing et al., 2012).

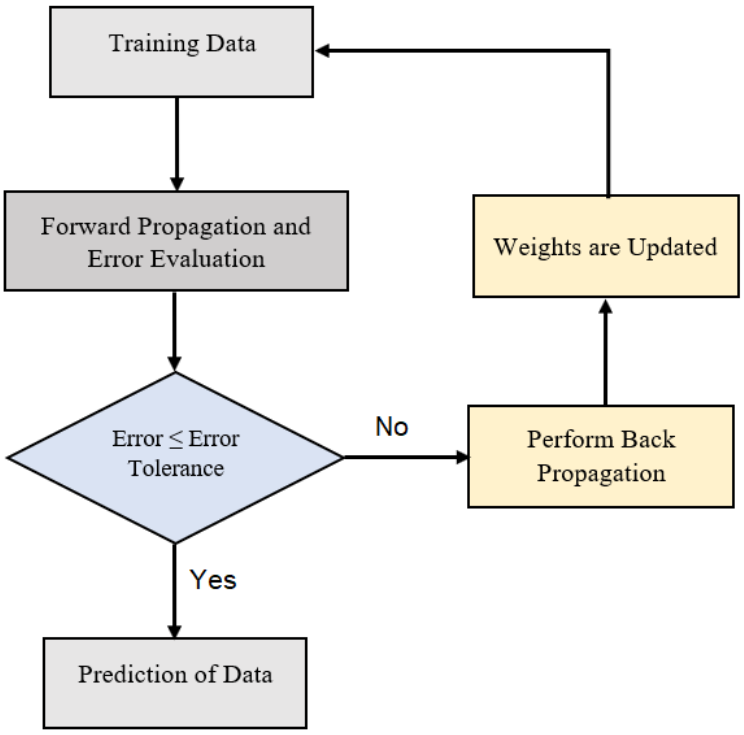


Figure 6.3: Flowchart of the backpropagation algorithm (Shaf et al., 2016)

The main objective of the backpropagation learning method is to adjust the values of weights in the training dataset to get the same value as the correct output value of the network using the validation dataset. The flow chart of the backpropagation method is shown in Figure 6.3. In the forward pass, input weights are injected to the subsequent layer. The activation function is implemented to generate the weights for the next layer (Shaf et al., 2016). Finally, the output layer is ready to generate the output value. The generated and original values of the output are utilized to derive the error which is propagated further back to the input layer. This process will continue until the error becomes less than a pre-defined error tolerance and the network is ready to be used or training will be terminated when reaching the maximum number of epochs (Okut, 2016; Shaf et al., 2016).

6.2 Implementation of ANFIS

ANFIS is a Sugeno-type FIS in which the parameters associated with MFs are computed using either a backpropagation learning method alone or in combination with a least square method (hybrid). It has been widely applied to random data sequences with highly irregular dynamics (Gao et al., 2015). Implementing the ANFIS requires building a fuzzy logic system with defining linguistic expressions for both input and output, defining fuzzy sets for input and output, specifying MFs, building the fuzzy rules, and train the neural network. Since the proposed risk estimation technique was implemented using the fuzzy logic system previously, as discussed in Chapter 5, so this chapter focuses only on training the ANFIS to achieve the best accuracy for the proposed risk estimation process.

The ANFIS model of the proposed risk estimation technique was trained to determine the appropriate number of epochs, MF, and learning method that produce the lowest error and the best fit with the learning process. Figure 6.4 shows the structure of the ANFIS model of the proposed risk estimation technique.

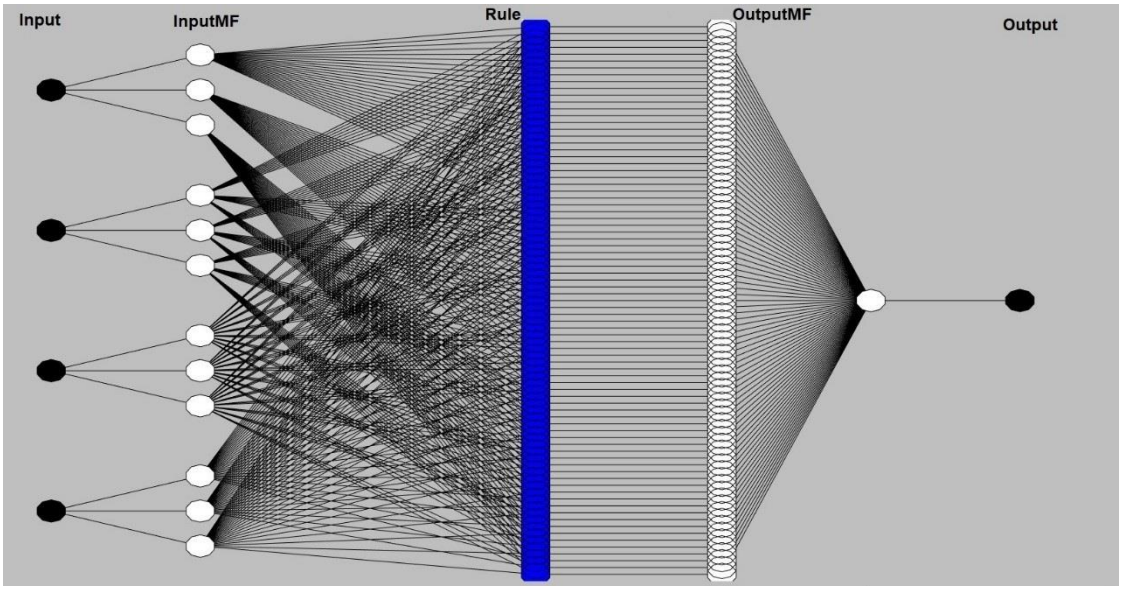


Figure 6.4: ANFIS model of the proposed risk estimation technique

As discussed earlier, the ANFIS model has five layers. The input layer contains four risk factors of the proposed risk-based access control model involving user context, resource sensitivity, action severity, and risk history. The second layer contains fuzzy sets of each input in which each risk factor is represented by three fuzzy sets. The third layer represents the fuzzy rules of the risk estimation technique, which are 81 rules. The fourth layer represents the output MF, which was represented by five fuzzy sets. The fifth layer represents the output layer which is the estimated risk value of the risk estimation process.

The main objective of training the ANFIS model of the proposed risk estimation technique is to tune different MFs and determine the appropriate MF that produces the lowest error and the best fit with the learning process. In addition, adding the learning capability to the risk estimation process to adapt

to new changes of various IoT applications and increase the accuracy of resultant risk values for future access requests.

6.2.1 Data collection

Implementing the ANFIS model requires having a dataset or examples for training. After implementing the proposed risk estimation technique using the fuzzy logic system with expert judgment, as discussed in Chapter 5, a dataset containing 160,000 records was created to train the ANFIS. To avoid possible bias in the sample data to the ANFIS model, the dataset was randomized and divided into two sets using the cross-validation method.

- **Training set**: This set contains 112,000 data records (70% of the dataset) to train the ANFIS model.
- **Checking set**: This set contains 48,000 data records (30% of the dataset) to test the ANFIS model.

6.2.2 Experimental Results

Several experiments were carried out to train the ANFIS model of the proposed risk estimation technique to increase the accuracy of the output risk, tune different MFs and identify the appropriate MF that lead to the lowest error and the best fit with the learning process at different number of training epochs. All training functions and experiments were coded and executed using MATLAB software.

6.2.2.1 Performance Evaluation

The ANFIS model was trained and the performance was evaluated using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), correlation coefficient (R), and coefficient of determination (R-square or R^2) as recommended in related ANFIS models (Ghorbanzadeh et al., 2018; Tiwari et al., 2018). The performance of the ANFIS model of the proposed risk estimation technique was tested at three different epochs; 20, 100, and 300 to observe error rates at different epochs and observe the performance when increasing the number of epochs.

1. **Root Mean Squared Error (RMSE)**

RMSE is a quadratic scoring rule that measures the average magnitude of the error. It's the square root of the average of squared differences between the predicted and actual output (Konaté et al., 2015). The mathematical representation of RMSE is as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(O_i - P_i)^2} \qquad (6.1)$$

Where n is the total number of data, O_i is the observed (target) value, and P_i is the predicted value.

2. Mean Absolute Error (MAE)

MAE had been cited in several ANFIS models as the primary measure of performance (Ahmed & Shah, 2017; De Myttenaere et al., 2015; Rahbari et al., 2018). MAE directly calculates the arithmetic mean of absolute errors. Hence, it is very easy to compute and understand. However, it may produce biased results when extremely large outliers exist in datasets (Konaté et al., 2015). The mathematical representation of MAE is as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(O_i - P_i) \tag{6.2}$$

Where n is the total number of data, O_i is the observed (target) value, and P_i is the predicted value.

3. Correlation Coefficient (R)

Correlation Coefficient (R) is used to measure the correlation between the observed value and the predicted value. It measures the strength of a linear relationship between the observed value and predicted variables (Shaf et al., 2016). In other words, it is an indicator of the scatters around the fit line. If R is close to 1, it means that the relationship between the observed and predicted variables is positive and thereby indicating that the data points fall nearly along a fit line with a positive slope. Whereas, when R is close to -1, the relationship between the observed and predicted variables is negative and the data points fall nearly along a fit line with a negative slope. When R is close to zero, it implies a weak relationship between the observed and predicted variables and the data points are scattered around the fit line and most of the data points are not in good agreement with the fit line (Konaté et al., 2015). The mathematical representation of R is as follows:

$$R = \frac{n\sum_{i=1}^{n}O_iP_i - \sum_{i=1}^{n}O_i\sum_{i=1}^{n}P_i}{\sqrt{(n\sum_{i=1}^{n}O_i^2 - (\sum_{i=1}^{n}O_i)^2)(n\sum_{i=1}^{n}P_i^2 - (\sum_{i=1}^{n}P_i)^2)}} \tag{6.3}$$

Where n is the total number of data, O_i is the observed (target) value, and P_i is the predicted value.

4. Coefficient of Determination (R^2)

The coefficient of determination is a measure of how well the regression line represents the data. If the regression line passes exactly through every point on the scatter plot, it would be able to explain all the variation (Tiwari et al., 2018). This coefficient is a statistical index that expresses the quality of fit of the regression equation and the intensity of the linear relationship. It helps to have a general idea of the model fit. Its value varies between 0 and 1, and if R^2 is close to 1, it will be sufficient to say that the fit is good (Konaté et al., 2015). The mathematical representation of R^2 is as follows:

$$R^2 = 1 - \left(\frac{\sum_{i=1}^{n} (O_i - \bar{O}) \times (P_i - \bar{P})}{\sqrt{\sum_{i=0}^{n} (O_i - \bar{O})^2 \times \sqrt{\sum_{i=0}^{n} (P_i - \bar{P})^2}} \right)^2 \quad (6.4)$$

Where n is the total number of data, O_i is the observed (target) value, and P_i is the predicted value, $\bar{O}$ is the mean observed value, and $\bar{P}$ is the mean predicted value.

6.2.2.2 Training ANFIS Model

The performance of most machine learning techniques is improved by training. This section discusses the training of the ANFIS model. The training process begins by dividing the dataset into a training dataset and checking dataset. The training dataset is a set of input and output vectors. Two vectors are used to train the ANFIS system: the input vector and the output vector. The training dataset is used to find the premise parameters for the MFs. A threshold value for the error between the observed and predicted output is determined to be 0.05 (Al-Hmouz et al., 2012). The consequent parameters are decided using the least-squares method. If this error is larger than the threshold value, then the premise parameters are updated using the gradient descent method. The process is terminated when the error becomes less than the threshold value. The checking dataset is then used to test the ANFIS model with the actual data (Jang, 1993).

The ANFIS model of the proposed risk estimation technique was trained using both hybrid and backpropagation learning methods. Eight MFs were used in the training process to determine the appropriate learning method as well as the appropriate MF to implement the risk estimation process of the proposed risk-based model. These MFs include TriMF, TrapMF, GbellMF, GaussMF, Gauss2MF, PimF, DsigMF, and PsigMF.

Figure 6.5 shows the ANFIS training process. The first step is to prepare the training data to work with ANFIS model in MATLAB. The dataset used as the input for the *anfis* function must be in a matrix form, where the last column in the matrix represents the output, and the matrix contains as many columns as needed to represent the inputs of the system. The rows represent all the existing data combinations. The GUI method of the fuzzy logic toolbox in MATLAB was utilized to determine the type of MF for fuzzy sets before starting the training process. Then, the ANFIS model was trained using eight MFs to determine the suitable MF with the lowest error and the best fit. When the training process finished, the final MFs and training errors from the training dataset were recorded. In addition, the checking dataset was used in conjunction with the training dataset to enhance the performance (Al-Hmouz et al., 2012). When the checking process finished, the trained FIS is utilized to evaluate the performance of the system.

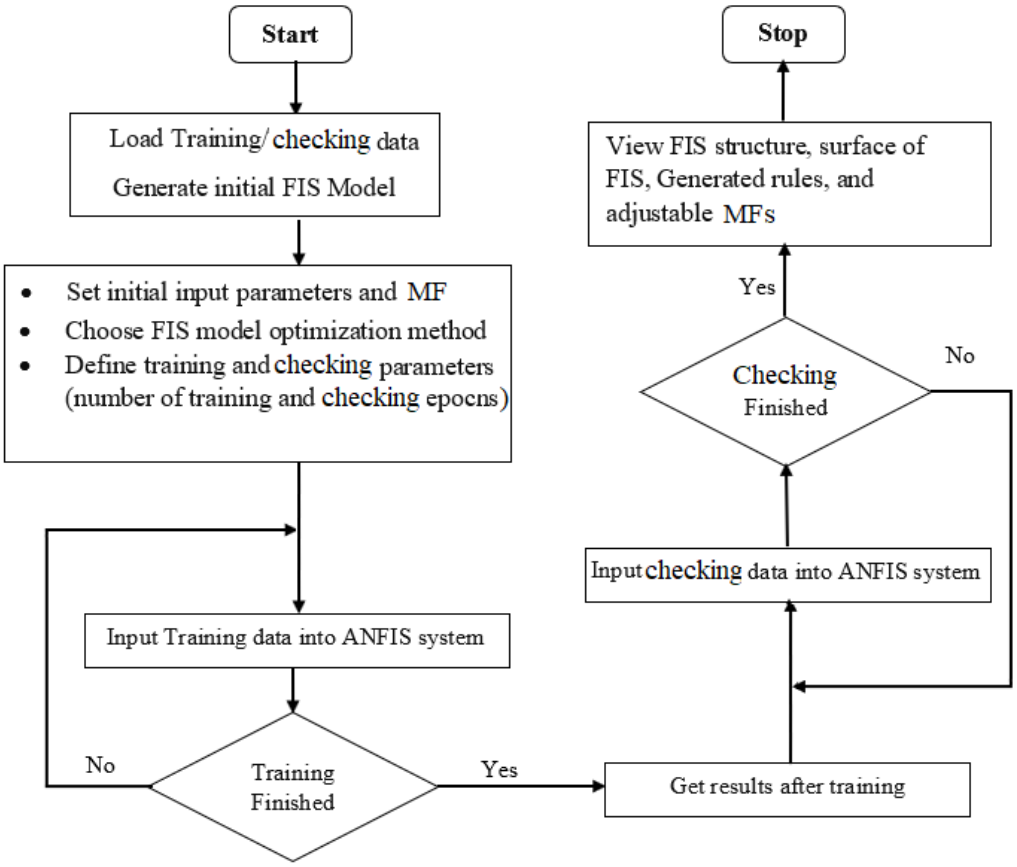


Figure 6.5: ANFIS training process (Al-Hmouz et al., 2012)

After the training completed, the performances of the ANFIS model was evaluated to determine the best fuzzy parameters with the lowest error and the best fit. The trained FIS of each MF was utilized to produce the predicted output. Then, the predicted output was compared with the observed output to determine the error using MAE and RMSE and determine the best fit with the learning process using R and R^2. Several experiments were carried out to train the ANFIS model and evaluate the performance of the trained FIS. This process took more than four months working on three PCs simultaneously to train and evaluate the performance of the ANFIS model.

One of the common problems that may occur during the training of the ANFIS model is Overfitting. It occurs when the data are overtrained. Generally, every trained dataset has its maximum number of epochs before overfitting occurs. Overfitting causes the predicted output to be over its accuracy (Al-Hmouz et al., 2012). Therefore, each dataset should be trained using an optimal number of epochs, which can be decided by conducting numerous experiments. Overfitting is analysed by plotting the training and checking errors from the ANFIS simulation. To avoid the overfitting problem, the ANFIS model should be trained for a different number of epochs.

6.2.2.3 Training Results

Various experiments were carried out using two separate datasets: training dataset (112,000 records) and checking dataset (48,000 records). The training dataset was used to train the ANFIS, whereas

the checking data set was used to verify the accuracy of the trained ANFIS model. To produce the lowest error and the best fit with the learning process, the ANFIS model was trained at three different epochs; 20, 100, and 300. In the next section, the results of training the ANFIS model at 20, 100, and 300 epochs will be discussed.

6.2.2.3.1 Training at 20 Epochs

The optimal setting of the ANFIS model depends on different MFs, learning method and the number of epochs for each training. Several experiments were carried out to train the ANFIS model using both hybrid and backpropagation training algorithms at 20 epochs with eight MFs to determine the best MF that produces the lowest error and the best fit with the learning process.

Table 6.1: Performance evaluation of the ANFIS model at 20 epochs

Learning algorithm	**MF**	**Training Error**	**Checking Error**	**Performance Evaluation**			
				RMSE	**MAE**	**R**	**R^2**
Hybrid	TriMF	5.3507	5.4031	5.3784	4.2339	0.9641	0.9294
	TrapMF	4.6438	4.6552	4.6647	3.5611	0.9731	0.9469
	GbellMF	5.1626	5.1762	5.2392	4.0783	0.9659	0.9330
	GaussMF	5.2102	5.2341	5.1913	4.0109	0.9666	0.9342
	Gauss2MF	4.6611	4.6706	4.6810	3.5720	0.9729	0.9465
	PiMF	4.8445	4.8525	4.8678	3.7118	0.9707	0.9422
	DsigMF	4.6974	4.7069	4.7184	3.5982	0.9725	0.9457
	PsigMF	4.6975	4.7068	4.7184	3.5984	0.9725	0.9457
Backpropagation	TriMF	51.5436	51.5447	51.5337	48.2481	0.8317	-5.4804
	TrapMF	51.3364	51.3235	51.3188	48.1093	0.7262	-5.4264
	GbellMF	52.0326	52.0314	52.0209	48.6320	0.8700	-5.6035
	GaussMF	51.7004	51.6991	51.6893	48.3604	0.8710	-5.5195
	Gauss2MF	51.3429	51.3335	51.3266	48.1076	0.7379	-5.4284
	PiMF	51.3242	51.3098	51.3064	48.0995	0.7106	-5.4233
	DsigMF	51.3346	51.3242	51.3180	48.0988	0.7339	-5.4262
	PsigMF	51.3346	51.3242	51.3180	48.0988	0.7339	-5.4262

In addition, after the ANFIS model has been trained, the entire dataset was utilized to check the performance and the accuracy of the ANFIS model. RMSE, MAE values were used to indicate the error value between the predicted values obtained from the trained ANFIS model against the original values. In addition, R and R^2 were used to show the model fitness with the training process. Results of training the ANFIS model at 20 epochs can be shown in Table 6.1.

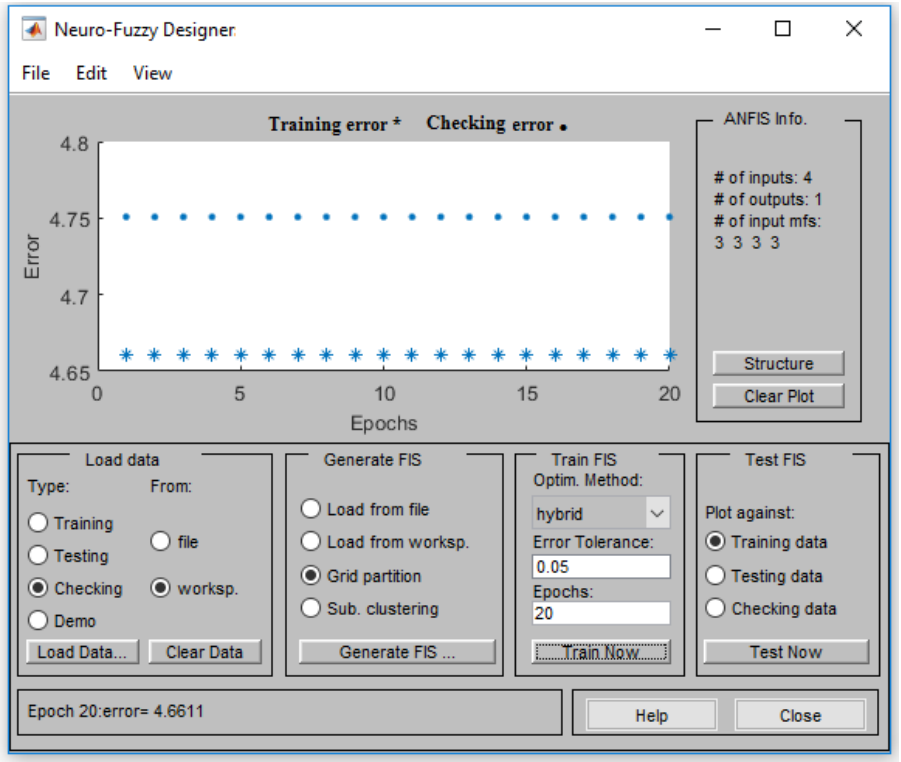


Figure 6.6: Training and checking error when applying TrapMF with the hybrid learning at 20 epochs

With the hybrid learning method, the training and checking errors were very small for all eight MFs. The combination of backpropagation and descent algorithms in the hybrid learning method has demonstrated it can reach small error values only after 20 epochs. In addition, the results have demonstrated that four MFs including TrapMF, PiMF, DsigMF, and Gauss2MF produced the same training and checking errors during all 20 epochs, which illustrates that no error enhancement occurs with these MFs when increasing the number of epochs. Figure 6.6 shows training and checking errors when applying TrapMF with the hybrid learning method at 20 epochs, which illustrates that no improvements occur when increasing the number of epochs. While another four MFs including TriMF, GbellMF, PiMF, and GaussMF show a slight decrease in training and checking errors when increasing the number of epochs from 1 to 20, as depicted in Figure 6.7.

Although TrapMF does not show any improvement when increasing the number of epochs, the results demonstrated that it is the best MF that provided the lowest RMSE and MAE error values among other MFs and the best fit with the learning process with values of R and R^2 as 0.9731 and 0.9469 respectively. Figure 6.8 shows the regression plot of the entire dataset used to evaluate the performance when applying TrapMF with the hybrid learning method. It shows that the predicted values are very close to the ideal linear line and the proposed ANFIS model is well fit as well.

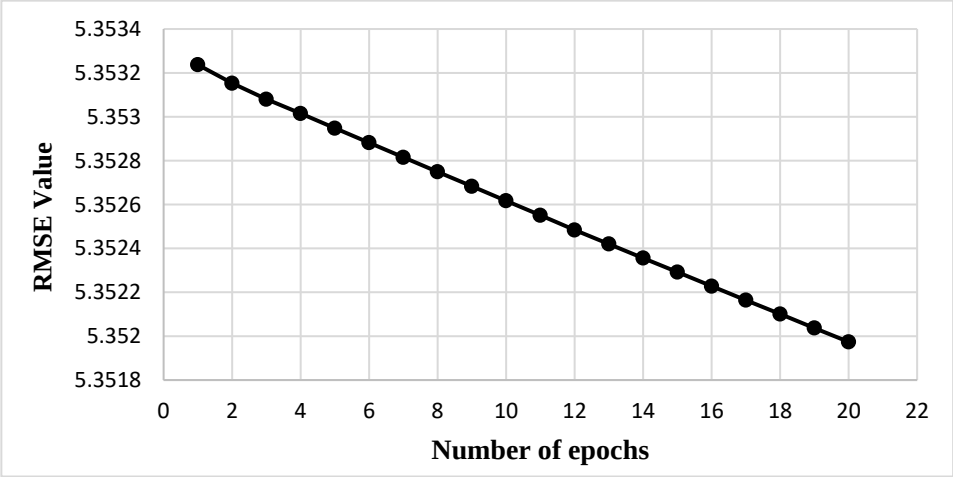


Figure 6.7: RMSE training error when applying TriMF with the hybrid learning method. It shows a slight decrease in the error when increasing the number of epochs.

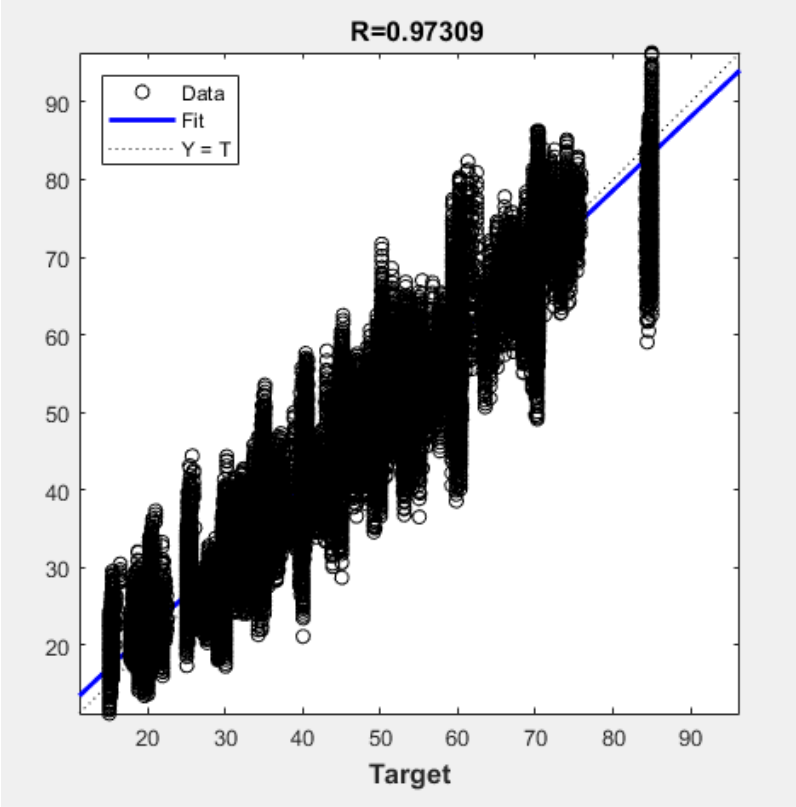


Figure 6.8: Regression after applying TrapMF with the hybrid learning method

In addition, the backpropagation learning method was utilized to train the ANFIS model at 20 epochs. The results showed that the backpropagation learning method produced a large decrease in both training and checking errors. All eight MFs showed a large decrease in both training and checking errors. However, the results showed that the backpropagation learning method produces large RMSE and MAE error values and small R and R^2 values. This, in turn, reflects the fact that the relationship between the predicted and observed data is less efficient and need more training. In addition, the R^2 values were negative which implies there is an inverse relationship between the predicted and observed data such that the increase in the predicted data will cause a decrease in the observed data.

The significant aspect observed from applying the backpropagation learning method at 20 epochs is that the training and checking errors were decreased dramatically when increasing the number of epochs with all eight MFs. Figure 6.9 shows a dramatic decrease in both training and checking RMSE errors when applying TrapMF with backpropagation learning method at 20 epochs. It also shows that there is no overfitting.

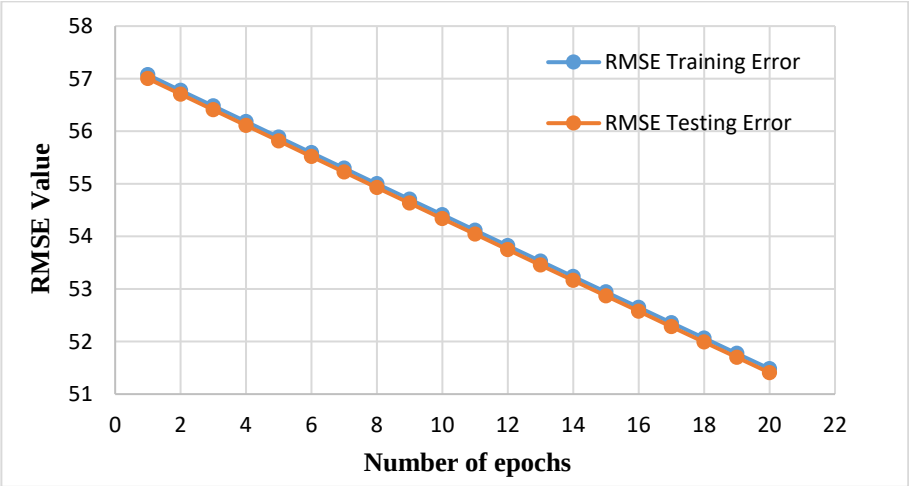


Figure 6.9: RMSE of training and checking errors when applying TrapMF with backpropagation learning method at 20 epochs

After training the ANFIS model of the proposed risk estimation technique with both hybrid and backpropagation learning methods at 20 epochs, the results demonstrated that increasing the number of epochs have a slight effect on training and checking errors when applying the hybrid learning method, while it has a significant effect on training and checking errors when applying the backpropagation learning method. In addition, the TrapMF with the hybrid learning approach is the optimal MF that produces the lowest error and the best fit with the learning process

6.2.2.3.2 Training at 100 Epochs

Defining the best settings for the proposed ANFIS model to train it and produce the highest accuracy depend on MFs, learning method and the number of epochs for each training. After the ANFIS model was trained at 20 epochs, it was trained at 100 epochs to observe the performance when increasing the number of epochs. The reason to train the ANFIS model at 100 epochs in this experiment is that the training at 20 epochs demonstrated a slight effect on decreasing training and checking errors with the hybrid learning method and a significant decrease in training and checking errors with the backpropagation learning method. So, increasing the number of epochs may produce more improvements, especially with the backpropagation learning method. Hence, the target is to train the ANFIS model at 100 epochs and observe training and checking errors to reach the best fit and the lowest error.

Several experiments were carried out at 100 epochs with eight MFs to determine the best MF that produces the lowest error and the best fit with the learning process using both hybrid and

backpropagation training methods. Training and checking errors and performance evaluation metrics resulted from the training can be shown in Table 6.2.

Table 6.2: Performance evaluation of the ANFIS model at 100 epochs

Learning algorithm	MF	Training Error	Checking Error	Performance Evaluation			
				RMSE	MAE	R	R^2
Hybrid	TriMF	5.3473	5.3998	5.3748	4.2320	0.9641	0.9295
	TrapMF	4.6438	4.6552	4.6647	3.5611	0.9731	0.9469
	GbellMF	5.1084	5.1298	5.1370	3.9757	0.9673	0.9356
	GaussMF	5.1928	5.2197	5.2222	4.0634	0.9662	0.9335
	Gauss2MF	4.6611	4.6706	4.6810	3.5720	0.9729	0.9465
	PimF	4.8392	4.8467	4.8623	3.7075	0.9707	0.9423
	DsigMF	4.6974	4.7069	4.7184	3.5982	0.9725	0.9457
	PsigMF	4.6975	4.7068	4.7184	3.5984	0.9725	0.9457
Backpropagation	TriMF	29.5731	29.5992	29.5857	27.1901	0.8436	-1.1359
	TrapMF	29.4777	29.4701	29.4643	26.7725	0.7857	-1.1184
	GbellMF	31.3456	31.3513	31.3471	28.8931	0.8770	-1.3978
	GaussMF	30.0580	30.0639	30.0624	27.7479	0.8781	-1.2053
	Gauss2MF	29.3942	29.3970	29.3847	26.7108	0.7888	-1.1070
	PimF	29.5411	29.5357	29.5294	26.7731	0.7794	-1.1278
	DsigMF	28.3863	28.3875	28.3793	25.5234	0.7763	-0.9653
	PsigMF	28.3862	28.3874	28.3792	25.5233	0.7763	-0.9652

The results demonstrated that the ANFIS behaviour at 100 epochs was similar to the one at 20 epochs. For the hybrid learning method, the training and checking errors showed a very slight decrease compared to error values produced at 20 epochs. In other words, a group of MFs including TrapMF, Gauss2MF, DsigMF, PsigMF did not show any differences in training and checking errors as well as performance evaluation metrics when increasing the number of epochs from 20 to 100. While another group of MFs including TriMF, GbellMF, GaussMF, and PiMF have shown a very small decrease in training and checking errors when increasing the number of epochs to 100. Figure 6.10 shows training and checking errors when applying GbellMF with the hybrid learning method at 100 epochs. It showed a slight decrease in training and checking errors when increasing the number of epochs.

Again, although the TrapMF does not show any improvements when increasing the number of epochs to 100, the results demonstrated that it is still the best MF that provided the lowest RMSE and MAE error values among other MFs and the best fit with the learning process with values of R and R^2 as 0.9731 and 0.9469 respectively.

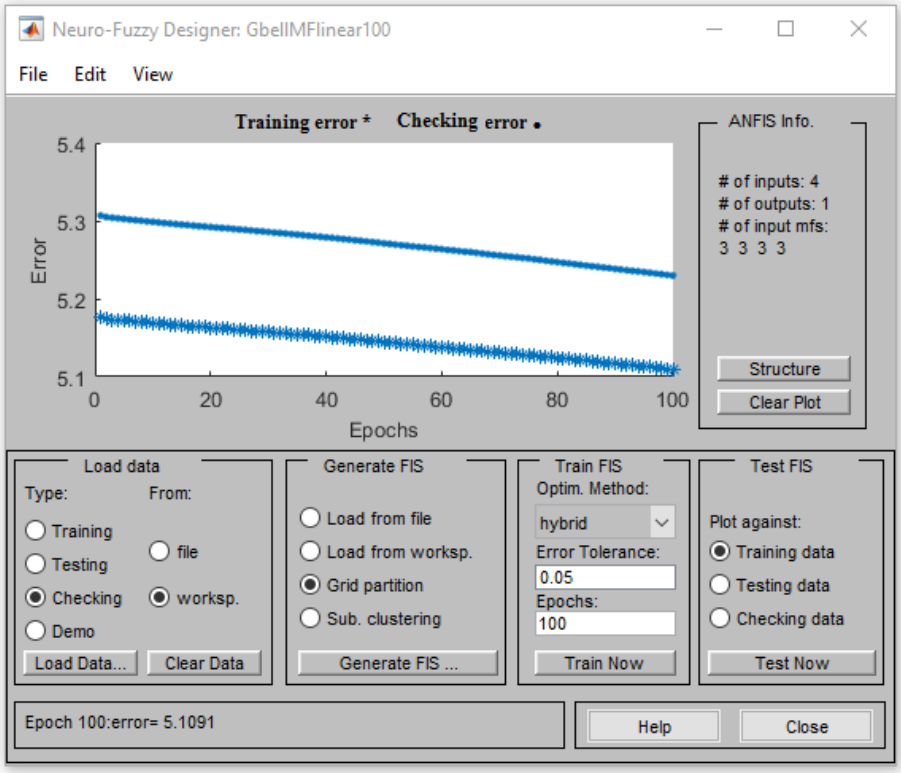


Figure 6.10: Training and checking errors when applying the GbellMF with the hybrid learning method at 100 epochs

For the backpropagation learning method, increasing the number of epochs to 100 demonstrated a dramatic decrease in training and checking errors for all MFs. The training error decreased from 51.3 at 20 epochs to reach 28.3 at 100 epochs for both DsigMF and PsigMF, which demonstrates the effect of increasing the number of epochs. Figure 6.11 shows training and checking errors at 100 epochs when applying the TriMF with the backpropagation learning method. It also shows that there is no overfitting as training and checking values have the same behaviour.

Although both training and checking errors were decreased dramatically when applying the backpropagation learning method, it still quite high which reflects on the performance evaluation metrics where it produced high RMSE and MAE values and small R and R^2 values. This demonstrates that the relationship between the predicted and observed data is less efficient and need more training. In addition, R^2 values are negative which implies there is an inverse relationship between the predicted and observed data.

After training the ANFIS model of the proposed risk estimation technique with both hybrid and backpropagation learning methods at 100 epochs, the results demonstrated that increasing the number of epochs have a slight decrease in training and checking errors with the hybrid learning method and a significant effect on decreasing training and checking errors with the backpropagation learning method. In addition, the TrapMF with the hybrid learning approach is still the optimal MF that produces the lowest error and the best fit with the learning process.

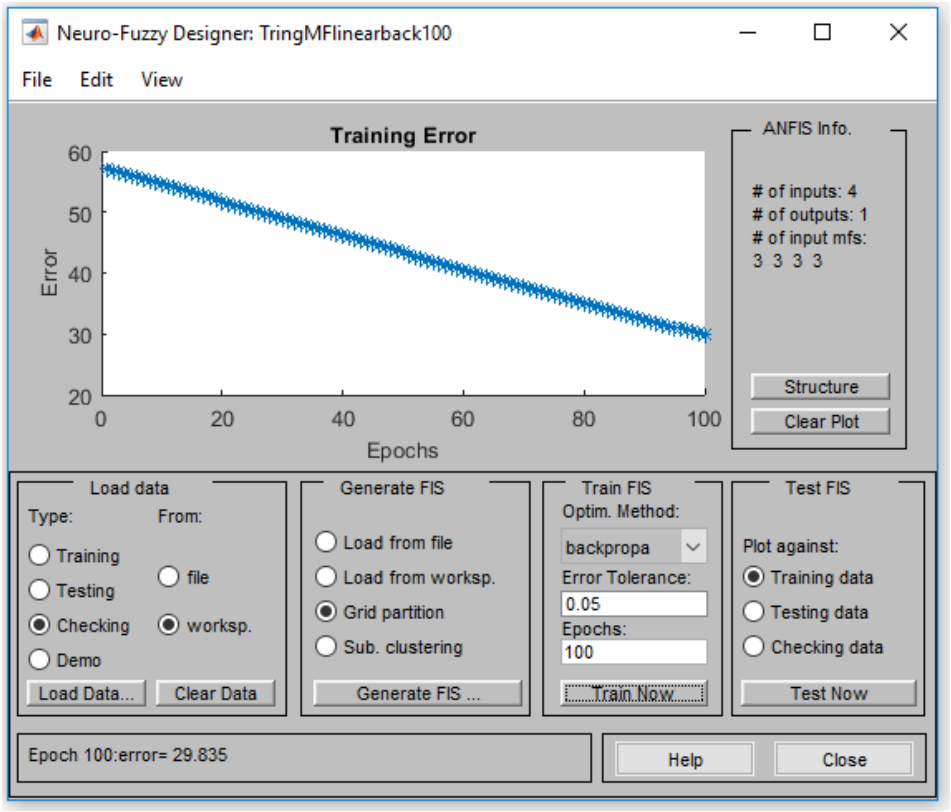


Figure 6.11: Training and checking errors when applying the TriMF with the backpropagation learning method at 100 epochs

6.2.2.3.3 Training at 300 Epochs

Training the ANFIS model with the hybrid learning method provides adequate results but it also showed that it needs more training with the backpropagation learning method. Therefore, the ANFIS model was trained at 300 epochs to observe the performance when increasing the number of epochs to 300. Several experiments were carried out at 300 epochs with eight MFs using both hybrid and backpropagation training methods to determine the best MF that produces the lowest error and the best fit with the learning process. Results of training the ANFIS model at 300 epochs can be shown in Table 6.3.

The results demonstrated that the ANFIS behaviour at 300 epochs was similar to the one at 20 and 100 epochs. For the hybrid learning method, the training and checking errors showed a very slight decrease compared to error values produced at 20 or 100 epochs. In other words, a group of MFs including TrapMF, Gauss2MF, DsigMF, PsigMF did not show any differences in training and checking errors as well as performance evaluation metrics when increasing the number of epochs to 300.

Table 6.3: Performance evaluation of the ANFIS system with 300 epochs

Learning algorithm	MF	Training Error	Checking Error	Performance Evaluation			
				RMSE	MAE	R	R^2
Hybrid	TriMF	5.3392	5.3919	5.3660	4.2282	0.9642	0.9297
	TrapMF	4.6438	4.6552	4.6647	3.5611	0.9731	0.9469
	GbellMF	4.9888	5.0047	5.0127	3.8696	0.9689	0.9387
	GaussMF	5.1389	5.1714	5.1681	4.0091	0.9669	0.9348
	Gauss2MF	4.6611	4.6706	4.6810	3.5720	0.9729	0.9465
	PiMF	4.8294	4.8357	4.8521	3.6985	0.9709	0.9426
	DsigMF	4.6974	4.7069	4.7184	3.5982	0.9725	0.9457
	PsigMF	4.6975	4.7068	4.7184	3.5984	0.9725	0.9457
Backpropagation	TriMF	6.3084	6.3647	6.3402	5.0299	0.9497	0.9019
	TrapMF	5.9086	5.9446	5.9357	4.6255	0.9561	0.9140
	GbellMF	6.2915	6.3496	6.3289	4.9915	0.9500	0.9023
	GaussMF	6.4113	6.4639	6.4493	5.0978	0.9480	0.8985
	Gauss2MF	5.8614	5.8983	5.8884	4.5774	0.9568	0.9154
	PiMF	6.0306	6.0661	6.0598	4.7266	0.9542	0.9104
	DsigMF	9.9248	10.0127	9.9500	7.8745	0.8949	0.7584
	PsigMF	7.5377	7.5890	7.5597	6.0057	0.9317	0.8605

While another group of MFs including TriMF, GbellMF, GaussMF, and PiMF have shown a very small decrease in training and checking errors when increasing the number of epochs to 300. Figure 6.12 shows training and checking errors when applying the GbellMF with the hybrid learning method at 300 epochs. It showed a slight decrease in training and checking errors when increasing the number of epochs. It also showed that at about 275 epochs, the error improvement has stopped, which implies that there is no need for more training, as increasing the number of epochs will not produce any changes. Also, it indicates that there is no overfitting. Also, the results demonstrated that the TrapMF is still the best MF that provided the lowest RMSE and MAE errors among other MFs and the best fit with the learning process with values of R and R^2 as 0.9731 and 0.9469 respectively.

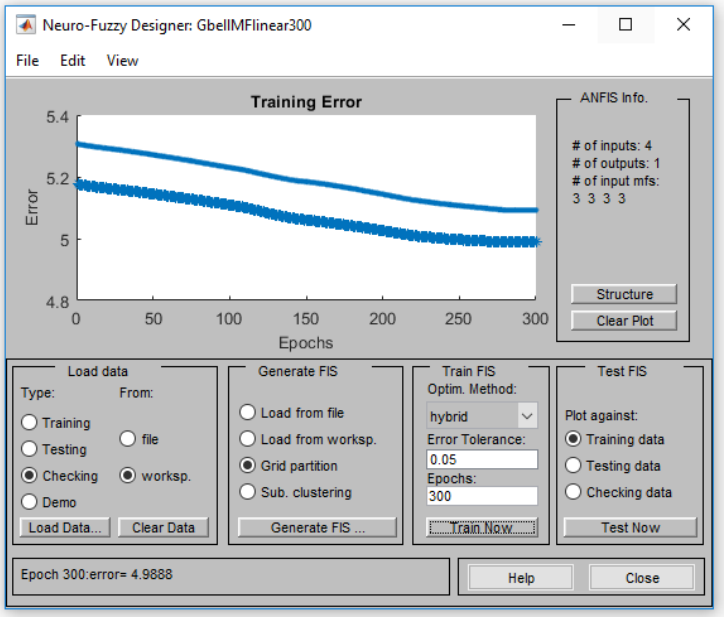


Figure 6.12: Training and checking errors when applying the GbellMF with the hybrid learning method at 300 epochs

For the backpropagation learning method, increasing the number of epochs to 300 demonstrated a dramatic decrease in training and checking errors for all MFs. The training error decreased from 29.3 at 100 epochs to 5.8 at 300 epochs for the Gauss2MF, which demonstrates the effect of increasing the number of epochs. Figure 6.13 shows training and checking errors when applying the TriMF with the backpropagation learning method at 300 epochs. It showed that at about 280 epochs, the error decrease has almost stopped, which implies that there is no need for more training. It also shows that there is no overfitting.

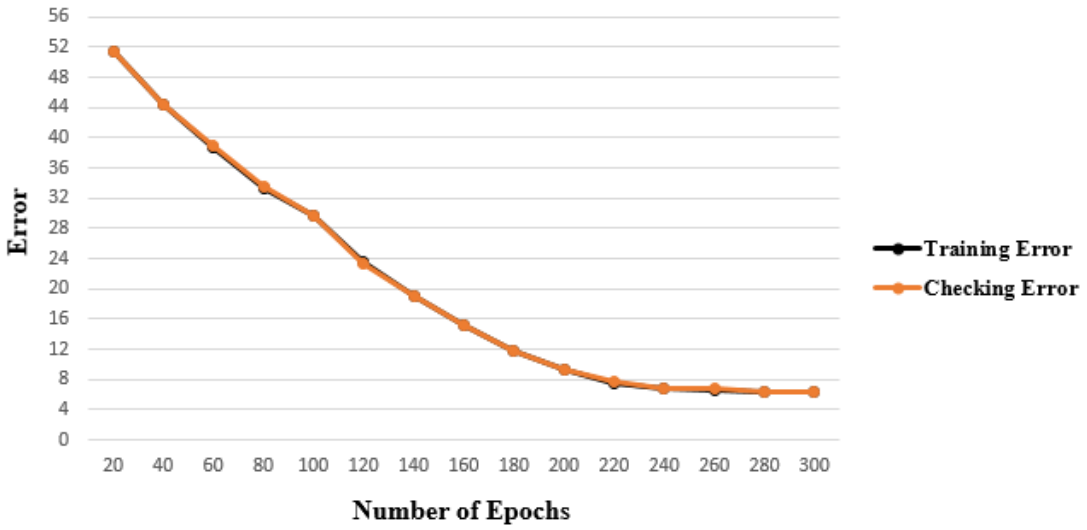


Figure 6.13: Training and checking errors with epoch number when applying TriMF with the backpropagation learning method

The performance evaluation of the backpropagation learning method at 300 epochs showed different results. TriMF and Gauss2MF have produced the lowest RMSE and MAE errors as well as the highest R and R^2 values. On the other hand, DsigMF has produced the highest RMSE and MAE errors and the lowest R and R^2 values. In addition, although the backpropagation learning approach showed a dramatic decrease in both training and checking errors when increasing the number of epochs as it reaches its lowest error (5.86) after about 280 epochs, the hybrid learning approach reaches to its lowest error value (4.64) only after one epoch with the TrapMF.

6.2.2.4 Comparison of Learning Methods at different Epochs

The ANFIS model of the proposed risk estimation technique was trained using both hybrid and backpropagation learning techniques at three different epochs numbers 20, 100, and 300 to investigate the learning rate of the ANFIS model with different epochs and determine the best MF that produces the lowest error and the best fit with the learning process.

The results of the hybrid learning method at 20, 100, and 300 epochs have demonstrated that a group of MFs including TrapMF, Gauss2MF, DsigMF, and PsigMF did not show any changes in RMSE, MAE errors as well as R and R^2 values when increasing the number of epochs from 1 to 300, as depicted in Table 6.4. While another group of MFs including TriMF, GbellMF, GaussMF, and PiMF

have shown a very slight decrease in RMSE and MAE errors values and a very small increase in R and R^2 values when increasing the number of epochs. For instance, the RMSE value of the TriMF is decreased from 5.378 to 5.375 when increasing the number of epochs from 20 to 100 and further decreased to 5.366 when increasing the number of epochs to 300. The same behaviour continued for this group of MFs except for GaussMF which showed a different behaviour, in which the RMSE value increased from 5.191 at 20 epochs to 5.222 when increasing the number of epochs to 100, but it decreased again to reach 5.168 when increasing the number of epochs to 300. In addition, the GbellMF produced the largest amount of error decrease among other MFs in which its RMSE value decreased from 5.239 at 20 epochs to 5.013 at 300 epochs.

Table 6.4: Performance evaluation of the ANFIS model with the hybrid learning method at different epochs

MF	At 20 epochs				At 100 epochs				At 300 epochs			
	RMSE	**MAE**	**R**	**R^2**	**RMSE**	**MAE**	**R**	**R^2**	**RMSE**	**MAE**	**R**	**R^2**
TriMF	5.378	4.234	0.964	0.929	5.375	4.232	0.964	0.930	5.366	4.228	0.964	0.930
TrapMF	4.665	3.561	0.973	0.947	4.665	3.561	0.973	0.947	4.665	3.561	0.973	0.947
GbellMF	5.239	4.078	0.966	0.933	5.137	3.976	0.967	0.936	5.013	3.870	0.969	0.939
GaussMF	5.191	4.011	0.967	0.934	5.222	4.063	0.966	0.934	5.168	4.009	0.967	0.935
Gass2MF	4.681	3.572	0.973	0.947	4.681	3.572	0.973	0.947	4.681	3.572	0.973	0.947
PiMF	4.868	3.712	0.971	0.942	4.862	3.708	0.971	0.942	4.852	3.699	0.971	0.943
DsigMF	4.718	3.598	0.973	0.946	4.718	3.598	0.973	0.946	4.718	3.598	0.973	0.946
PsigMF	4.718	3.598	0.973	0.946	4.718	3.598	0.973	0.946	4.718	3.598	0.973	0.946

The results of the backpropagation learning method at 20, 100, and 300 epochs have demonstrated that all MFs have shown a significant decrease in both RMSE and MAE values and a significant increase in R and R^2 values when increasing the number of epochs, as shown in Table 6.5. For example, the RMSE value of the TriMF is decreased from 51.53 to 29.59 when increasing the number of epochs from 20 to 100 and further decreased to 6.34 when increasing the number of epochs to 300. There was a negative sign of R^2 values at 20 and 100 epochs which implies there was an inverse relationship between the predicted and observed data. This negative sign disappeared when increasing the number of epochs to 300. After applying the backpropagation learning method with different number of epochs, the results demonstrated that the Gauss2MF is the best MF as it produced the lowest RMSE (5.888) and MAE (4.577) values and the highest R (0.957) and R^2 (0.915) values.

In conclusion, increasing the number of epochs was having a significant effect on MFs with the backpropagation learning method in which the learning process took about 280 epochs to reach its lowest RMSE and MAE values. Although the RMSE error values of MFs with the backpropagation learning method decreased significantly when increasing the number of epochs, it could not reach the lowest error value produced by the hybrid learning method only after one epoch.

Table 6.5: Performance evaluation of the ANFIS model with the backpropagation learning methods at different epochs

MF	With 20 epochs				With 100 epochs				With 300 epochs			
	RMSE	**MAE**	**R**	**R^2**	**RMSE**	**MAE**	**R**	**R^2**	**RMSE**	**MAE**	**R**	**R^2**
TriMF	51.53	48.25	0.832	-5.480	29.59	27.19	0.844	-1.136	6.340	5.030	0.950	0.902
TrapMF	51.32	48.11	0.726	-5.426	29.46	26.77	0.786	-1.118	5.936	4.626	0.956	0.914
GbellMF	52.02	48.63	0.870	-5.604	31.35	28.89	0.877	-1.398	6.329	4.992	0.950	0.902
GaussMF	51.69	48.36	0.871	-5.520	30.06	27.75	0.878	-1.205	6.449	5.098	0.948	0.899
Gass2MF	51.33	48.11	0.738	-5.428	29.39	26.71	0.789	-1.107	5.888	4.577	0.957	0.915
PiMF	51.31	48.10	0.711	-5.423	29.53	26.77	0.779	-1.128	6.060	4.727	0.954	0.910
DsigMF	51.32	48.10	0.734	-5.426	28.38	25.52	0.776	-0.965	9.950	7.875	0.895	0.758
PsigMF	51.32	48.10	0.734	-5.426	28.38	25.52	0.776	-0.965	7.560	6.006	0.932	0.861

Investigating the results with both hybrid and backpropagation learning methods demonstrates that the TrapMF with the hybrid learning method at 20 epochs is the optimal combination to implement the ANFIS model of the proposed risk estimation technique. It produced the lowest RMSE, MAE values as well as the highest R and R^2 values among all other MFs at different number of epochs only after one epoch, as shown in Figure 6.14. It reached the best fit with the learning process with a correlation of 0.9731, which shows that the predicted values are very close to the ideal linear line and the proposed ANFIS model is well trained.

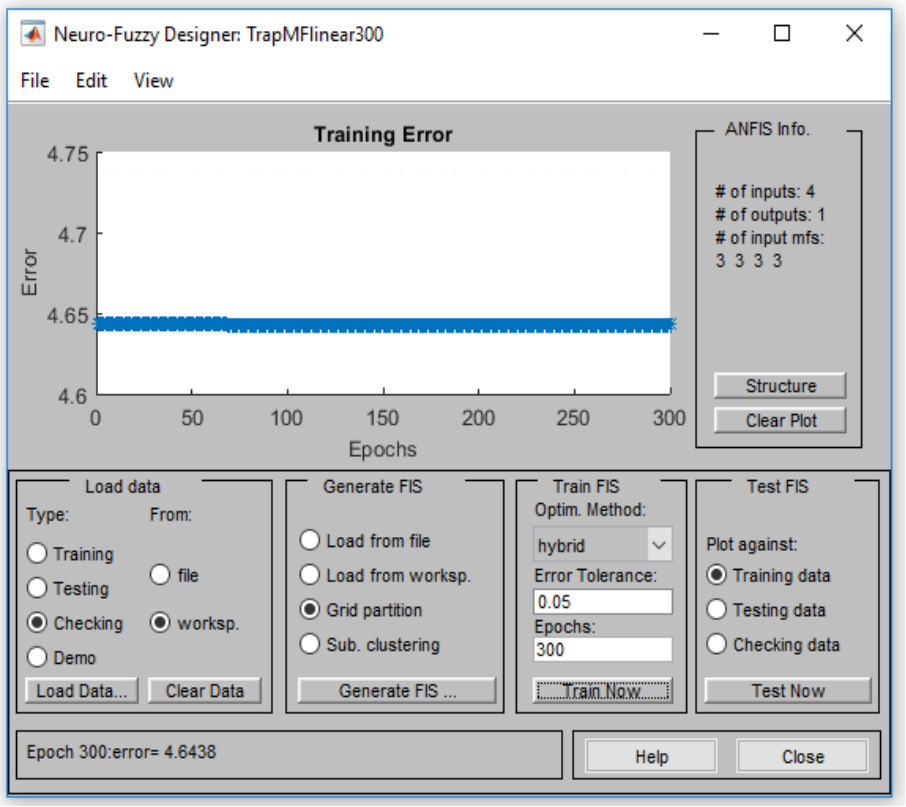


Figure 6.14: Training and checking errors when applying the TrapMF with the hybrid learning method. It reached the lowest training and checking error only after one epoch and still the same with increasing the number of epochs

6.3 Fuzzy System and ANFIS

The proposed risk estimation technique was first implemented using the fuzzy logic system through Mamdani FIS. One of the challenges that stands as a barrier for adopting the Mamdani FIS in the risk estimation is choosing the appropriate MF that provides the best accuracy for the output risk. Therefore, the ANFIS was adopted to provide a good way to tune different MFs to select the optimal method that results in increasing the accuracy of the output as well as adding the learning capability to the proposed risk estimation technique to increase accuracy.

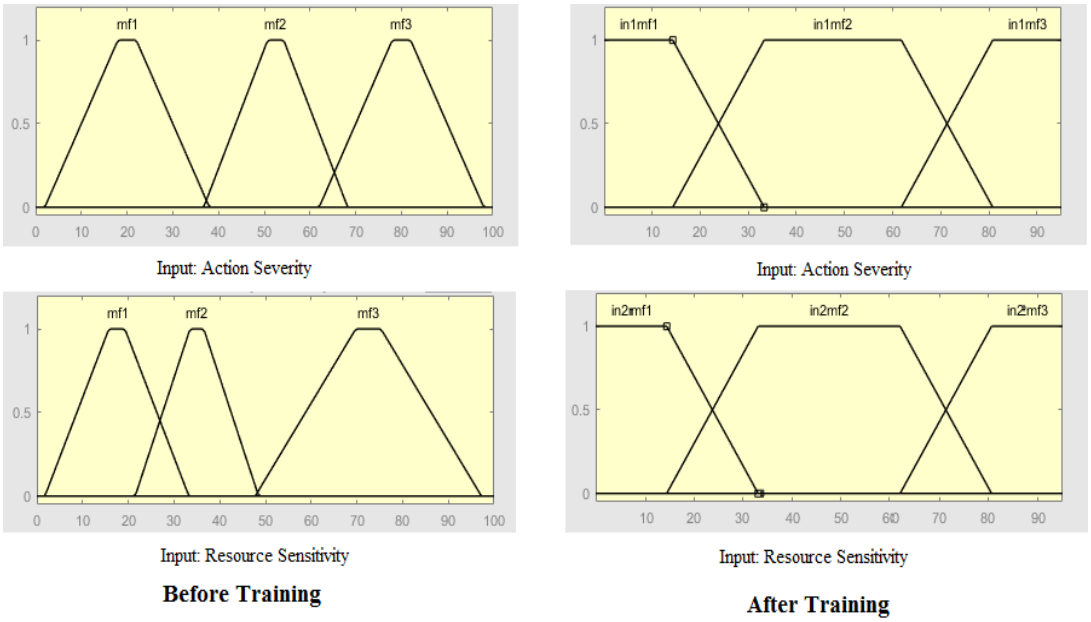


Figure 6.15: Shape of fuzzy sets of the TrapMF before and after the training for action severity and resource sensitivity

After several experiments, the TrapMF with hybrid learning method at 20 epochs was selected as the optimal MF to implement the proposed risk estimation technique, as illustrated in Table 6.4. Figure 6.15 shows the effect of the training on the shape of the MF. It shows the TrapMF of the action severity and resource sensitivity before and after 20 epochs of training using the hybrid learning method. It is clear that significant modifications have been done in the shapes of MFs through the learning process. The same scenario can be seen in Figure 6.16, which represents the TrapMF of the user context and risk history before and after the training process. In addition, Figure 6.17 shows the effect of the training on fuzzy rules and the output risk value in which the output risk was 60 before the training and become 55.2 after the training for the same input combinations.

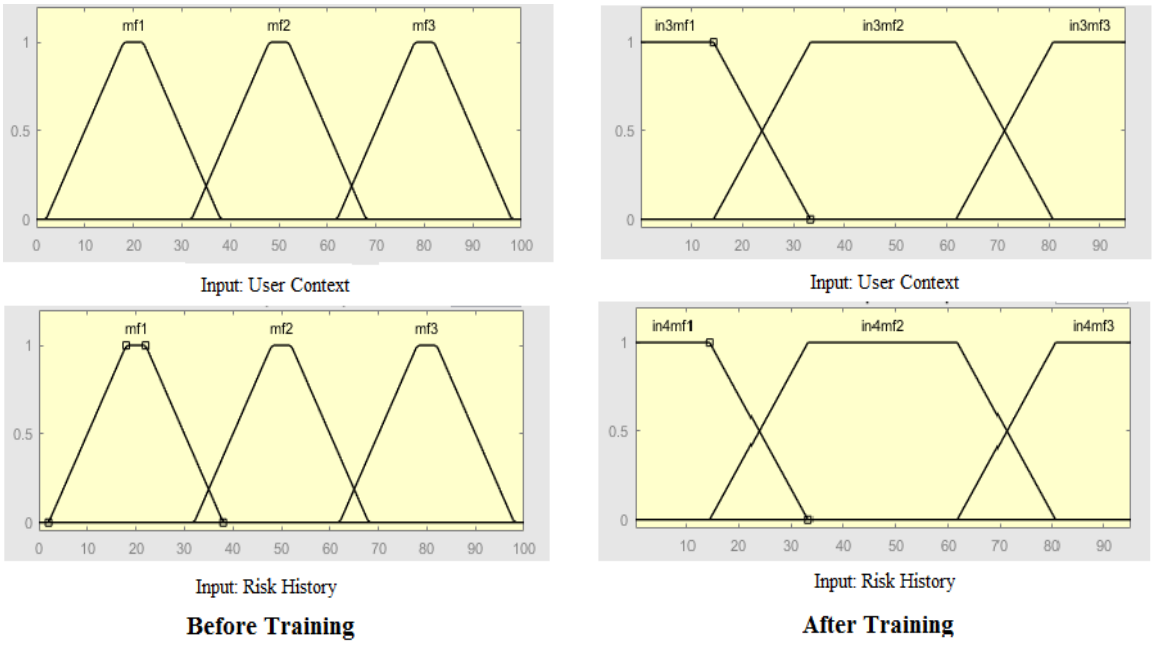


Figure 6.16: Shape of fuzzy sets of the TrapMF of the user context and risk history before and after the training process

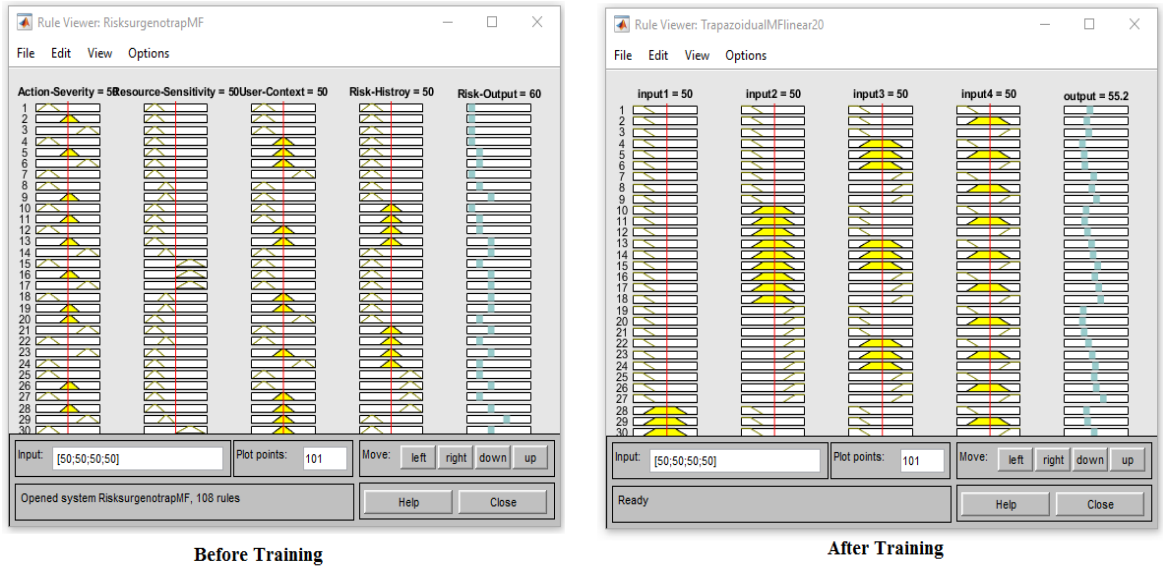


Figure 6.17: Fuzzy rules of the TrapMF before and after the training process

6.4 Summary

Chapter 6 has presented the implementation of the risk estimation process using the ANFIS. The ANFIS is considered the first integrated hybrid neuro-fuzzy model that integrates the benefits of the ANN and the fuzzy logic system. The key objective of the ANFIS is to optimize the parameters of the fuzzy logic system by applying a learning algorithm using input-output datasets. This optimization is done in a way that the error measure between the target and the actual output is minimized. To tune different MFs and add the learning capability, the ANFIS has been adopted to implement the risk estimation technique of the proposed risk-based model. Several experiments were carried out using two separate datasets: training dataset (112,000 records) and checking dataset

(48,000 records) to train and verify the accuracy of the trained ANFIS model. Both hybrid and backpropagation learning methods were utilized to train the ANFIS model with different MFs at three different number of epochs; 20, 100, and 300. The results have demonstrated that the TrapMF with the hybrid learning method at 20 epochs is the optimal combination to implement the ANFIS model of the proposed risk estimation technique as it produced the lowest RMSE and MAE values and the best fit with the learning process with a correlation of 0.9731, which shows that the ANFIS model is well trained. The next chapter presents the implementation of the risk estimation process using the NFS technique.

Chapter 7: Implementation of Risk Estimation using NFS

This chapter provides a discussion of the implementation of the proposed risk estimation technique using the NFS. It starts by providing an overview of the NFS by highlighting the main objectives and types of NFS methods. Then, section 7.2 presents the implementation of the risk estimation technique using the NFS by showing different experimental results of training the NFS model using four learning algorithms to determine the learning method with the lowest error and the best fit with the learning process. Section 7.3 compares the results of the NFS with the fuzzy logic system. Then, section 7.4 compares the results of the NFS with the ANFIS. The chapter closes by providing a summary of the main points discussed through the chapter and introduces the next chapter.

7.1 An Overview of NFS

NFS is the result of integrating the ANN with the fuzzy logic system. It integrates the human-like reasoning of fuzzy logic systems with the learning and connectionist of the ANN (Jang, 1993). Several NFS models were implemented successfully in various social and technological applications. The NFS provides powerful and flexible universal approximations with the capability to recognize interpretable IF-THEN rules (Kar et al., 2014).

The NFS is simply a fuzzy logic system that is trained by a learning algorithm derived from the ANN theory. Although the ANN and the fuzzy logic system have advantages and strengths as independent systems, their drawbacks motivated several researchers to develop a hybrid NFS that reduces these drawbacks. One of the most important advantages of the ANN is the capability to learn from examples, however, it is hard to prove that the ANN is working as expected. In addition, it is like a "black box" to the user in which the method for obtaining the output is not revealed (Jang et al., 1997).

On the other hand, the fuzzy logic system is easy to build and understand by using linguistic expressions to resolve imprecise information (Gray & MacDonell, 1997; Jang et al., 1997). However,

it is not easy to guarantee that a fuzzy logic system with a number of complex rules will provide an appropriate degree of meaningfulness. Also, the fuzzy logic system uses static fuzzy rules that lack the adaptability to resolve unpredicted changes in the environment (Gray & MacDonell, 1997).

The integration of the ANN with the fuzzy logic system resolved some of these issues. The resultant hybrid NFS combines parallel computation and learning abilities of the ANN with the human reasoning of fuzzy systems and clarity of systems representation. Therefore, the ANN becomes more transparent and the fuzzy logic system becomes capable of learning (Shaf et al., 2016).

7.1.1 Multilayer Perceptron Model

MuliLayer Perceptron (MLP) model is a feed-forward ANN. It is the most common and widely used ANN model in various applications (Okut, 2016). The MLP is used to explore complex and nonlinear models. It is based on a supervised learning technique that needs the desired output for each input to be known to calculate the error (Werbos, 1974).

Typically, the MLP model consists of three layers; input layer, hidden layer, and output layer, as depicted in Figure 7.1. The input layer represents input variables of the system by a circle (neuron) for each variable. While the output variable is represented by a single circle in the output layer. The middle layer is the hidden layer that is not visible to the outside. This layer is responsible for carrying out intermediate computations. Deciding the number of hidden layers, hidden neurons and type of transfer function plays an important role in implementing an efficient MLP model (White, 1992).

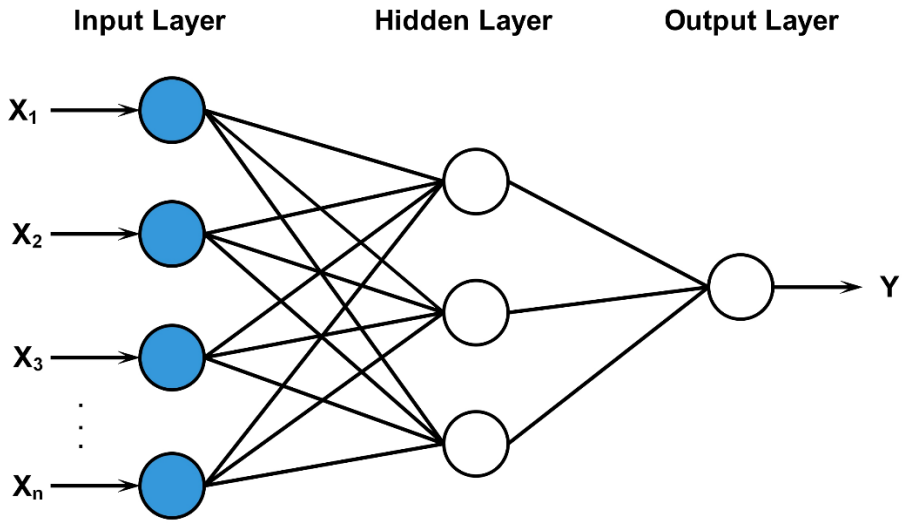


Figure 7.1: Layers of the MTP model (Okut, 2016)

The supervised learning is used repeatedly to adjust the weight of each connection to produce accurate output (Kohonen, 1982). This is achieved by using the backpropagation learning algorithm that estimates the derivatives of the network's error with respect to all of its weights and adjust the weights to yield a small error, where the error is the difference between the network's output and the target output for the same input (Viharos & Kis, 2015).

Building an NFS model require employing one of the common models of the ANN. This research selected MLP model to implement the NFS model of the risk estimation technique, as this feed-forward model is efficient and commonly used in various applications.

7.1.2 Types of NFS

Generally, the term NFS refers to all systems that resulted from the integration of the ANN with the fuzzy logic system. This integration can be done in three different ways; cooperative, concurrent, and hybrid. This section provides a brief discussion of each type.

7.1.2.1 Cooperative NFS

Cooperative NFS is used to describe the integration of the ANN with the fuzzy logic system in which the ANN is used to tune the fuzzy logic system without changing the functionality of the variables. In other words, the ANN is used as a pre-processing stage in which the ANN learning algorithm is used to determine some of the fuzzy logic variables. For example, clustering algorithms can be used to determine fuzzy sets and fuzzy rules. After determining variables of the fuzzy logic system through the learning algorithm, the ANN is removed and the fuzzy logic system works on its own (Vieira et al., 2004). Hence, the ANN is used only in the initial stage of the fuzzy logic system (Abraham, 2001). The architecture of the cooperative NFS is shown in Figure 7.2.

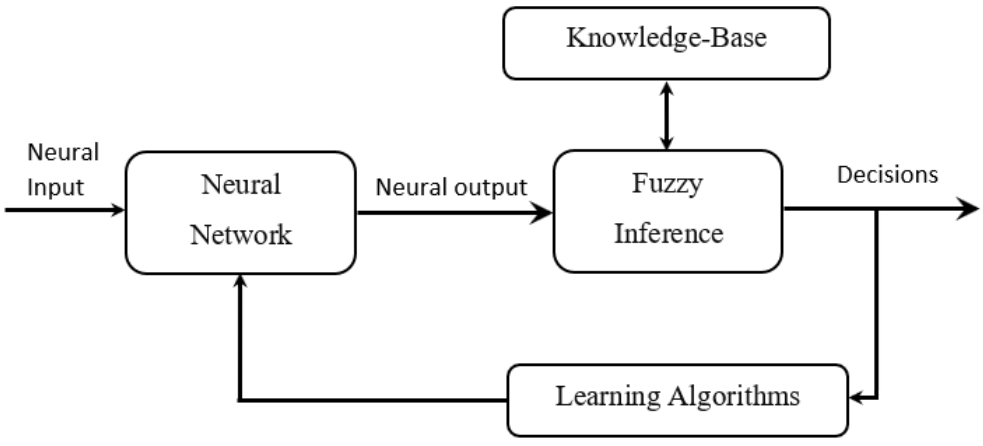


Figure 7.2: Cooperative NFS (Vieira et al., 2004)

7.1.2.2 Concurrent NFS

Concurrent NFS refers to the system where the ANN and the fuzzy logic system work together in which the inputs entered the fuzzy logic system are pre-processed and then the ANN processes the outputs of the concurrent system or in a reverse way, as depicted in Figure 7.3. One of the disadvantages associated with the concurrent NFS is that the results cannot be interpreted completely (Vieira et al., 2004). Moreover, the weights are substituted by MFs in which the result of each

weighting process is a membership value of the corresponding input in the fuzzy set (Naidu & Sun, 1997).

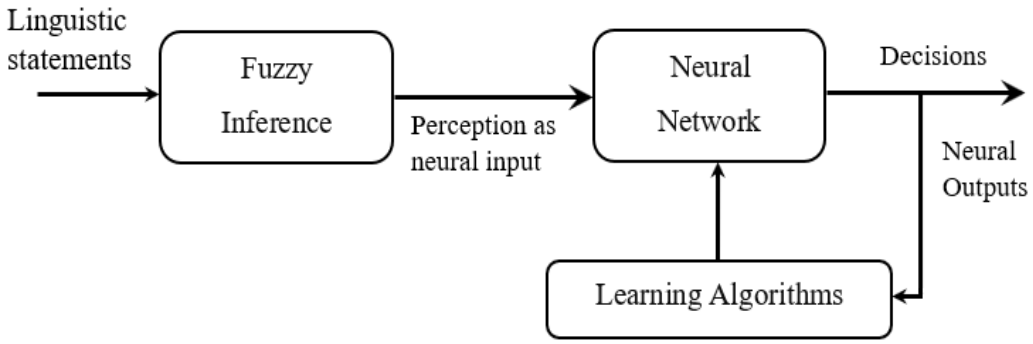


Figure 7.3: Concurrent NFS (Vieira et al., 2004)

7.1.2.3 Hybrid NFS

Nauck et al. (1997) defined the hybrid NFS as: "A fuzzy system that uses a learning algorithm based on gradients or inspired by neural networks theory (heuristic learning strategies) to determine its parameters (fuzzy sets and fuzzy rules) through the patterns processing (input and output)".

In the hybrid NFS, both fuzzy logic and ANN models are used independently, in which each model is used to perform a certain task in the system to reach a common target. The concept of the hybrid NFS refers to the explanation of the fuzzy logic system with respect to the ANN. Hence, fuzzy sets can be interpreted as weights, and fuzzy rules, input and output variables can be interpreted as neurons. In other words, one of the advantages of hybrid NFS is its architecture since both fuzzy system and neural network do not have to communicate any more with each other. They are one fully fused entity (Abraham, 2001).

There are several different ways to develop hybrid neuro fuzzy systems, therefore, there are various models which are built based on context. These models are similar in its essence, but they present basic differences. Many types of NFS are represented by neural networks that implement logical functions. This is not necessary for the application of a learning algorithm into a fuzzy system, however, the representation truth a neural network is more convenient because it allows to visualise the flow of data through the system and the error signals that are used to update its parameters. Although hybrid NFS models are different, they are similar in their core (Vieira et al., 2004).

7.2 Implementation of NFS

The hybrid NFS was utilized to implement the risk estimation technique of the proposed risk-based access control model. The fuzzy logic system with expert judgment was utilized to implement the risk estimation technique. The information collected from IoT security experts was utilized to confirm and build accurate fuzzy rules, as discussed in section 5.3.3.2. The results demonstrated that

combining the fuzzy logic system with expert judgment can provide accurate and realistic results in assessing security risks of access control operation. However, an access control model for the IoT system serves thousands of users. In addition, the scalability of the fuzzy logic system seems to be questionable. Therefore, the NFS was utilized to reduce the processing time for each access request by using the parallel computation of the ANN and add the learning capability to the proposed risk estimation technique to adapt to changes of the IoT environment.

Implementing a hybrid NFS is performed in two separate stages. The fuzzy logic system is first implemented, and the database built. Then, the ANN will use this dataset to train the system and improve the performance. Since the proposed risk estimation technique using the fuzzy logic system was implemented previously, as discussed in chapter 5, this chapter focuses only on using the dataset created from the fuzzy logic system for training the NFS model.

Implementing the NFS model of the proposed risk estimation technique consists of three layers; input layer, hidden layer, and output layer. The input layer involves input risk factors; user context, resource sensitivity, action severity, and risk history, as depicted in Figure 7.4. The output layer represents the output risk value resulted from the risk estimation process. The middle layer is the hidden layer that is responsible for carrying out computations and updating weights between different connections.

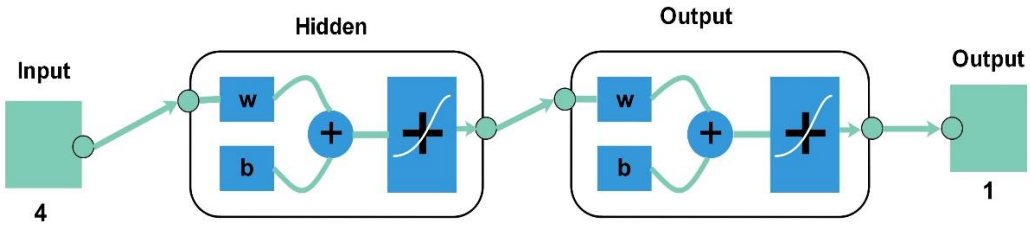


Figure 7.4: Implementation of the NFS model of the proposed risk estimation technique

One of the challenges associated with implementing the NFS model of the risk estimation technique is to determine the appropriate number of hidden layers and the appropriate number of neurons for each hidden layer. The number of hidden layers needed for the NFS model depends on the complexity of the relationship between the input and the target parameters. It represents a major impact on the learning process. However, a Feed-Forward Back Propagation (FFBP) network encompassing of more than one hidden layer is very rare (Konaté et al., 2015). Hornik, Stinchcombe and White (1989) have proved that an FFBP network with one hidden layer is enough for most problems in various applications.

In addition, determining the optimal number of neurons in the hidden layer plays a significant role in the implementation of the NFS model. If an insufficient number of neurons are used, the NFS model will be unable to model complicated data, and the resulting fit will be poor. While using a large number of neurons in the hidden layer affects its performance on new data and its ability to

provide a generalized model will be compromised (Abraham, 2005). Indeed, increasing the number of neurons ensures correct training, but it also affects the NFS performance. Therefore, a compromise needs to be reached between too many and too few neurons in the hidden layer.

7.2.1 Data Collection

Implementing the NFS model of the proposed risk estimation technique requires having a dataset or examples for training. After implementing the fuzzy logic system with the help of IoT security experts, a dataset consisting of 160,000 records was created. To avoid possible bias in the sample data to the NFS, the dataset was randomized and divided into three sets using the cross−validation method.

- **Training set:** This set contains 96,000 records (60% of the dataset) to train the NFS model.
- **Testing set:** This set contains 32,000 records (20% of the dataset) to test the NFS model.
- **Validation set**: This set contains 32,000 records (20% of the dataset) to validate the NFS model.

7.2.2 Experimental Results

Implementing the NFS model of the proposed risk estimation technique require determining the number of hidden layers and the number of neurons in the hidden layers. Based on the literature, one hidden layer is sufficient for most problems (Al-Hmouz et al., 2012; Hornik et al., 1989). Therefore, one hidden layer was utilized to implement the NFS model of the risk estimation technique. To determine the appropriate number of neurons in the hidden layer, the NFS model was trained using four learning algorithms. Several experiments were carried out and Mean Square Error (MSE), RMSE, and R values of training, testing, and validation were utilized to determine the appropriate number of neurons and the best learning algorithm. After determining the appropriate number of neurons in the hidden layer, the NFS model of the risk estimation technique was trained, and then the trained model was tested with different numbers of access requests in term of processing time.

All experiments were carried out on Windows 7 (64-bit) operating system with an i7 processor and 16 GB RAM. All training functions and experiments were coded and executed using MATLAB and ANN toolbox.

7.2.2.1 Performance Evaluation

The commonly used performance evaluation measures in forecasting problems were utilized to compare and evaluate the accuracy of the NFS model (Cerezuela-Escudero et al., 2016). The NFS model was trained, and the performance was observed using MSE, RMSE, and R. The number of

neurons in the hidden layer with the lowest MSE and RMSE and the highest R was selected to implement the NFS model of the risk estimation technique. Since the RMSE and R were introduced in the previous chapter in section 6.2.2.1, this section discusses only the MSE.

MSE measures the average of the squares of the errors which help to understand and interpret the difference between the observed and predicted values. It acts as an indicator to measure how near a fit line is to data points. The smaller the MSE, the nearer the fit is with the data points (Konaté et al., 2015).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(O_i - P_i)^2 \qquad (7.1)$$

Where n is the total number of data, O_i is the observed (target) value, and P_i is the predicted value.

7.2.2.2 Training Algorithms

To reach network generalization and good fit with all the data points, the NFS model of the risk estimation technique was trained using four learning algorithms; Levenberg-Marquardt (trainlm), Bayesian Regulation (trainbr), Conjugate Gradient with Fletcher-Reeves Resrarts (traincgf), and Scaled Conjugate Gradient (trainscg) to determine the optimal learning algorithm that guarantees network generalization with the minimum error (lowest RMSE and MSE) and the maximum fit (highest R). These training algorithms are decided based on the literature that assure that these algorithms are the most common and can work with different context (Al-Hmouz et al., 2012; Hornik et al., 1989).

This section discusses the results of each learning algorithm. It starts by providing an overview of each learning algorithm, then presents the results with increasing number of neurons.

7.2.2.2.1 Levenberg-Marquardt

Levenberg-Marquardt (LM) algorithm is an iterative method that locates a local minimum of a multivariate function. It is expressed as the sum of squares of several non-linear and real-valued functions. The LM algorithm is widely adopted in various disciplines to deal with data-fitting applications. It has also become a standard method for nonlinear least-squares problems. The LM learning algorithm can be considered as a combination of steepest descent and the Gauss-Newton method (Lourakis & Argyros, 2005). It also considered the fastest learning algorithm. The only limitation associated with this algorithm is that it consumes more memory (Demuth & Beale, 1998; Pramanik & Panda, 2009).

The LM algorithm with one hidden layer was utilized to train the NFS model of the proposed risk estimation technique. Several experiments were carried out to determine the appropriate number of neurons that produces the lowest error and the best fit with the learning process. The NFS model was

trained using the LM learning algorithm with increasing the number of neurons in the hidden layer from 20 to 1000 and MSE and RMSE values were observed, as depicted in Figure 7.5 and Figure 7.6. The three lines representing training, testing, and validation data are almost identical and have the same behaviour. The MSE and RMSE values dramatically decreased when increasing the number of neurons from 20 to 100. The decrease in the MSE continued to reach its lowest value with 0.977 for training, 1.22 for validation, and 1.19 for testing data at 1000 neurons. Similarly, the RMSE reached its lowest value at 1000 neurons with 0.989 for training, 1.10 for validation, and 1.09 for testing data.

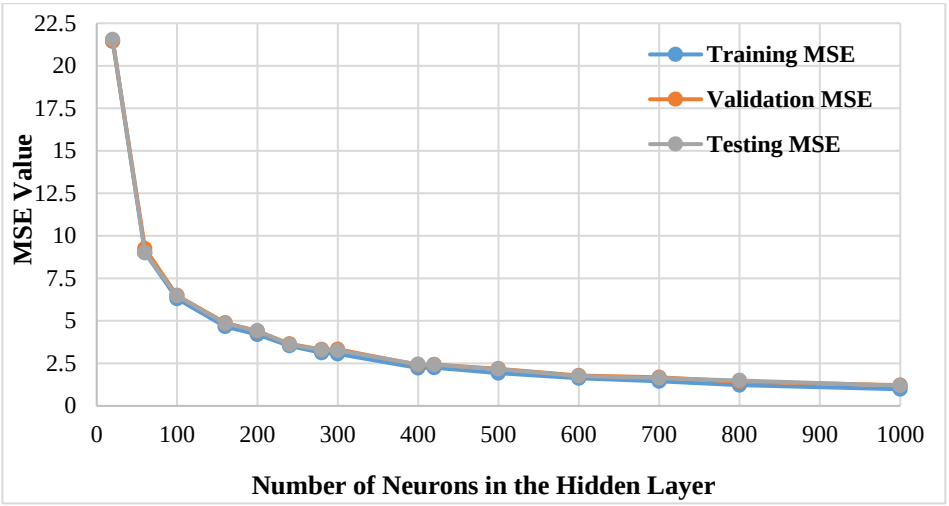


Figure 7.5: MSE of training, validation, and testing data of the NFS model with the LM learning algorithm when increasing the number of neurons in the hidden layer

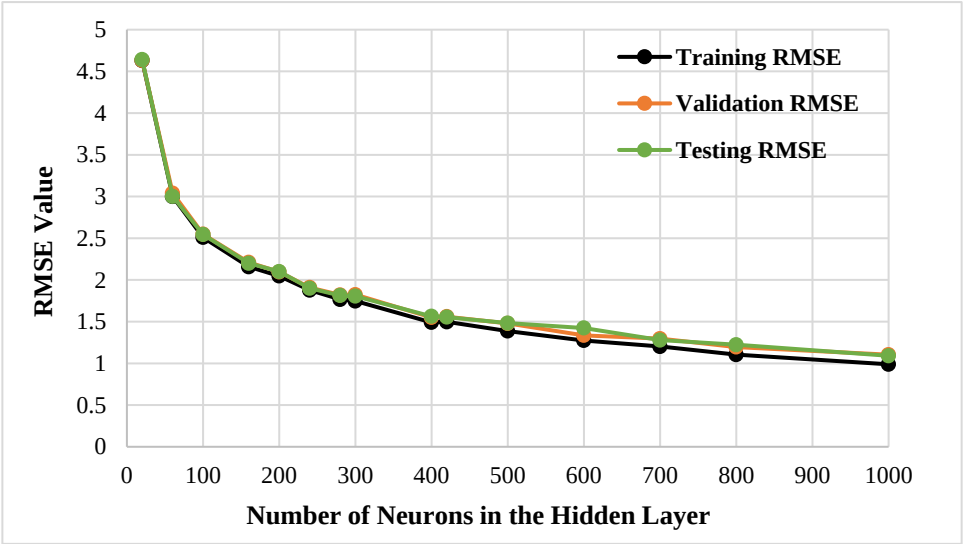


Figure 7.6: RMSE of training, validation, and testing data of the NFS model using the LM learning algorithm when increasing the number of neurons in the hidden layer

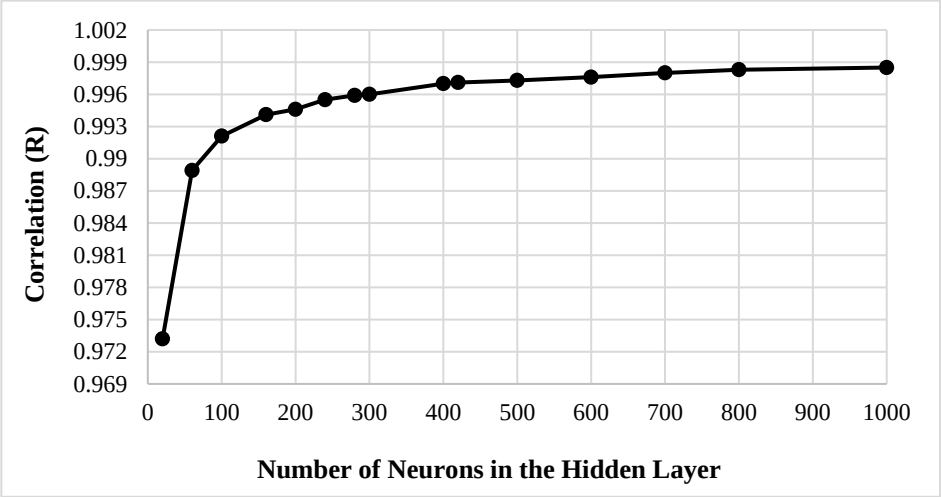


Figure 7.7: Value of R when increasing the number of neurons in the hidden layer with the LM learning algorithm

In addition, Figure 7.7 shows the value of R when increasing the number of neurons in the hidden layer from 20 to 1000 neurons. It increased dramatically from 0.973 at 20 neurons to reach 0.995 at 200 neurons. This increase continued to reach its maximum value, 0.999, at 1000 neurons.

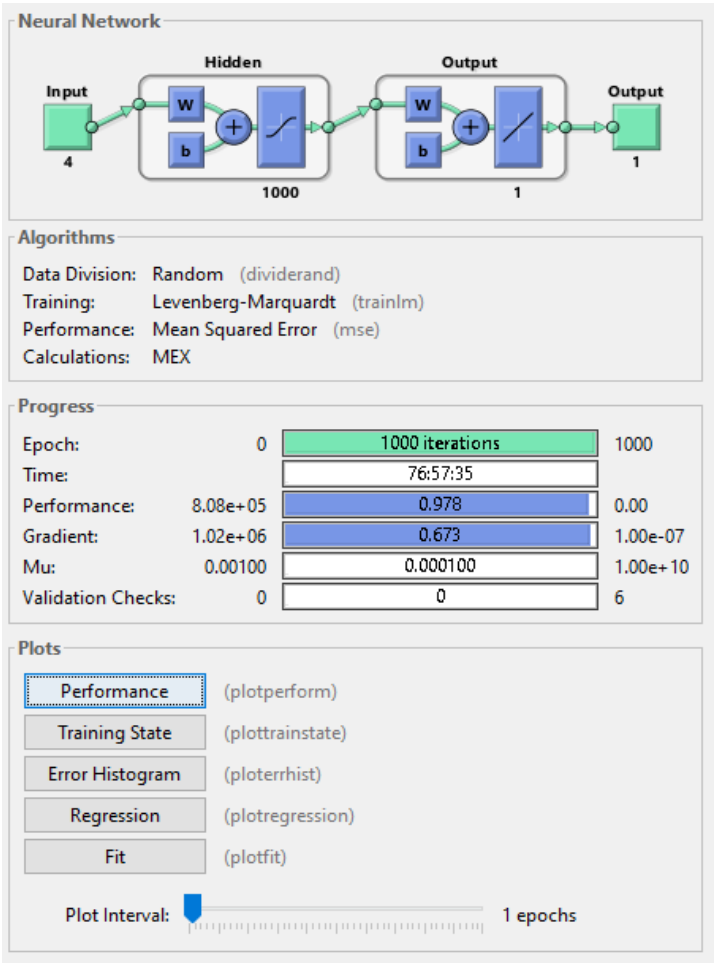


Figure 7.8: Training the NFS model with 1000 neurons in the hidden layer using the LM learning algorithm

With the LM learning algorithm, the results demonstrated that increasing the number of neurons in the hidden layer led to decreasing both MSE and RMSE values for training, testing, and validation

data. It also showed that increasing the number of neurons led to increasing the value of R. However, training the NFS model of the proposed risk estimation technique with a large number of neurons takes a long time. The last experiment with 1000 neurons took more than 77 hours. Therefore, 1000 neurons in the hidden layer were considered as the appropriate number of neurons to implement the NFS model of the proposed risk estimation technique. The results demonstrated that the NFS model has the lowest MSE and RMSE values for training, testing, and validation at 1000 neurons. Also, the NFS model has the highest value of R, 0.999, at 1000 neurons, which is an adequate correlation that indicates the NFS model is well trained and fit with the learning process as the value of R is very close to 1.

The NFS model of the proposed risk estimation technique was implemented using the LM learning algorithm with 1000 neurons in the hidden layer, as shown in Figure 7.8. After the NFS model was trained, the performance graph represented MSE values of training, testing, and validation data, as shown in Figure 7.9. The result is reasonable, and the NFS model is a good fit with the learning process with the value of R is 0.999, which is very close to 1. In addition, no overfitting has occurred as training, validation, and testing data have the same behaviour.

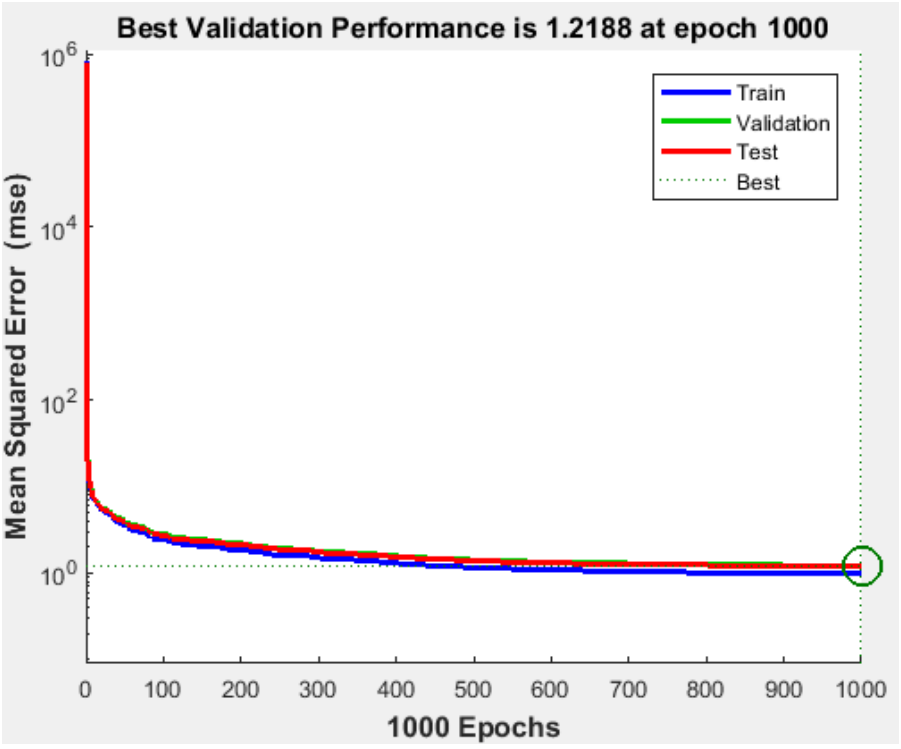


Figure 7.9: Performance of training, validation, and testing data at different number of epochs with the LM learning algorithm at 1000 neurons in the hidden layer.

In addition, Figure 7.10 shows regression plots of the NFS model of the proposed risk estimation technique with respect to targets for training, validation, and testing data. For a perfect fit, the data should fall along a 45-degree line, where the network outputs are equal to the targets. For the NFS model of the proposed risk estimation technique, the fit is reasonably good for all training, validation and testing data with the value of R is 0.999, which is very close to the ideal case.

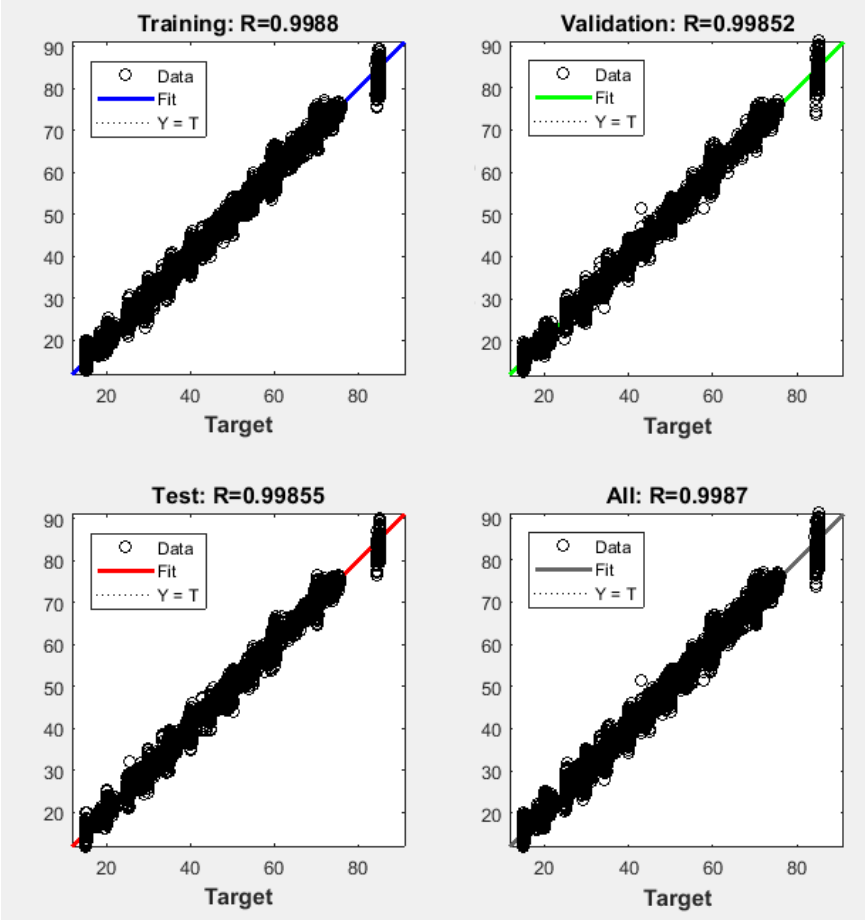


Figure 7.10: Regression plots of training, testing, and validation when applying LM learning algorithm with 1000 neurons in the hidden layer.

7.2.2.2.2 Bayesian Regularization

Generally, regularization techniques are employed with the backpropagation learning method to produce a small error. However, the major problem associated with this mixture is that its convergence is very slow, which can cause overfitting issues (Saini, 2008). Therefore, new backpropagation learning techniques were developed by researchers to overcome the slow convergence issue. In the same way, some regularization techniques were established to resolve the overfitting issue. The LM and Bayesian Regularization (BR) learning methods are examples of these new techniques that are used to produce lower mean squared errors than other techniques especially with functioning approximation problems (Sorich et al., 2003).

The main target of the BR learning algorithm is to utilize the sum of squares and the sum of squared weights to minimize errors and achieve a good generalized model. It updates the weight and bias values using the same optimization method used in the LM learning method (Bishop & Tipping, 1998).

The BR learning algorithm with one hidden layer was utilized to train the NFS model of the proposed risk estimation technique. Several experiments were carried out to determine the appropriate number

of neurons that produces the lowest error and the best fit with the learning process. The NFS model was trained with increasing the number of neurons in the hidden layer from 50 to 900 with observing MSE and RMSE values. One of the important features of the BR learning algorithm is that it does not require a validation dataset separate from the training dataset. Therefore, only values of training and testing error were observed in these experiments, as shown in Figure 7.11 and Figure 7.12. The two lines representing training and testing data are almost identical and have the same behaviour. So, there is no overfitting. The results of training the NFS model using the BR learning method showed unstable behaviour when increasing the number of neurons in the hidden layer. The MSE error of training data dramatically decreased from 9.46 at 50 neurons to reach 6.23 at 100 neurons. This decrease continued to reach 4.86 at 200 neurons. Then, the MSE error of training data increased to reach 5.61 at 300 neurons. The same unstable behaviour continued until reaching the lowest MSE and RMSE values for both training and testing data at 600 neurons. The MSE values were 2.01 and 2.16 for training and testing respectively. Similarly, the RMSE reached its lowest values at 600 neurons with 1.42 for training, 1.47 for testing data.

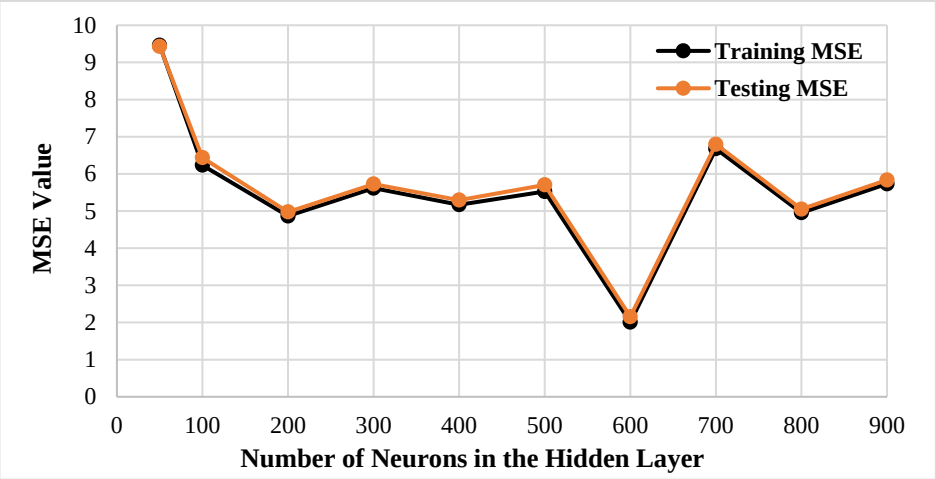


Figure 7.11: MSE of training and testing data of the NFS model when increasing the number of neurons in the hidden layer with the BR learning algorithm

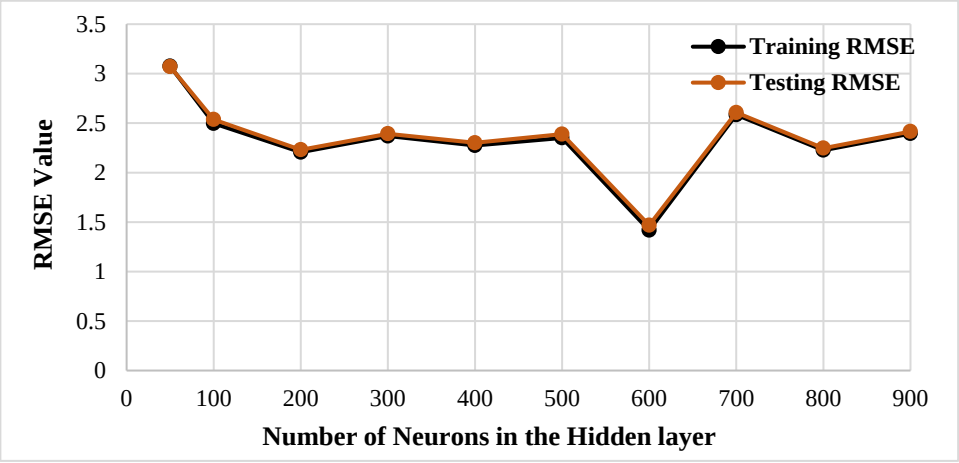


Figure 7.12: RMSE of training and testing data of the NFS model when increasing the number of neurons in the hidden layer with the BR learning algorithm

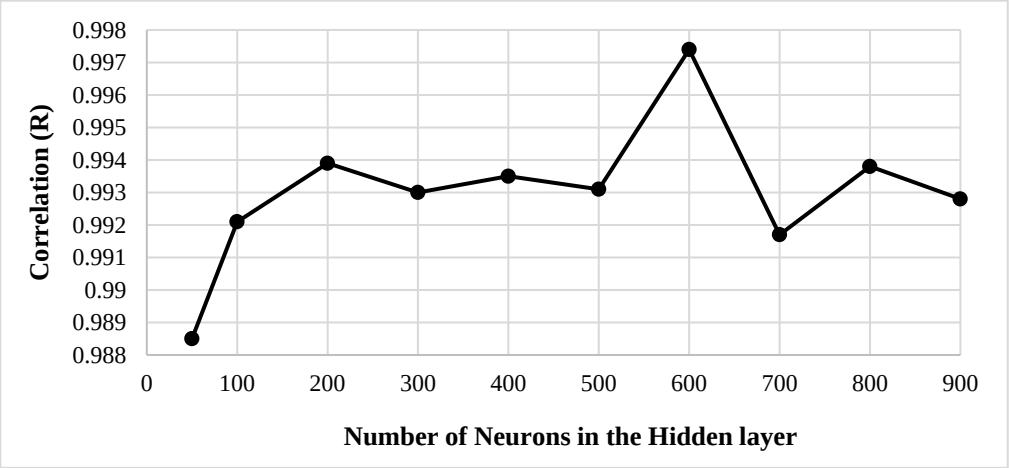


Figure 7.13: Value of R when increasing the number of neurons in the hidden layer with the BR learning algorithm

In addition, Figure 7.13 shows the value of R when increasing the number of neurons in the hidden layer from 50 to 900 neurons. The results demonstrated that the value of R showed unstable behaviour. It increased from 0.989 at 20 neurons to 0.994 at 200 neurons. Then, it decreased at 300, 400, and 500 neurons. But at 600 neurons, the value of R reached its highest value which is 0.997.

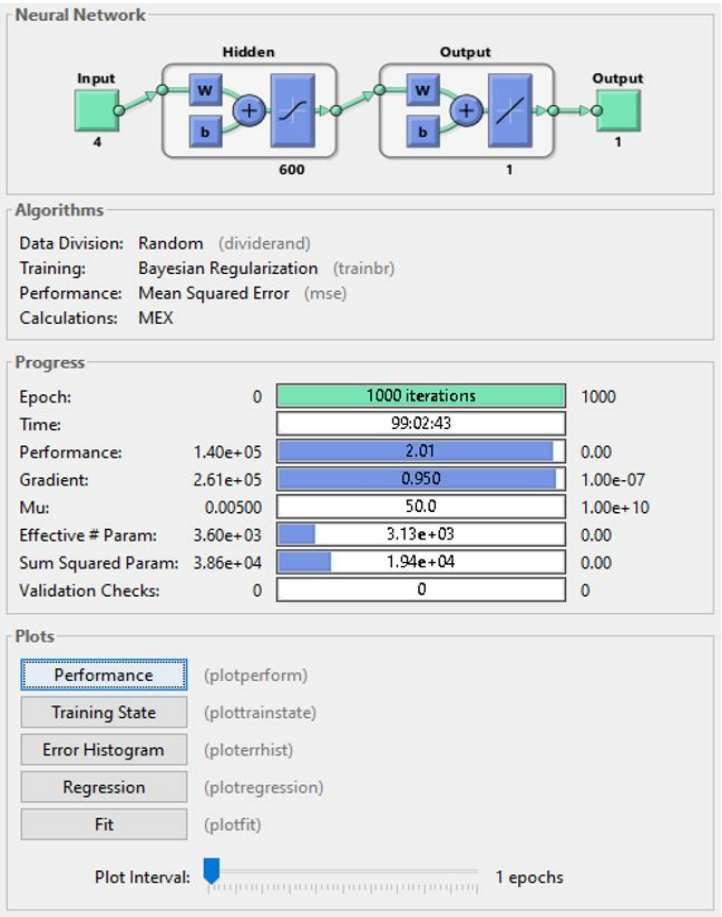


Figure 7.14: Training the NFS model at 600 neurons in the hidden layer using the BR learning algorithm

With the BR learning algorithm, the results demonstrated that increasing the number of neurons in the hidden layer led to unstable behaviour for both training and testing error values. It showed that

the appropriate number of neurons in the hidden layer to implement the NFS model of the proposed risk estimation technique is 600 where it produced the lowest MSE and RMSE values for both training and testing data. It also produced the highest value of R (0.997) at 600 neurons, which indicates that the NFS model is well trained as the value of R is very close to 1.

The NFS model of the proposed risk estimation technique was implemented using the BR learning algorithm with 600 neurons in the hidden layer, as shown in Figure 7.14. After the NFS model was trained, the performance graph showed MSE values of training and testing data, as shown in Figure 7.15. The NFS model is a good fit with the learning process as the value of R is very close to 1. In addition, no overfitting has occurred as training and testing data have the same behaviour.

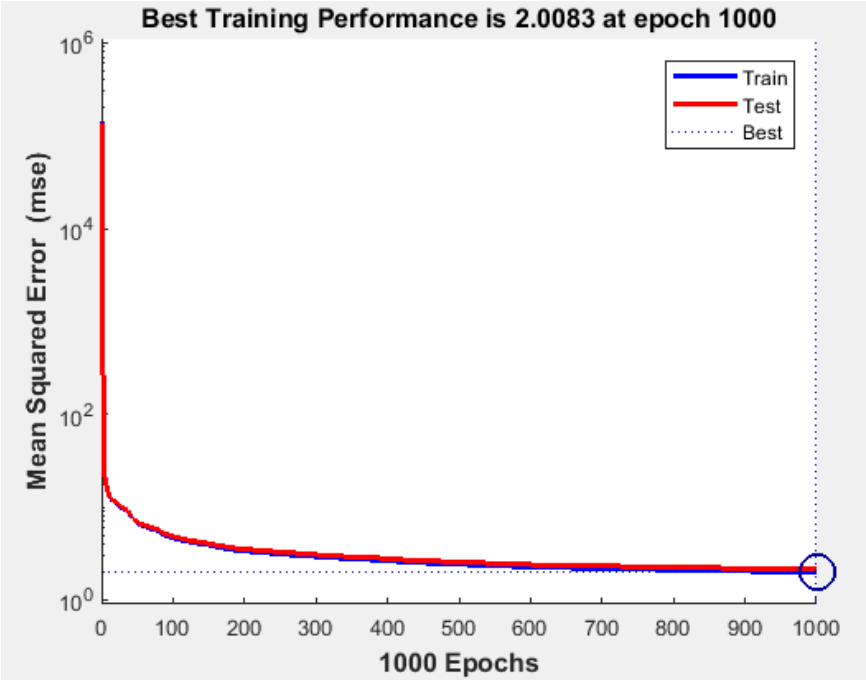


Figure 7.15: Performance of training and testing data at different number of epochs with the BR learning algorithm at 600 neurons in the hidden layer.

In addition, Figure 7.16 shows regression plots of the NFS model of the proposed risk estimation technique with respect to targets for training and testing data. The fit with the learning process is reasonably good for all training and testing data with 0.997 in the value of R, which is very close to the ideal case.

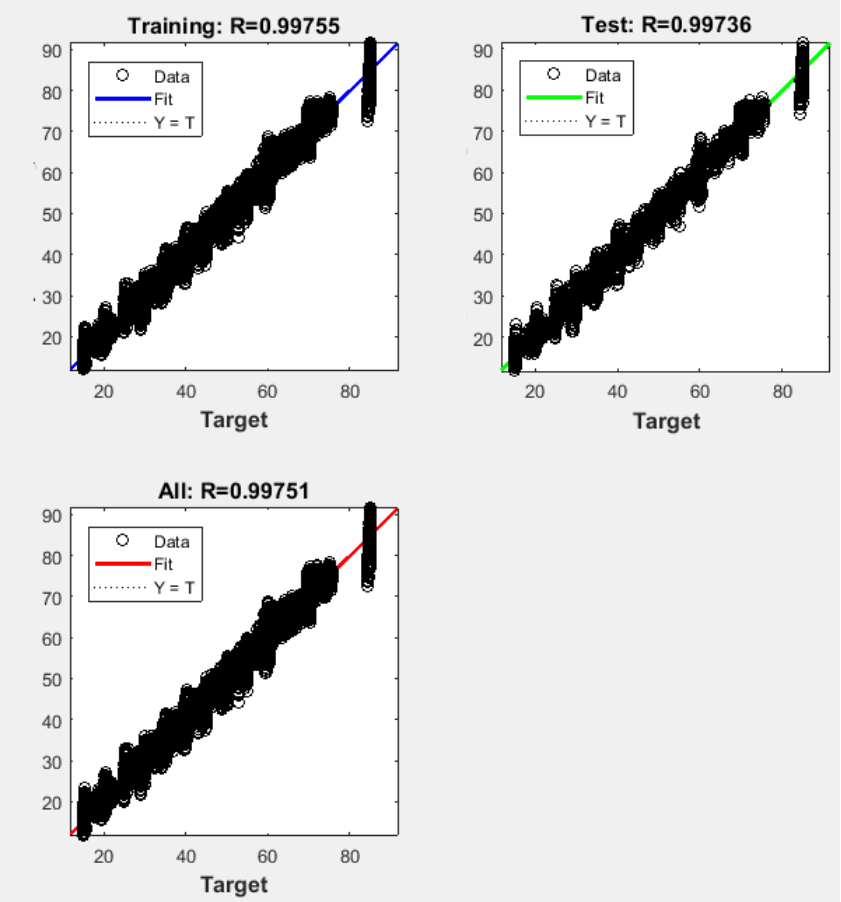


Figure 7.16: Regression plots of training and testing data when applying the BR learning algorithm with 600 neurons in the hidden layer

7.2.2.2.3 Conjugate Gradient with Fletcher-Reeves

Gradient-based learning algorithm is one of the most commonly used error minimization techniques. It is a gradient descent local optimization algorithm that includes the backward error correction of the network weights (Nawi et al., 2010). Conjugate gradient algorithm is one of the backpropagation techniques used to train multilayer ANN in a supervised way. It updates weight and bias values based on the conjugate gradient backpropagation with Fletcher-Reeves updates (Fletcher & Reeves, 1964). Therefore, it is called Conjugate Gradient with Fletcher-Reeves (CGF) learning method. The conjugate gradient algorithms are usually much faster than variable learning rate backpropagation. However, they require more storage than simple algorithms, so they are often a good choice for networks with a large number of weights (Ellah et al., 2015).

The CGF learning algorithm with one hidden layer was utilized to train the NFS model of the proposed risk estimation technique. Several experiments were carried out to determine the appropriate number of neurons that produces the lowest error and the best fit with the learning process. The NFS model was trained with increasing the number of neurons in the hidden layer from 50 to 1200 with observing MSE and RMSE values, as shown in Figure 7.17 and Figure 7.18. The three lines representing training, validation, and testing data are almost identical and have the same

behaviour. So, there is no overfitting. The results of training the NFS model using the CGF learning method showed unstable behaviour when increasing the number of neurons in the hidden layer. The MSE value of the training data decreased from 25.27 at 50 neurons to reach 21.04 at 100 neurons. This decrease continued to reach 20.59 at 200 neurons. Then, the MSE value of the training data increased to reach 26.75 at 300 neurons. Then, the MSE reached its lowest value at 400 neurons with 19.25, 19.54 and 19.14 for training, validation, and testing data respectively. Increasing the number of neurons in the hidden layer from 400 to 1200 showed the same inconsistent behaviour. However, the MSE value at 400 neurons produced the lowest error. This was the same scenario for the RMSE value for training, validation, and testing data where it produced the lowest RMSE values at 400 neurons with 4.39 for training, 4.42 for validation, and 4.37 for testing data.

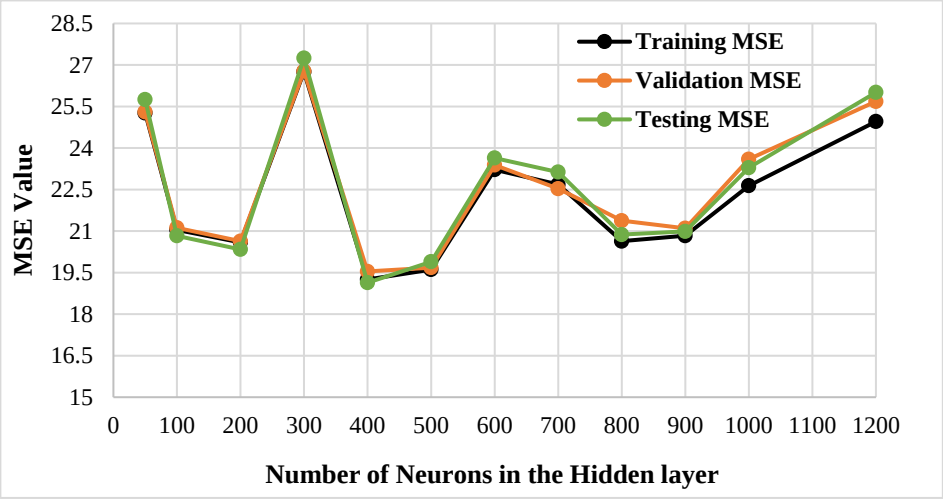


Figure 7.17: MSE of training, validation, and testing data of the NFS model when increasing the number of neurons in the hidden layer with the CGF learning algorithm

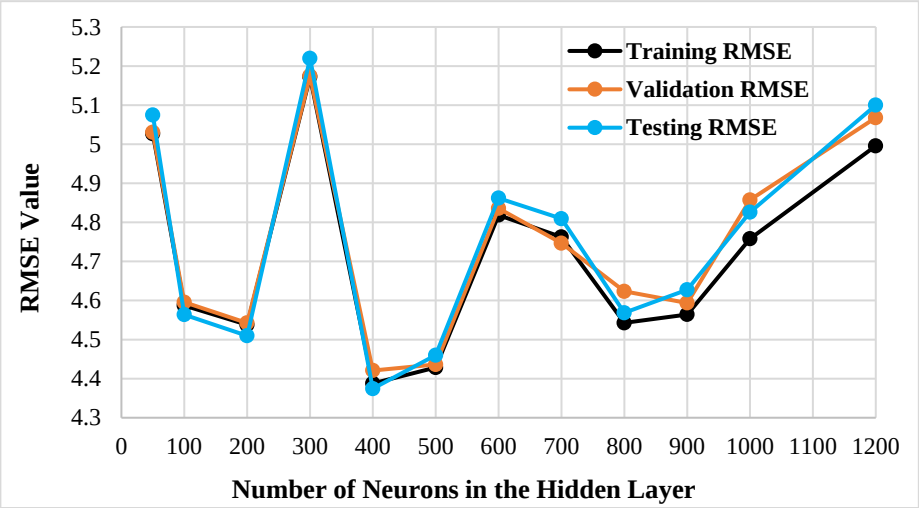


Figure 7.18: RMSE of training, validation, and testing data of the NFS model when increasing the number of neurons in the hidden layer with the CGF learning algorithm

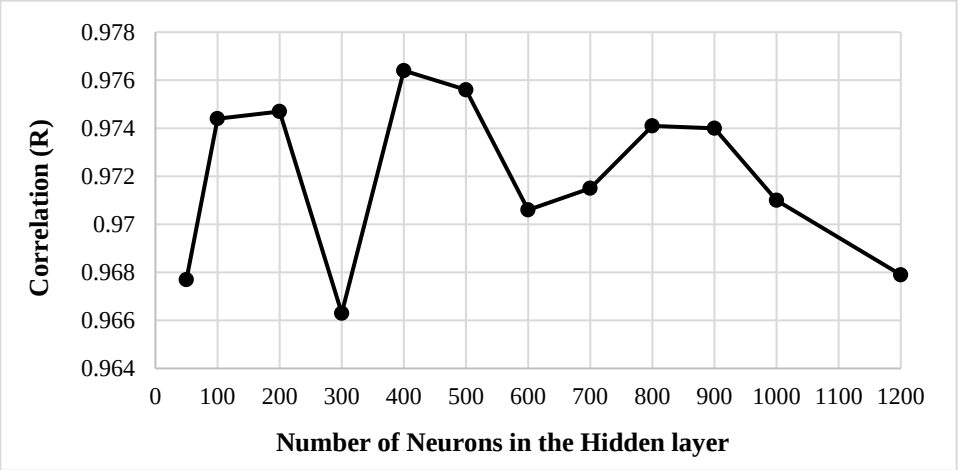


Figure 7.19: Value of R when increasing the number of neurons in the hidden layer with the CGF learning algorithm

In addition, Figure 7.19 shows the value of R when increasing the number of neurons in the hidden layer from 50 to 1200 neurons. The chart has the same behaviour of MSE and RMSE values in which the lower the error, the higher the correlation. The value of R is increased from 0.968 at 50 neurons to reach 0.975 at 200 neurons. Then, it reached its lowest value at 300 neurons. The highest correlation for the NFS model with the CGF learning algorithm was realized when applying 400 neurons in the hidden layer.

With the CGF learning algorithm, the results demonstrated that increasing the number of neurons in the hidden layer led to inconsistent behaviour of training, validation and testing error values. It showed that the appropriate number of neurons in the hidden layer to implement the NFS model of the proposed risk estimation technique is 400 neurons where it produced the lowest MSE and RMSE error for training, validation, and testing data. It also produced the highest value of R, 0.976, which indicates that the NFS model is well trained.

The NFS model of the proposed risk estimation technique was implemented using the CGF learning algorithm with 400 neurons in the hidden layer, as shown in Figure 7.20. After the NFS model was trained, the performance graph showed MSE values of training, validation, and testing data, as shown in Figure 7.21. The NFS model is a good fit as R value is close to 1. In addition, no overfitting has occurred as training, validation, and testing data have the same behaviour. However, the lowest error produced with the CGF learning algorithm is quite high.

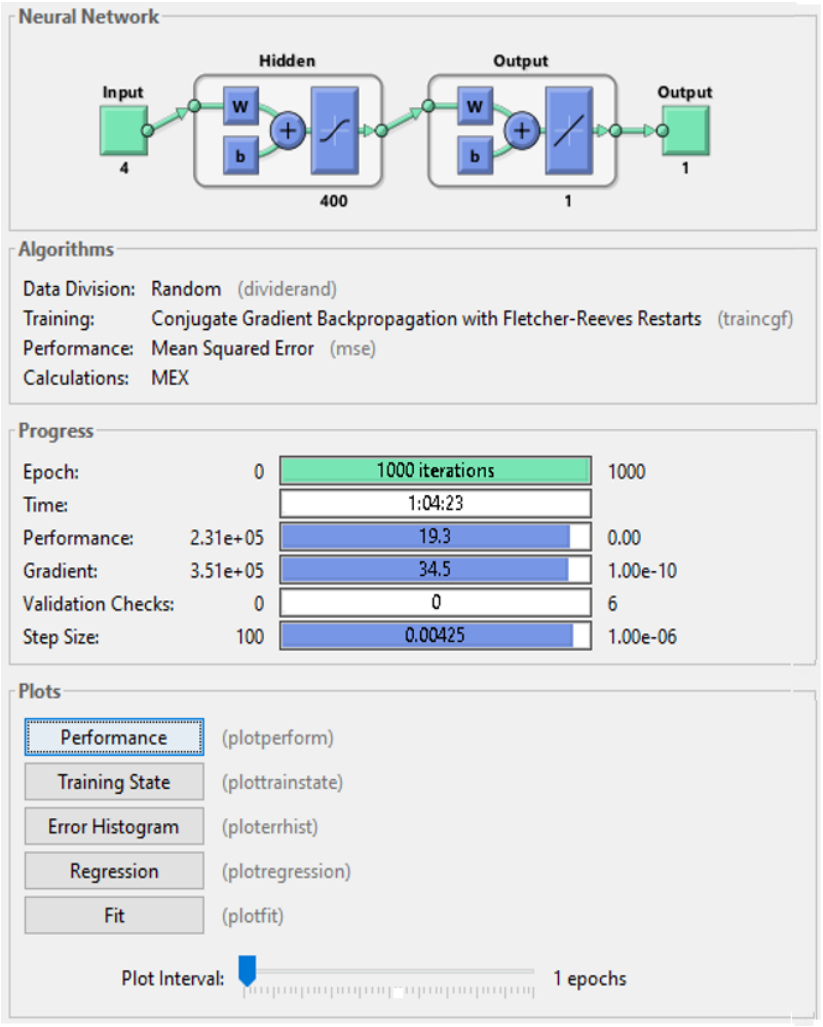


Figure 7.20: Training the NFS model with 400 neurons in the hidden layer using the CGF learning algorithm

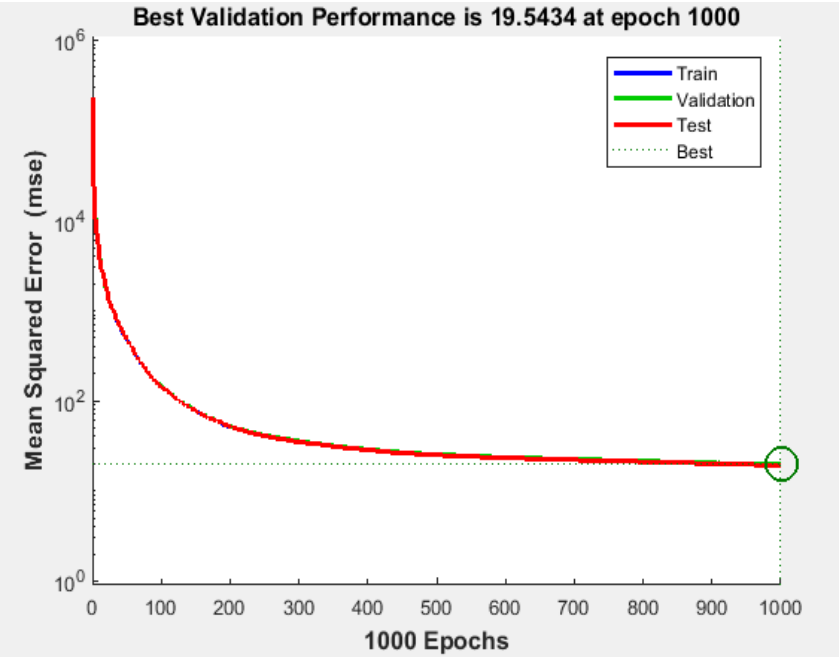


Figure 7.21: Performance of training, validation, and testing data at different number of epochs using the CGF learning algorithm with 400 neurons in the hidden layer

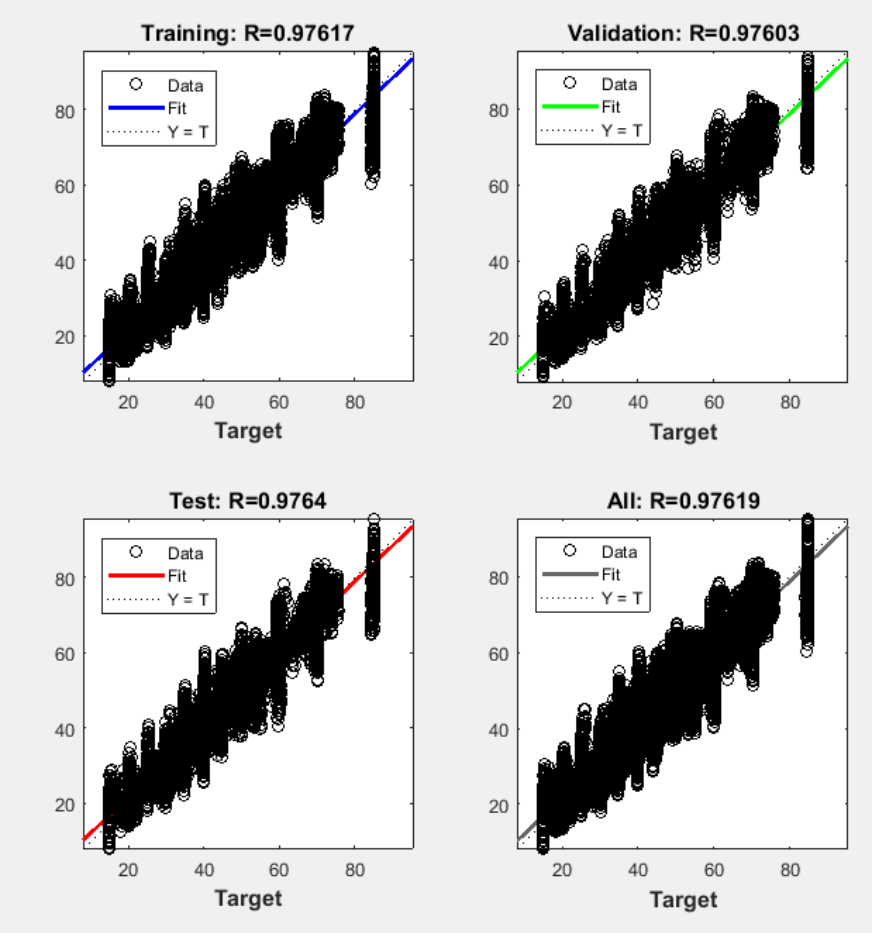


Figure 7.22: Regression plots of training, validation, and testing data when applying the CGF learning algorithm with 400 neurons in the hidden layer

In addition, Figure 7.22 shows regression plots of the NFS model of the proposed risk estimation technique with respect to targets for training, validation, and testing data. For the NFS model, the fit is good for all training, validation and testing data with the value of R is 0.976, which is close to 1.

7.2.2.2.4 Scaled Conjugate Gradient

Conjugate gradient methods need a line search at each iteration, which is computationally expensive as it requires that the network respond to all training inputs and estimate multiple times for each search. Scaled Conjugate Gradient (SCG) learning algorithm was developed by Moller in 1993 (Moller, 1993). It primarily built to overcome the time-consuming line search associated with conjugate gradient learning methods. The SCG algorithm utilizes the second order information from the ANN to reach faster convergence. It is also fully automated so there are no user-dependent parameters and it avoids time-consuming line-search in each iteration to determine appropriate step size (Moller, 1993).

The SCG learning algorithm with one hidden layer was utilized to train the NFS model of the proposed risk estimation technique. Several experiments were carried out to determine the appropriate number of neurons that produces the lowest error and the best fit with the learning

process. The NFS model was trained with increasing the number of neurons in the hidden layer from 50 to 1200 with observing MSE and RMSE values, as shown in Figure 7.23 and Figure 7.24. The three lines representing training, validation and testing data are almost identical and have the same behaviour. So, there is no overfitting. The results of training the NFS model using the SCG learning method showed unstable behaviour when increasing the number of neurons in the hidden layer. The MSE value of the training data increased from 19.99 at 50 neurons to 23.77 at 100 neurons. Then, the unstable behaviour continued until the MSE reached its lowest value at 1000 neurons. The MSE values were 14.75, 15.17 and 15.01 for training, validation, and testing data respectively. This was the same scenario for the RMSE for training, validation, and testing data where it produced the lowest values at 1000 neurons with 3.84 for training, 3.89 for validation, and 3.87 for testing data.

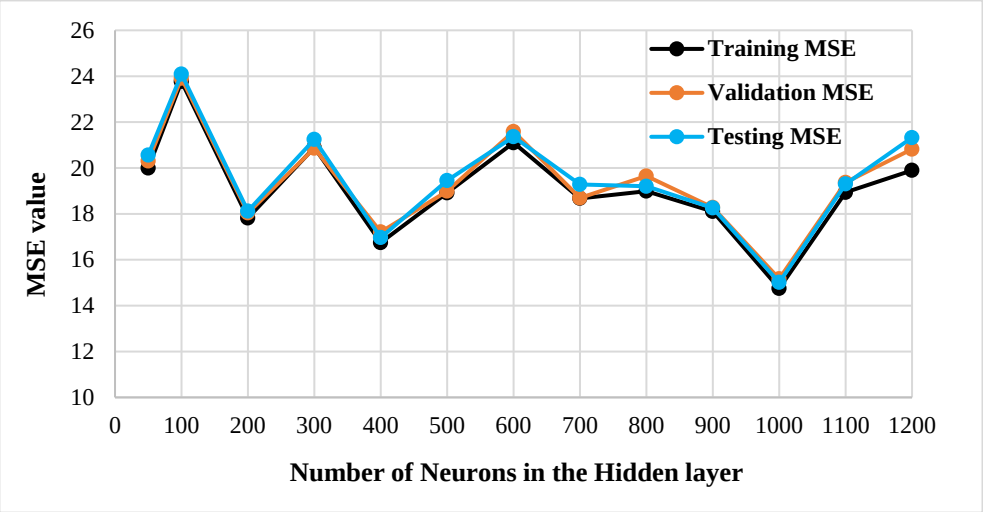


Figure 7.23: MSE of training, validation, and testing data of the NFS model when increasing the number of neurons in the hidden layer with the SCG learning algorithm

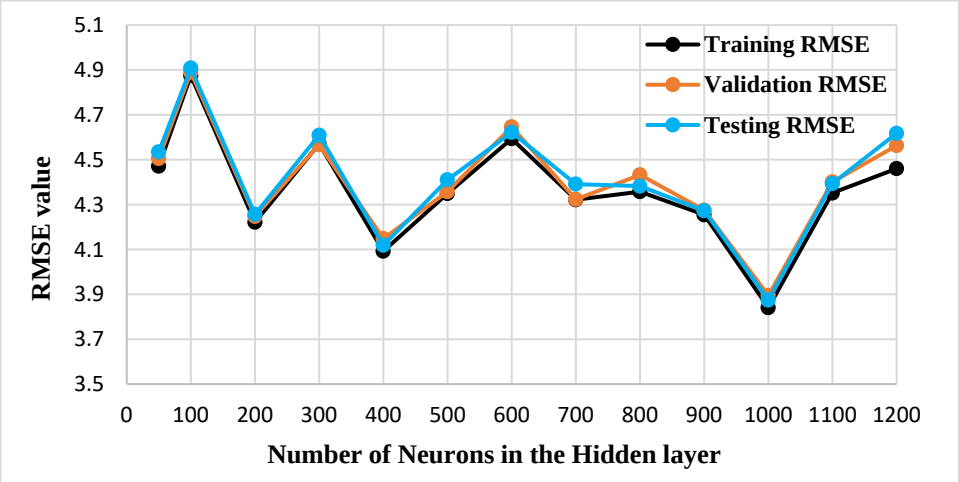


Figure 7.24: RMSE of training, validation, and testing data of the NFS model when increasing the number of neurons in the hidden layer with the SCG learning algorithm

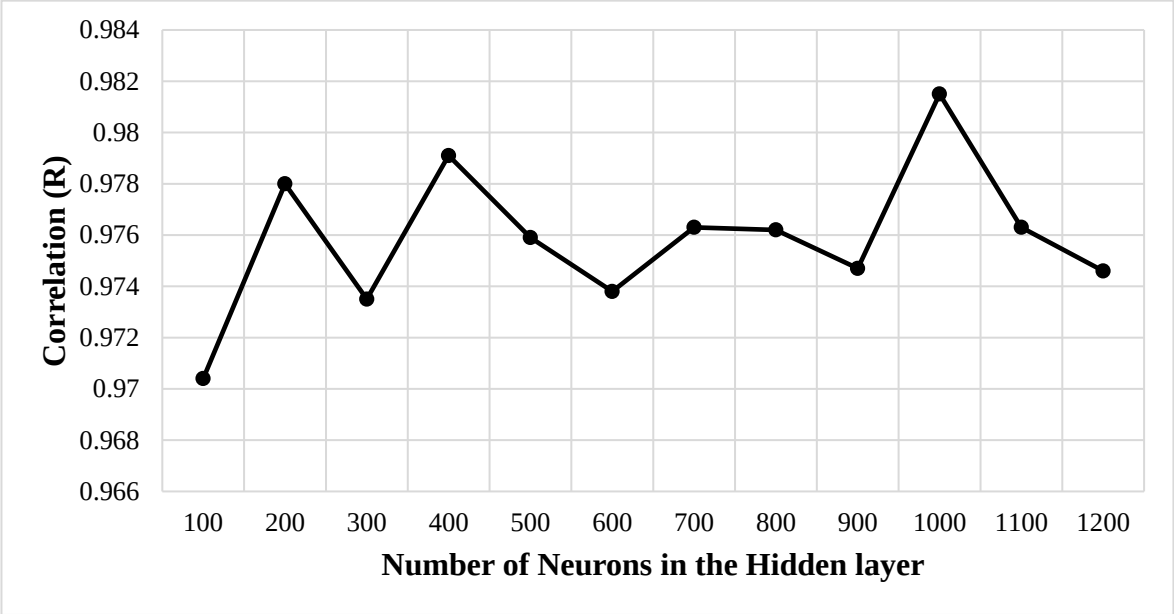


Figure 7.25: Value of R when increasing the number of neurons in the hidden layer with the SCG learning algorithm

In addition, Figure 7.25 shows the value of R when increasing the number of neurons in the hidden layer from 50 to 1200 neurons. The value of R decreased from 0.975 at 50 neurons to reach 0.970 at 200 neurons. The highest correlation for the NFS model using the SCG learning algorithm was realized when applying 1000 neurons in the hidden layer. The value of R was 0.982, which indicates that the NFS model is well trained.

With the SCG training algorithm, the results demonstrated that increasing the number of neurons in the hidden layer led to unstable behaviour of training, validation and testing error values. It showed that the appropriate number of neurons in the hidden layer to implement the NFS model of the proposed risk estimation technique is 1000 where it produced the lowest MSE and RMSE error for training, validation, and testing data. It also produced the highest value of R, 0.982, which indicates that the NFS model is well trained.

The NFS model of the proposed risk estimation technique was implemented using the SCG training algorithm with 1000 neurons in the hidden layer, as shown in Figure 7.26. After the NFS model was trained, the performance graph showed MSE values for training, validation, and testing data, as shown in Figure 7.27. The NFS model is a good fit as the value of R is close to 1. In addition, no overfitting has occurred as training, validation, and testing data have the same behaviour. However, the lowest error produced with the SCG learning algorithm is quite high.

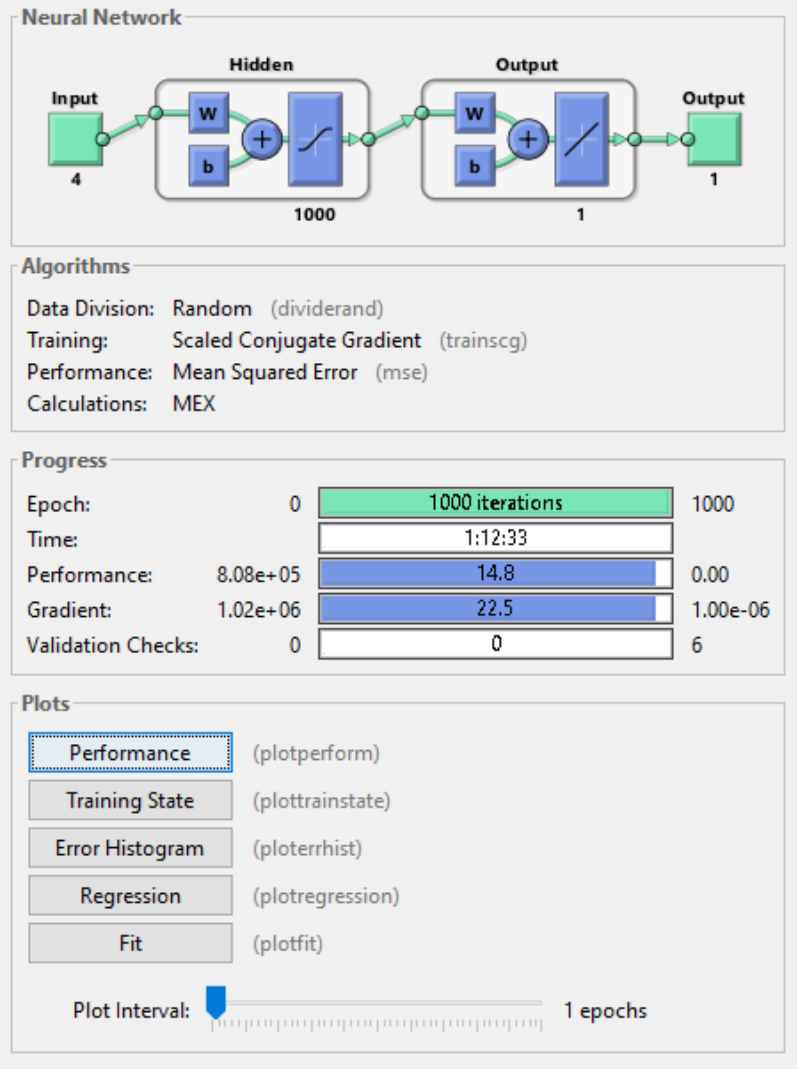


Figure 7.26: Training the NFS model with 1000 neurons in the hidden layer using the SCG learning algorithm

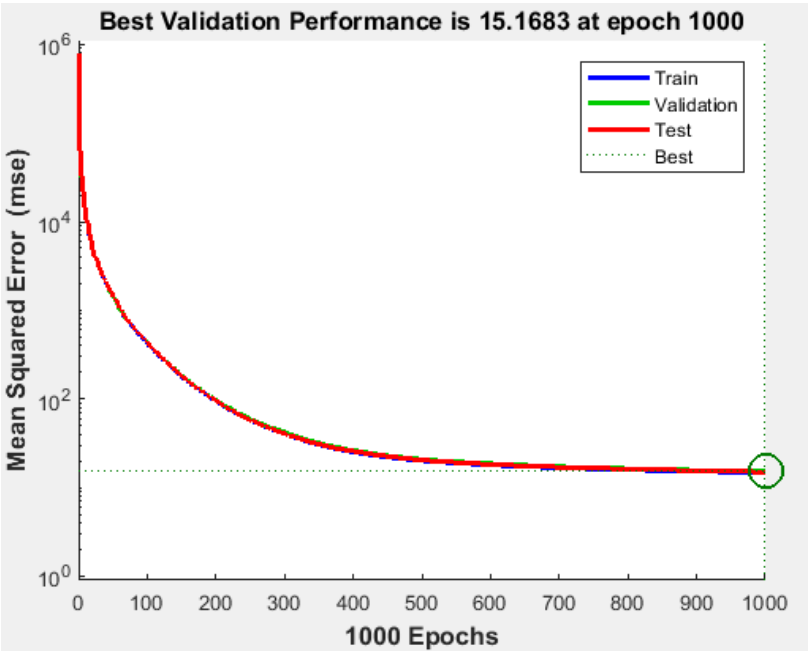


Figure 7.27: Performance of training, validation, and testing data at different number of epochs using the SCG learning algorithm with 1000 neurons in the hidden layer

In addition, the regression plots of the NFS model of the proposed risk estimation technique with respect to targets for training, validation, and testing data show that the fit of the NFS model is reasonably good for all training, validation and testing data with the value of R is close to 1, as depicted in Figure 7.28.

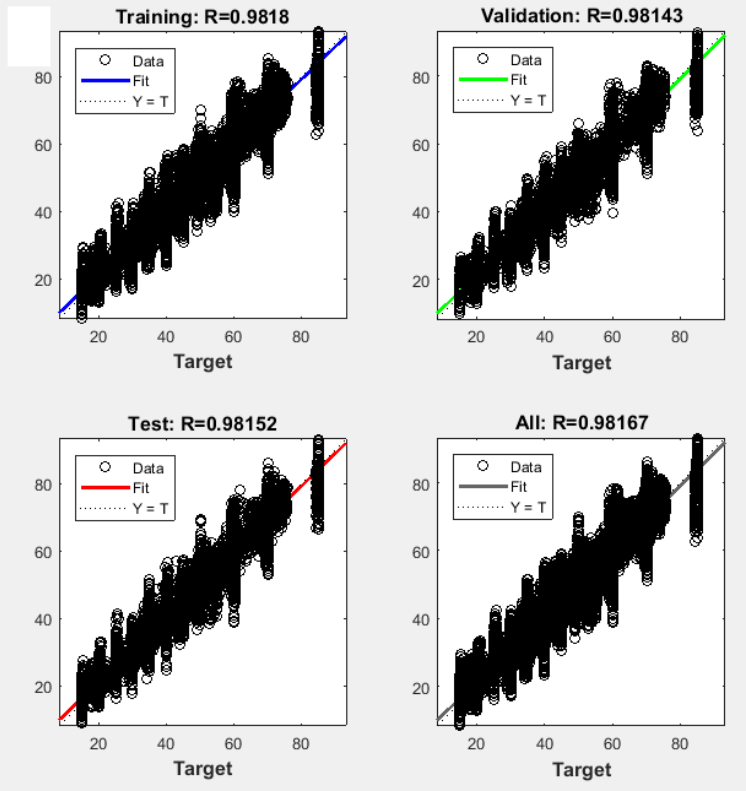


Figure 7.28: Regression plots of training, validation, and testing data when applying the SCG learning algorithm with 1000 neurons in the hidden layer

7.2.2.3 Comparison of Training Algorithms

The NFS model of the proposed risk estimation technique was trained using four different learning algorithms. Firstly, several experiments were carried out to determine the appropriate number of neurons in the hidden layer for each learning algorithm. Then, the learning algorithms were utilized to train the NFS model of the proposed risk estimation technique with the appropriate number of neurons that were previously determined.

A comparison between the four learning algorithms that were utilized to train the NFS model can be seen in Table 7.1. The results demonstrated that there is no optimal number of neurons for the hidden layer that can be used to produce the lowest error and the highest correlation with different learning algorithms. This was proved by having different number of neurons for each learning algorithm. In addition, although the LM and SCG learning algorithms use the same number of neurons in the hidden layer, the LM learning algorithm produced the best results.

Table 7.1: Comparison between learning algorithms used to train the NFS model of the proposed risk estimation technique

Item	Training algorithms			
	LM	**BR**	**CGF**	**SCG**
Appropriate number of neurons in the hidden layer	1000	600	400	1000
Number of epochs	1000	1000	1000	1000
MSE of training	0.978	2.008	19.253	14.750
MSE of validation	1.219	N/A	19.543	15.168
MSE of testing	1.190	2.157	19.136	15.011
RMSE of training	0.989	1.417	4.388	3.841
RMSE of validation	1.104	N/A	4.421	3.895
RMSE of testing	1.091	1.469	4.375	3.874
Correlation Coefficient (R)	0.999	0.997	0.976	0.982

In terms of MSE and RMSE values, the LM learning algorithm produced the lowest error for training, validation, and testing data among other learning algorithms. In addition, the LM learning algorithm produced the highest correlation with 0.999 in the value of R, which indicates that the NFS model is well trained and fit with the learning process. Therefore, the LM learning algorithm was selected as the best learning algorithm to be utilized to implement the NFS model of the proposed risk estimation technique in IoT applications.

7.3 NFS and Fuzzy System

The proposed risk estimation technique was first implemented with the fuzzy logic system using the Mamdani FIS and tested with different number of access requests. One of the challenges that faced adopting the proposed fuzzy risk estimation technique in real-world IoT applications is that it requires large processing time and its scalability seems to be questionable. To overcome this problem, the proposed risk estimation technique was implemented using the NFS with the LM learning method. The parallel computation and learning abilities of the NFS model added more improvements to the risk estimation technique.

The results demonstrated that utilizing the NFS with the LM learning algorithm to implement the proposed risk estimation technique provides less processing time as it uses only one-sixth the time used by the Mamdani FIS, as depicted in Table 7.2. Both methods followed a linear relationship in which increasing the number of access requests led to increasing the processing time.

In addition, the results demonstrated that the time per access request for the NFS model using the LM learning algorithm produced a very short time compared to the time per access request produced by the Mamdani FIS, as depicted in Figure 7.29. The trained NFS model with the LM learning algorithm has proved it provides more efficient processing time which can provide timeliness risk

estimation technique for various IoT applications. Besides adding the learning capability to the risk estimation technique will make it able to adapt to changes of the IoT environment.

Table 7.2: Processing time of the NFS model using the LM learning algorithm and Mamdani FIS

Number of access requests	**NFS using LM Algorithm**		**Mamdani FIS**	
	Time (sec)	Time per request (sec)	Time (sec)	Time per request (sec)
1000	10.8750	0.01088	57.385	0.0574
10,000	81.5469	0.00815	572.125	0.0572
20,000	146.5625	0.00733	1140.4	0.05702
30,000	211.4216	0.00705	1713.6	0.05712
40,000	277.6094	0.00694	2286.4	0.05716
50,000	341.7656	0.00684	2860.5	0.05721
60,000	407.1875	0.00679	3436.2	0.05727
70,000	472.1250	0.00674	4012.4	0.05732
80,000	537.2345	0.00672	4588.8	0.05736
90,000	602.2314	0.00669	5166.9	0.05741
100,000	667.1286	0.00667	5746.23	0.05746
150,000	995.4688	0.00664	8625.32	0.0575
200,000	1325.3124	0.00663	11506.14	0.05753
250,000	1634.8213	0.00654	14390.1	0.05756

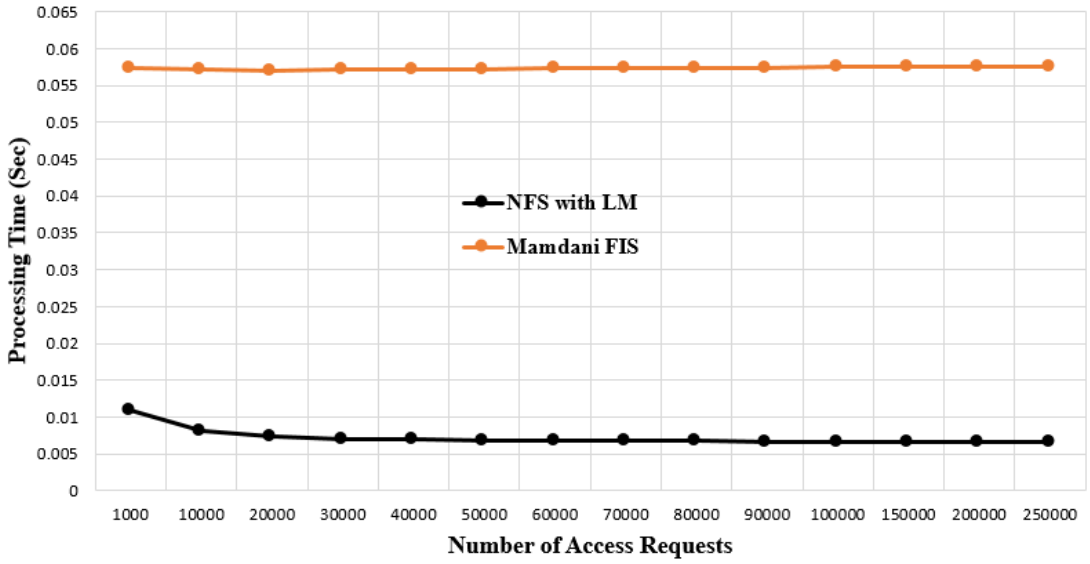


Figure 7.29: Time per access request of the NFS model using the LM learning algorithm and Mamdani FIS

7.4 NFS and ANFIS

The ANFIS model of the proposed risk estimation technique was implemented to tune different MFs and determine the appropriate MF that provides the lowest error and the best fit with the learning process to increase the performance of the risk estimation process. After training the ANFIS model with both hybrid and backpropagation learning techniques at three different epoch numbers (20, 100,

and 300), the results demonstrated that the TrapMF with the hybrid learning approach at 20 epochs is the best MF that produced the lowest error and the best fit with the learning process.

In addition, the NFS model of the proposed risk estimation technique was trained using four learning algorithms to overcome the time overhead associated with the fuzzy logic system and add the learning capability to the risk estimation technique to increase the accuracy and efficiency of the risk estimation process. Several experiments were carried out to determine the appropriate number of neurons in the hidden layer for each training algorithm. After comparing training, testing, and validation errors and correlation of four learning algorithms, the results showed that the LM learning algorithm produced the lowest error and the best fit with the learning process.

The results obtained from implementing the ANFIS and NFS models of the proposed risk estimation technique were compared, as depicted in Table 7.3. The results showed that the performance of the NFS model is better than the ANFIS model. The NFS model provides lower RMSE values in both training and testing data, which indicates how close the relationship between the observed and predicted data. In addition, the NFS model provides higher correlation with 0.999 and 0.997 in R and R^2 respectively, which demonstrates it is the best fit with the learning process as values of R and R^2 are very close to 1.

Table 7.3: Performances of ANFIS and NFS models of the proposed risk estimation technique

Model	**Training RMSE**	**Testing RMSE**	**R**	**R^2**
NFS with LM	0.9888	1.1040	0.9985	0.9974
ANFIS with TrapMF	4.6438	4.6552	0.9731	0.9469

Based on these results, the NFS model with the LM learning method is the best approach to implement the proposed risk estimation technique to increase the accuracy, reduce the processing time needed to provide access decisions and adapt to changes of various real-world IoT applications.

7.5 Summary

Chapter 7 has presented the implementation of the risk estimation process using the NFS. The NFS system which integrates the human reasoning of the fuzzy logic system with the ANN to increase accuracy and performance. The NFS was utilized to reduce the processing time by using the parallel computation of the ANN and add the learning capability to the proposed risk estimation technique to adapt to new changes of the IoT environment. To implement the NFS model, several experiments were carried out using three separate datasets: training dataset (96,000 records), validation dataset (32,000 records), and testing dataset (32,000 records) to train and verify the accuracy of the trained NFS model. To determine the appropriate number of neurons in the hidden layer, the NFS model was trained using four learning algorithms including LM, BR, CGF, and SCG. Several experiments were carried out and MSE, RMSE, and R values of training, testing, and validation were utilized to

determine the appropriate number of neurons and the best learning algorithm for the NFS model. The results demonstrated that the LM learning algorithm produced the lowest error for training, validation, and testing data and the highest correlation with 0.999, which indicates that the NFS model is well trained and fit with the learning process. In addition, the NFS model with the LM learning algorithm has proved it provides efficient processing time, as it uses only one-sixth the time used by the fuzzy logic system. Therefore, the NFS with the LM learning method is the best combination to implement the proposed risk estimation technique to increase the accuracy, reduce the processing time needed to provide access decisions and adapt to new changes of various IoT applications. The next chapter presents the access monitoring and evaluation of the proposed risk-based access control model.

Chapter 8: Access Monitoring and Model Evaluation

This chapter provides a discussion of access monitoring and model evaluation. It starts by providing an overview of access monitoring. Then, section 8.2 discusses utilizing smart contracts to monitor user activities during the access session. This includes simulating the operation of smart contracts using Simulink and presenting various access scenarios. Section 8.3 discusses the evaluation of the proposed risk-based access control model by presenting access control scenarios of three IoT applications including healthcare, smart home and network router. The chapter closes by providing a summary of the main points discussed through the chapter and introduces the next chapter.

8.1 Access Monitoring

The key objective of an access control model is to allow only authorized users to access system resources in an authorized way. Typically, access control models can be divided into two classes: stateless and stateful. Stateless access control is only concerned with the current state of the system to provide access decisions, while stateful access control integrates past and current accesses to determine the access decision. Although most existing access control systems are stateless, building a stateful access control model should be one of the fundamental priorities to guarantee security and privacy of system resources (Gomez & Trabelsi, 2014).

Since existing access control approaches do not provide a way to detect and prevent malicious actions after granting the access, the proposed risk-based model adds abnormality detection capability by utilizing smart contracts to track and monitor user activities during access sessions. Hence, the risk estimation module adjusts user's permission adaptively depending on their behaviour in the access session in which if an abnormal action is observed, user privileges will be reduced, or the access session will be terminated.

The next section provides a detailed discussion of using smart contracts to monitor access sessions in IoT systems.

8.2 Smart Contracts for Monitoring

Smart contracts are so powerful because of their flexibility. They can encrypt and store data securely, restrict access to data only to desired parties and then be programmed to utilize the data within a self-executing logical workflow of operations between parties (Watanabe et al., 2016). Smart contracts translate the business process into a computational process to improve operational efficiency (Watanabe et al., 2016). The key purpose of a smart contract is to execute terms or conditions of the contract automatically when certain conditions are verified or met.

An overview of smart contracts involving definitions of smart contracts, benefits, and how a smart contract works was presented previously in chapter 2 in section 2.6. Therefore, this chapter focuses only on discussing how smart contracts can be utilized to monitor access sessions in IoT systems.

In the proposed risk-based model, the smart contract is used as a mean to track and monitor user behaviour during access sessions. After granting the access, a smart contract will be created for each access request. The access permissions will be implemented as conditions or terms in the smart contract. Then, the monitoring module will compare the user behaviour with the terms and conditions of the contract to detect abnormal actions throughout access sessions. As depicted in Figure 8.1, the requesting user first defines the data or resource to be accessed and action to be performed in the access request. Then, if the access is granted, a smart contract will be created to implement user's permissions as conditions or terms to guarantee that the user has the ability to access only resources and perform actions that were requested. Then, resources and actions will be monitored to detect violations. if a violation is detected, the system will issue a warning, or the session will be terminated. If no violations are detected, the system will keep monitoring the user behaviour throughout the access session.

As discussed earlier, smart contracts are software code that runs using the blockchain technology. Due to the difficulty of implementing smart contracts and interfacing it with the risk estimation process of the proposed risk-based access control model, MATLAB Simulink was utilized to simulate the operation of smart contracts to validate its efficiency and effectiveness to monitor access sessions to detect and prevent malicious actions.

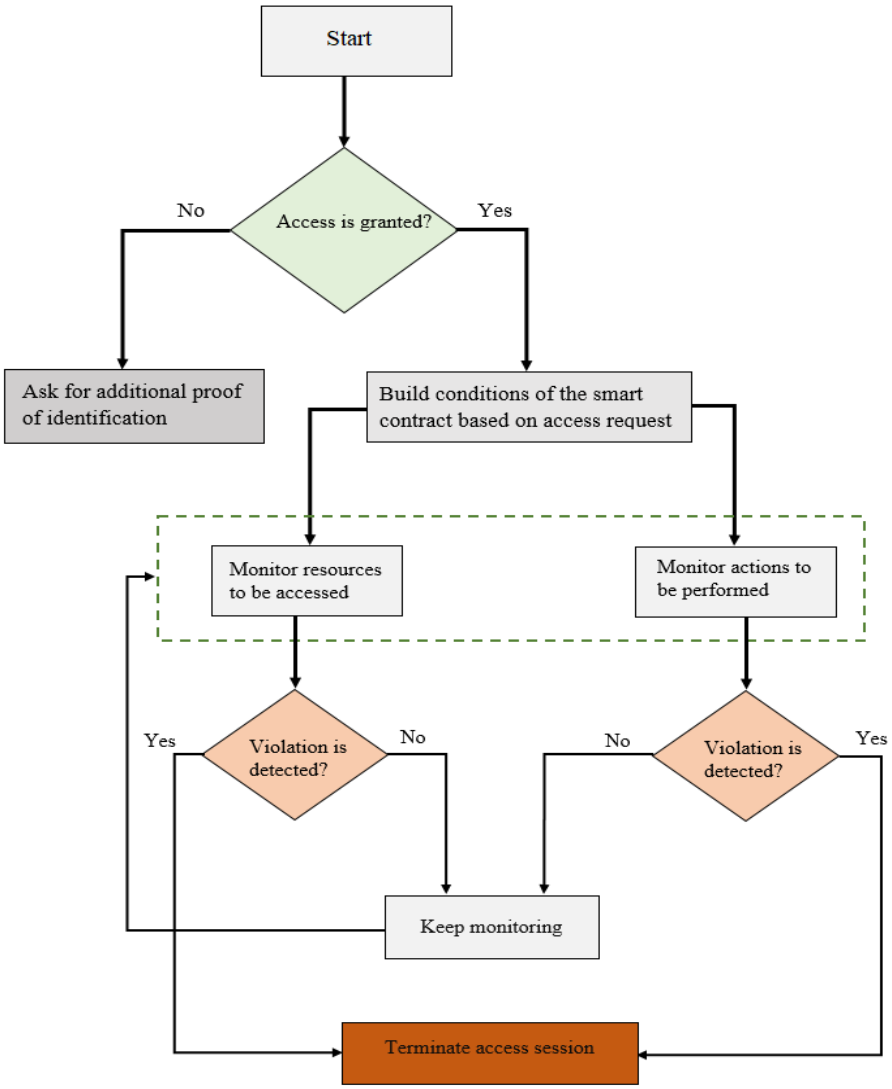


Figure 8.1: Flowchart of monitoring user activities using smart contracts

8.2.1 Simulation of Smart Contracts

Simulink is a graphical environment to model, simulate, and analyse multi-domain dynamic systems. It is primarily based on hierarchical data flow diagrams. A Simulink diagram consists of functional blocks connected by signals (wires). These blocks represent transformations of data, while the signals represent the flow of data between blocks. Each block contains input and output ports to connect with other blocks and transfer signals between blocks. The input ports provide data to the block, while the output ports provide the results computed by the blocks (Aung, 2007; Boström et al., 2010).

Stateflow is one of the main elements of the Simulink environment that was utilized to simulate the operation of smart contracts. Stateflow is an environment for modelling and simulating sequential decision logic based on state machines and flow charts. It combines graphical and tabular representations, including state transition diagrams, flow charts, state transition tables, and truth tables to model how the system reacts to events, time-based conditions, and external input signals (Mathworks, 2016).

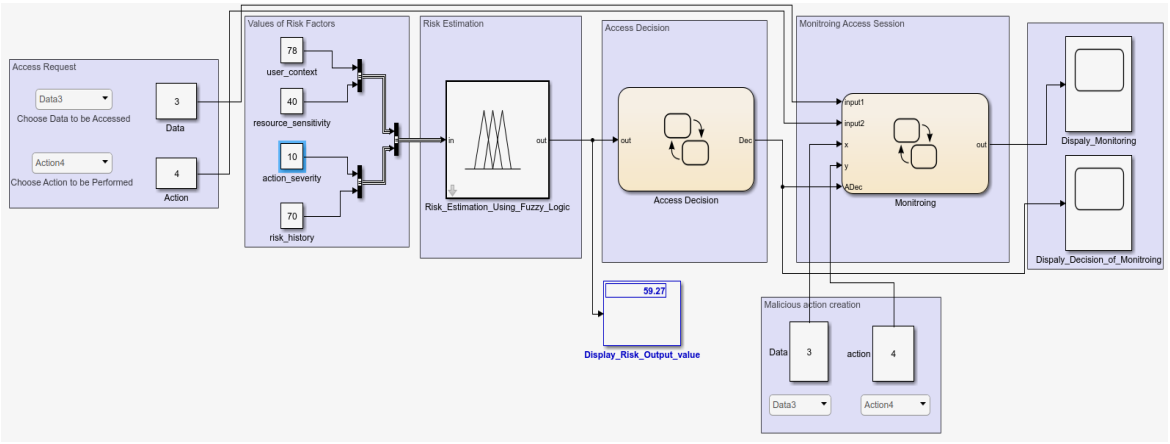


Figure 8.2: Simulation of the proposed risk-based access control model with monitoring user activities

Simulink was utilized to build a simulation model for the proposed risk-based access control model, as shown in Figure 8.2. The first block from the left represents the access request. The requesting user has to specify the data/resource to be accessed and the action to be performed on the system. In the simulation model, five different data and actions were assumed, and the requester has to choose one resource and one action for each access request. The second block represents input risk factors of the proposed risk-based model, which include user context, resource sensitivity, action severity, and risk history. Estimating the output risk value for each access request is based on these values.

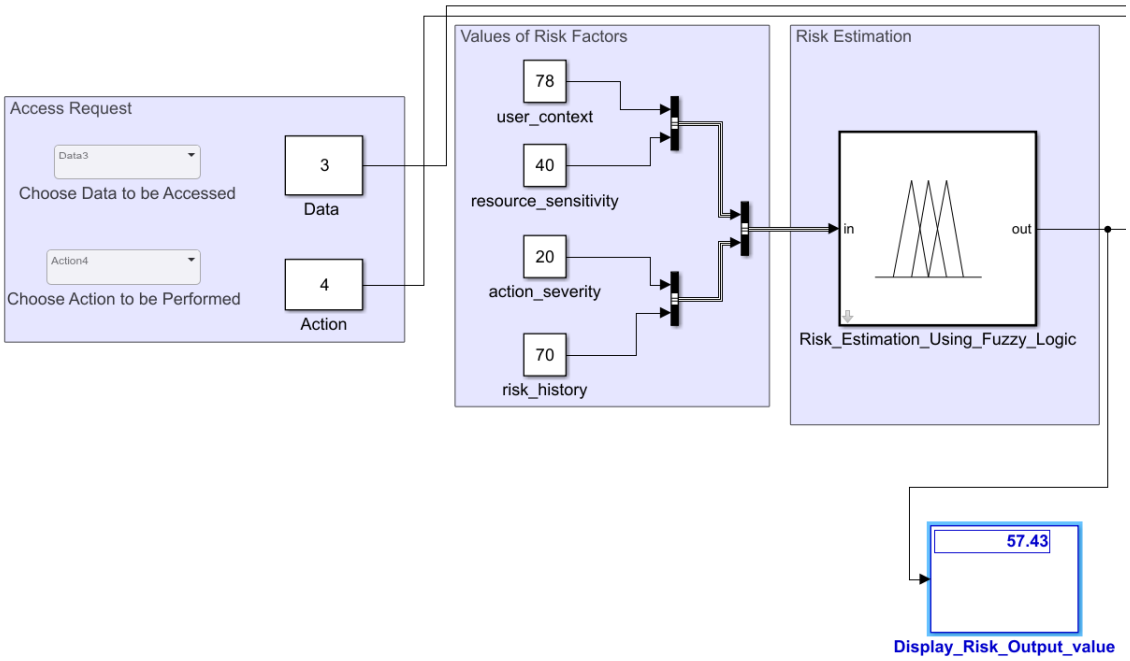


Figure 8.3: First part of the simulation of the proposed risk-based access control model

The third block represents the risk estimation technique using the fuzzy logic system with expert judgment. As discussed earlier in Chapter 5, 6 and 7, the risk estimation process was implemented using the fuzzy logic system with expert judgment, NFS and ANFIS. However, the fuzzy logic system with expert judgment was utilized in this simulation since the main target is to monitor the access session not to estimate the risk. Also, it is easier to implement with the Simulink. The main

objective of the third block is to estimate the risk value associated with the access request and display it on the display screen, which located under the third block. The first three blocks of the simulation model of the proposed risk-based model can be seen clearer in Figure 8.3.

The fourth block represents the access decision, as depicted in Figure 8.4. After estimating the risk value, the access decision should be determined. As discussed previously in section 5.4, three access decision bands were proposed for each access request including Grant, Grant with monitoring and Deny. The access decision and monitoring status depend on the output risk value as assumed in Table 8.1.

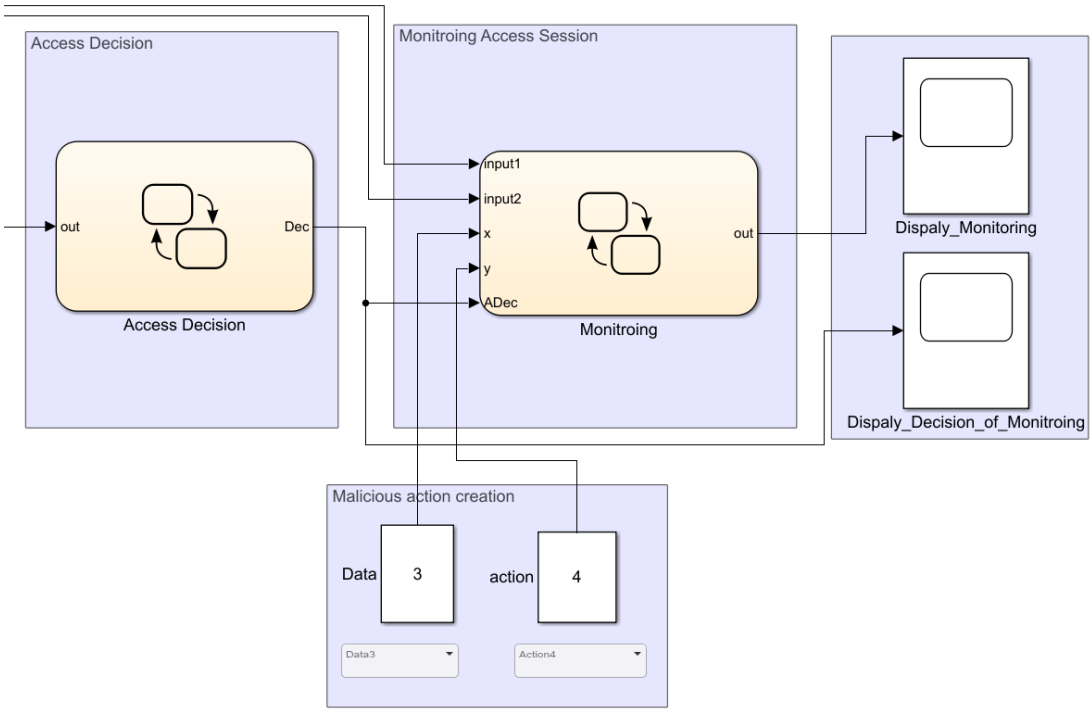


Figure 8.4: Second part of the simulation of the proposed risk-based access control model

Table 8.1: Output mapping of access decision and monitoring blocks

Access Decision Block			**Monitoring Block**		
Estimated Risk Value	**Output**	**Monitoring Status**	**State**	**Description**	**Monitoring Output**
1 < Risk ≤ 15	Grant without monitoring	0	Normal	Monitoring in progress	0
15 < Risk < 70	Grant with monitoring	1	Detect	Malicious action detected	1
Risk ≥70	Deny	0	No_monitroing	No monitoring needed	2

To implement these three access decision bands in the simulation, four states including *Decision*, *Granted_without_monitroing*, *Granted_with_monitroing*, and *Denied* were utilized, as shown in Figure 8.5. The transition from states depends on the estimated risk value. The *Decision* state takes the estimated risk value from the risk estimation technique (third block). Then, this value is compared against risk decision bands in which if the estimated risk value is higher than 1 and less than or equal

15, the state will be changed, and the control will be moved to *Granted_without_monitroing* state, and the monitoring status will be 0 to indicate that no monitoring is needed. During this state, the system will check the estimated risk value every second to reflect any changes in input risk factors on the output risk value. So, the decision module will be updated with the output risk value every second. While if the estimated risk value is higher than 15 and less than 70, the state will be changed, and the control will be moved to *Granted_with_monitroing* state, and the monitoring status will be 1 to indicate the need for monitoring user activities. During this state, the system will also check the estimated risk value every second to keep updated with the output risk value during the access session. If the estimated risk value is higher than or equal to 70, the state will be changed, and the control will be moved to *Denied* state and the monitoring status will be 0 to indicate that no monitoring is needed as the access request was rejected.

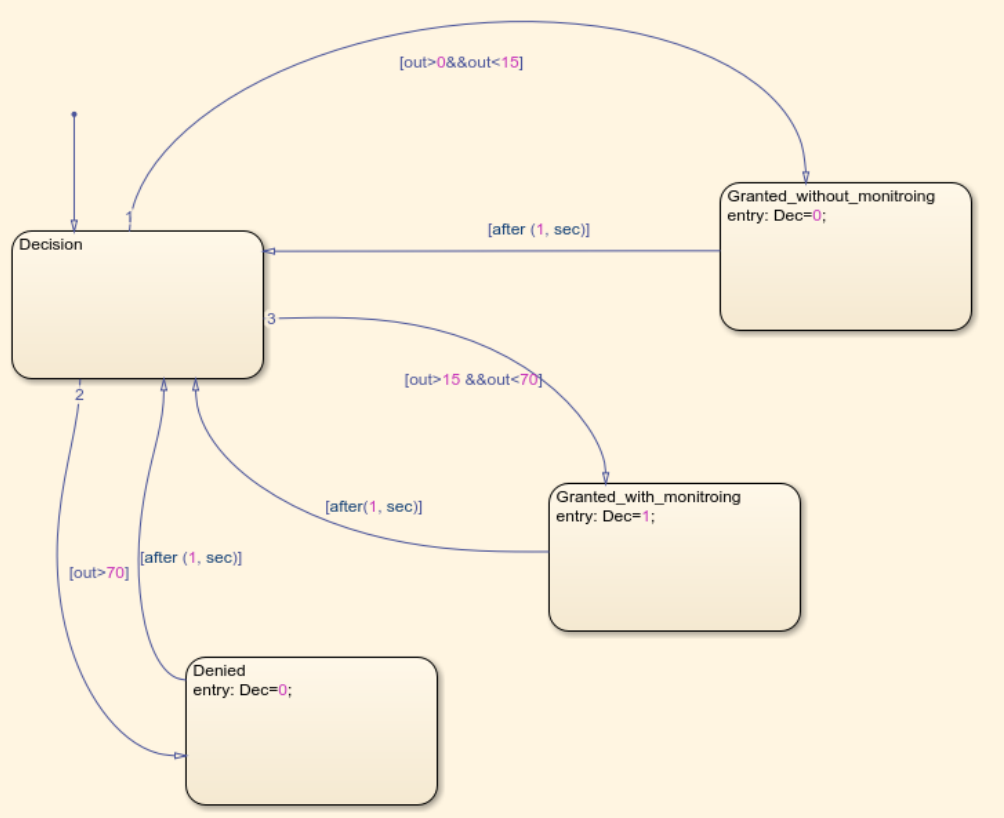


Figure 8.5: Stateflow charts to determine the access decision based on the estimated risk value

The fifth block represents the simulation of monitoring access session using smart contracts, as shown in Figure 8.4. As discussed earlier, smart contracts are software code that implements terms or conditions of the contract and executes it automatically when conditions are met. The key objective of simulating the operation of smart contracts is to ensure that the system is aware of actions and changes occurred during the access session. In addition, the response of the monitoring module should be fast enough to detect and prevent malicious actions in a very short time to stop the attacker from affecting system resources improperly.

The operation of smart contracts was simulated using four states including *Decision*, *No_monitroing*, *Normal* and *Detect*, as depicted in Figure 8.6. The *Decision* state takes the decision value from the access decision module. If the decision value was 0, this means that the access is either granted without monitoring or denied. Hence, the control will be moved to *No_ monitoring* state, and the monitoring output will be 2 to indicate that no monitoring is needed, as summarized in Table 8.1. While if the decision value was 1, the control will be moved to *Normal* state to indicate that the system is monitoring the user activities and no malicious action was detected. The *Normal* state represents the operation of the smart contract in which the user permissions will be coded as conditions or terms. For example, if the user decided in his/her access request to access *Data1*, and perform *Action3*, then the conditions or terms that allow the user to only access *Data1* and perform *Action3* will be implemented in the contract. Therefore, if the user tried to access other data or perform another action, this will be detected as a malicious action.

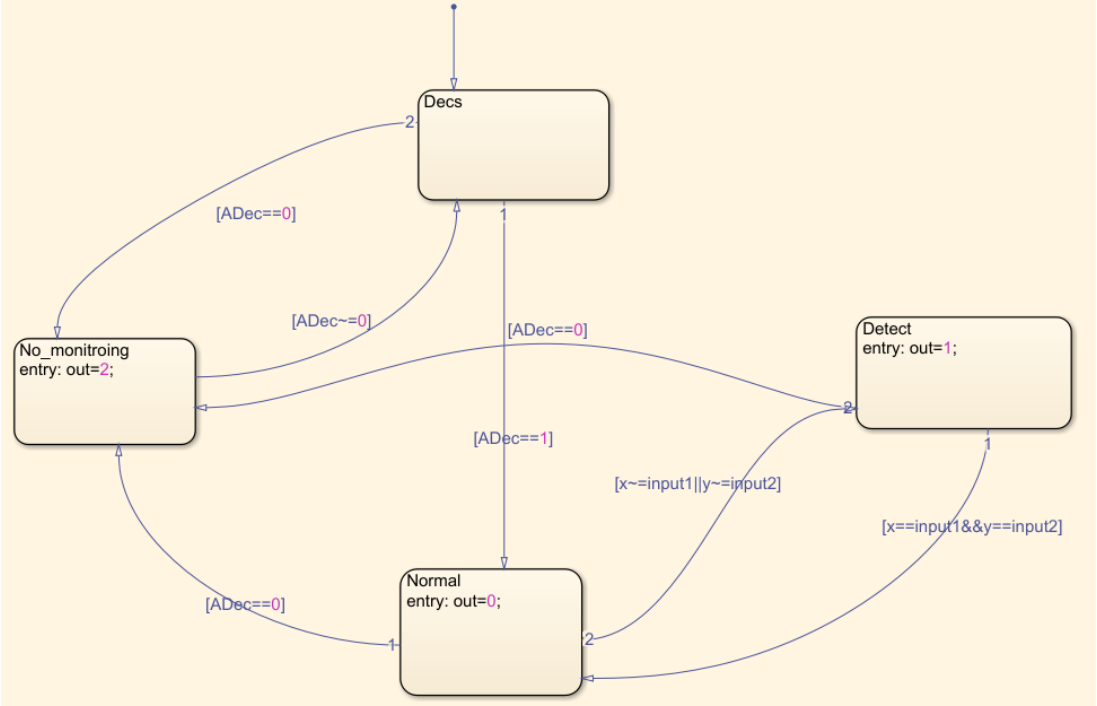


Figure 8.6: Stateflow charts of simulating the operation of smart contracts.

To simulate this scenario, the monitoring block was linked to another block that has two input variables x and y, where x refers to the data to be accessed and y refers to the action to be performed, as shown in Figure 8.4. These variables are identical and have the same values of input variables of the access request of the first block. These variables will be used to simulate creating malicious actions. In other words, after defining the type of data to be accessed and type of action to be performed in the access request and after granting the access, a smart contract will be created to ensure the user has the eligibility only to access data and perform the action specified in the access request. The variables x and y will be used to reflect the behaviour of the user during the access session. Hence, if values of x and y are the same as input variables of the access request that were implemented in the contract as conditions, then no malicious actions will be detected. While if values

of x and y are different from input variables implemented in the contract, then this will be classified as a malicious action that needs to be prevented.

If the user tried to perform a malicious action during the access session, then the control will be moved to *Detect* state, and the output value will be 1 to indicate that a malicious action was detected. Then, a warning message will be issued to guide the user to access only permitted data and perform permitted actions, if the user stops the malicious action, then the system will return to *Normal* state. Otherwise, the system will terminate the access session to stop malicious actions. Figure 8.7 summarizes the process flow of the monitoring module.

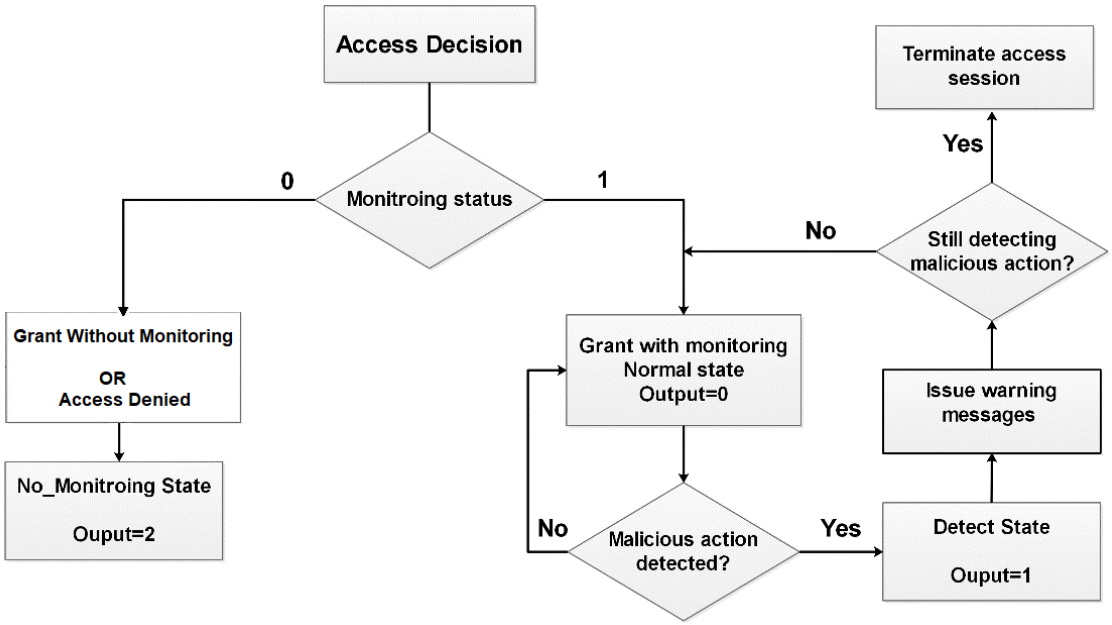
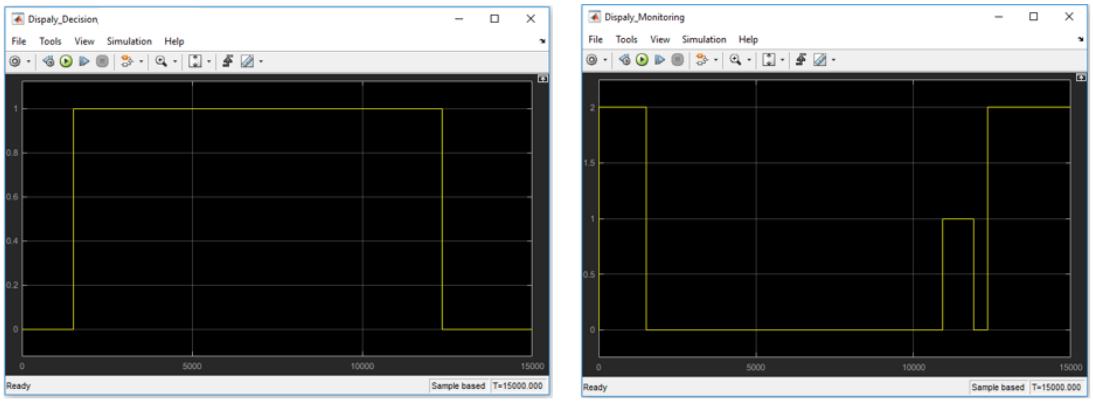


Figure 8.7: Process flow of the monitoring module

The last block of the simulation model of the proposed risk-based model represents two displays to show access decision and monitoring status, as shown in Figure 8.8. The first display shows two values of access decision in which 0 represents that the access is denied, whereas 1 represents that the access is granted, as shown in Figure 8.8 (a). The second display shows the monitoring status in which 2 represents no monitoring is needed, 0 represents that monitoring user activities is in progress, whereas 1 represents that a malicious action was detected, as shown in Figure 8.8 (b).



(a) This display shows the access decision (b) This display shows the monitoring status

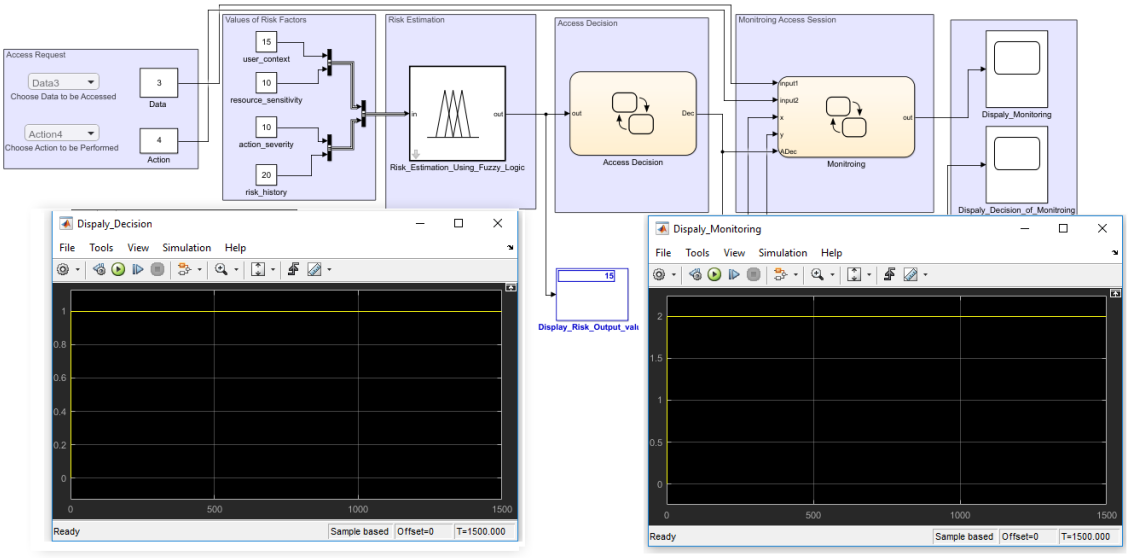
Figure 8.8: Two displays to show access decision and monitoring status

8.2.2 Access Scenarios

To show the effectiveness of monitoring user activities using smart contracts during the access session and the response of the proposed risk-based model in different situations, this section presents various access scenarios that can occur. In addition, the access decision and monitoring status will be discussed for each scenario.

8.2.2.1 Scenario 1: Access was granted without monitoring

The proposed risk-based access control model protects the user's privacy by allowing the user to access the system without monitoring user activities during the access session if the estimated risk value associated with the access request was very small. This access decision band is very narrow to reflect only IoT devices' owners. Therefore, if the estimated risk value associated with the access request was less than or equal 15, the access will be granted, and no monitoring will be needed.



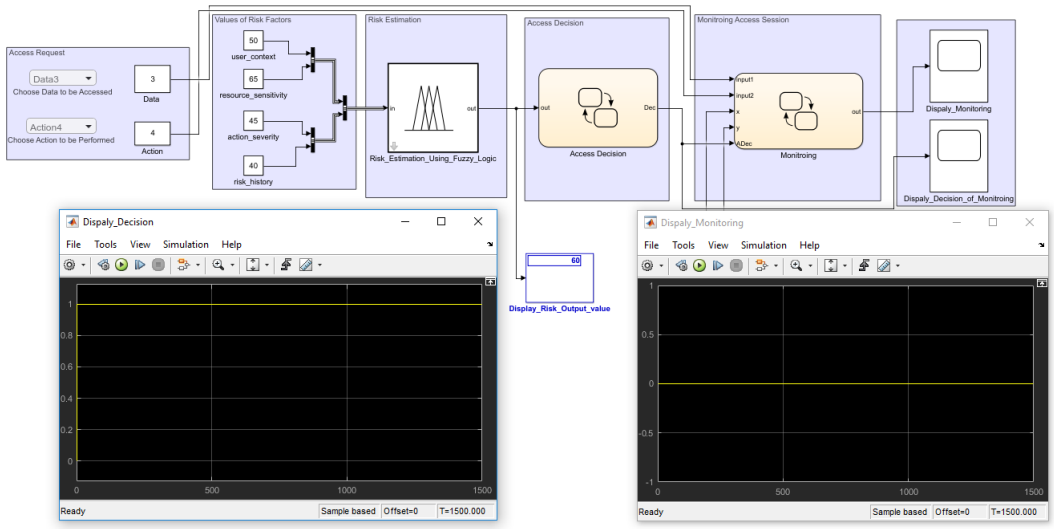
(a) Access was granted (b) No monitoring

Figure 8.9: Access control scenario when the access was granted without monitoring

As depicted in Figure 8.9, the access was granted without monitoring as the estimated risk value associated with the access request was 15. This indicated by having 1 on the access decision display (the first display from left) to indicate that the access was granted, as shown in Figure 8.9 (a). Also, having 2 on the monitoring status display to indicate that no monitoring is needed, as shown in Figure 8.9 (b).

8.2.2.2 Scenario 2: Access was granted, and monitoring is in progress

If the estimated risk value associated with the access request was higher than 15 and less than 70, the access will be granted with monitoring user activities during the access session. As shown in Figure 8.10, the estimated risk value was 60, which implies that the access was granted with monitoring. This indicated by having 1 on the access decision display to indicate that the access was granted, as shown in Figure 8.10 (a). Also, having 0 on the monitoring status display to indicate that monitoring user activities is in progress, as shown in Figure 8.10 (b).

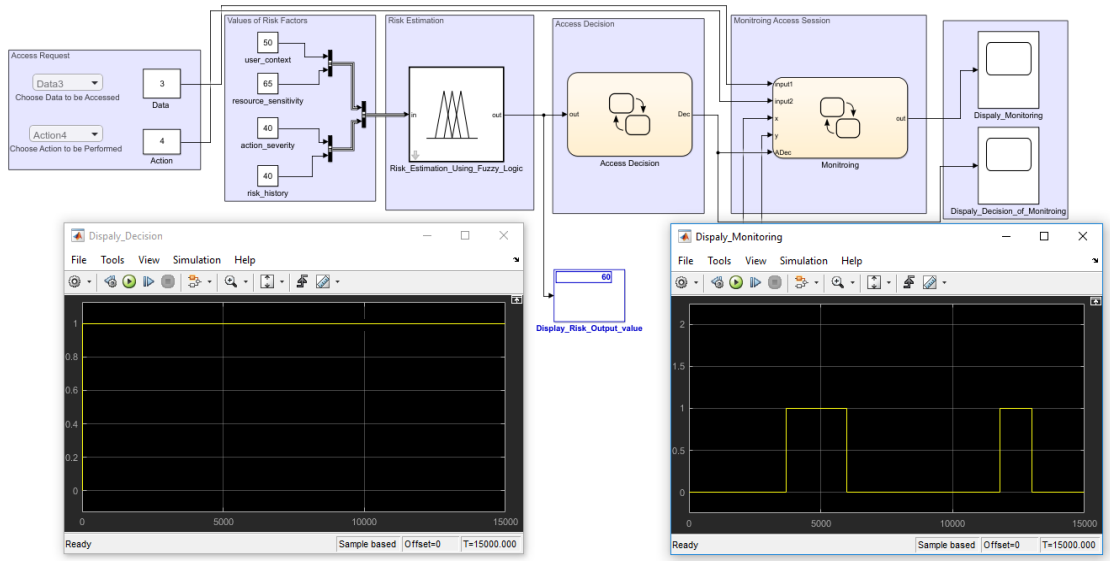


(a) Access was granted (b) Monitoring is in progress

Figure 8.10: Access control scenario when the access was granted and monitoring is in progress

8.2.2.3 Scenario 3: Access was granted, and a violation was detected

This scenario is similar to the previous scenario in which if the estimated risk value associated with the access request was higher than 15 and less than 70, then the access will be granted with monitoring user activities during the access session. Since the estimated risk value was 60, the access was granted with monitoring. This indicated by having 1 and 0 in the access decision display and the monitoring display respectively, as depicted in Figure 8.11.



(a) Access was granted (b) A violation was detected

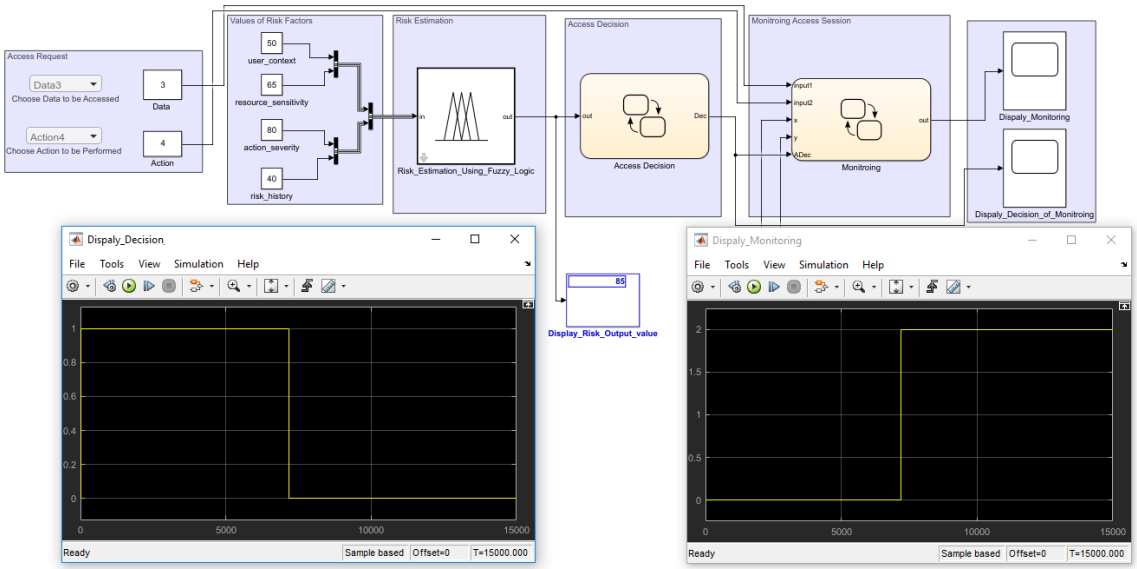
Figure 8.11: Access scenario when the access was granted with monitoring and a violation was detected

After granting the access, the monitoring module will track the user behaviour during the access session by comparing values of x and y with input values of the access request. When a violation is detected, the output of the monitoring display will change from 0 to 1 to indicate that there is a violation. Then, a warning message will be issued. If the user obeys the advice of the warning message, the monitoring module will return to tracking user behaviour and the output of the monitoring display will return to 0, as shown in Figure 8.11 (b). In the same way, if another violation is detected, the system will issue a warning message. Warning messages were not implemented in the simulation model, but it will be presented in the next section of simulating the proposed risk-based model on the web.

One of the most important features of a monitoring technique is the ability to detect and prevent malicious actions in a very short time. As depicted in Figure 8.11 (b), the response of the monitoring module was very good in which when one of the contract conditions was not verified, the monitoring module detects it immediately. This shows that the use of smart contracts and implementing user permissions as software code can detect and prevent malicious actions in a very short time.

8.2.2.4 Scenario 4: Make another access request

If the estimated risk value associated with the access request was higher than 15 and less than 70, the access will be granted with monitoring user activities during the access session. This indicated by having 1 and 0 in the access decision display and the monitoring display respectively, as shown in Figure 8.12.



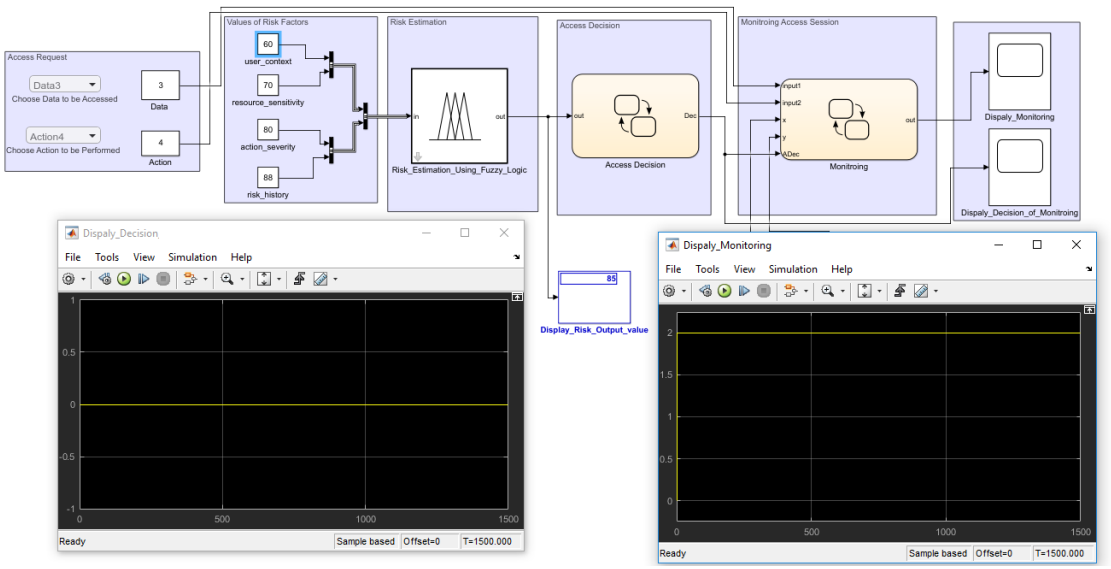
(a) Access was granted then denied for second request (b) Access was monitored when it granted only

Figure 8.12: The access control scenario when the access is granted with monitoring, and the second access request was denied

During the access session, if the user wants to access other data or perform another action rather than those implemented in the smart contract, the user has to make another access request. The access request can be accepted or rejected based on values of risk factors associated with the access request. If the access is granted, another contract will be created with new conditions that allow the user to access the new requested data and action and the monitoring module will continue tracking the user behaviour during the access session. While if the action is rejected, then the output of the decision display will be 0, as shown in Figure 8.12 (a). Also, the output of the monitoring module will be 2 to indicate that no monitoring is needed, as shown in Figure 8.12 (b). This scenario will be presented clearly in the next section while simulating the proposed risk-based model on the web.

8.2.2.5 Scenario 5: Access denied

If the estimated risk value associated with the access request is higher than 70, the access will be denied. As depicted in Figure 8.13, the estimated risk value was 85, so the access was denied. This indicated by having 0 on the access decision display to indicate the access was denied. Also, having 2 on the monitoring status display to indicate that no monitoring is needed as the access was denied.



(a) Access was denied (b) No monitoring

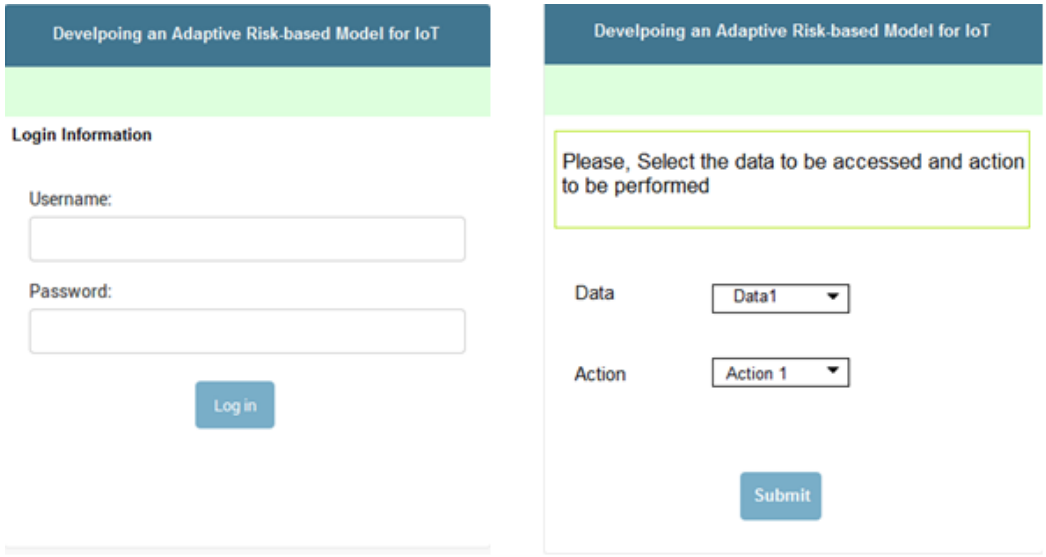
Figure 8.13: Access control scenario when access was denied

8.2.3 Simulation on the Web

The use of smart contracts to detect and prevent malicious actions during the access session adds another dimension to this research by integrating the existing centralized IoT system with the decentralized blockchain technology. However, the major issue in this research was how to prove the effectiveness of using smart contracts in monitoring user activities. To solve this issue, Simulink Stateflow charts were utilized to simulate the operation of smart contracts. The results of various access scenarios discussed earlier demonstrated that smart contracts can provide an effective solution to detect and prevent malicious actions in a timely manner.

To show how the proposed risk-based access control model works on the web, a simple web application was created to show various stages of the access control process. This section provides a detailed discussion starting from sending the access request and getting the response from the system. It also validates the operation of smart contracts in monitoring user activities during the access session.

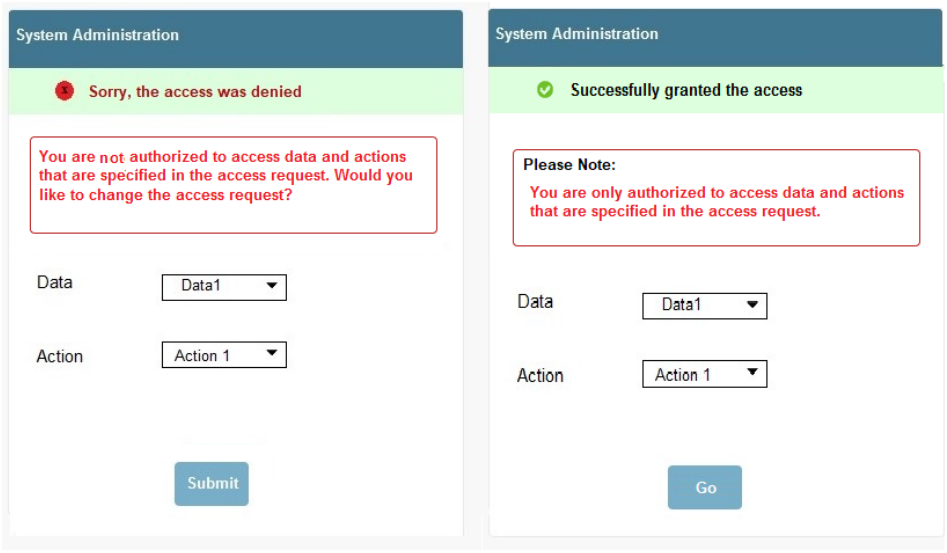
The journey starts when a certain user wants to access one of the system resources. The first stage is to verify the user identity through the common authentication method by using username and password, as depicted in Figure 8.14 (a). If the requesting user is successfully authenticated, then the system will ask the user to create an access request by specifying the data to be accessed and the action to be performed using the provided dropdown lists, as shown in Figure 8.14 (b). In this application, five types of data or resources and five actions can be selected.



(a) Verify the identity of the user (b) Specifying data and action for the access request

Figure 8.14: Login information and creating an access request

After submitting the access request, the system collects contextual information associated with the access request, sensitivity metric of the data to be accessed, severity metric of the action to be performed, and previous risk values of the user to estimate the risk value associated with the access request and make the access decision.



(a) Access was denied (b) Access was granted

Figure 8.15: Access decision based on the estimated risk value

If the access is denied, the system will notify the user to either make another access request or terminate the access session, as shown in Figure 8.15 (a). While if the access is granted, the system will notify the user that he/she is only allowed to access data and actions specified in the access request, as shown in Figure 8.15 (b). If the estimated risk value is less than or equal 15, no monitoring is needed. While if the estimated risk value is higher than 15 and less than 70, then the user activities will be monitored.

If the access is granted with monitoring user activities, then a smart contract will be created to implement terms and conditions that allow the user to access only data and actions specified in the access request. Then, the system will track user activities during the access session to make sure that the user obeys terms and conditions of the contract. If a violation is detected, the system will issue a warning message and terminate the access session, as shown in Figure 8.16.

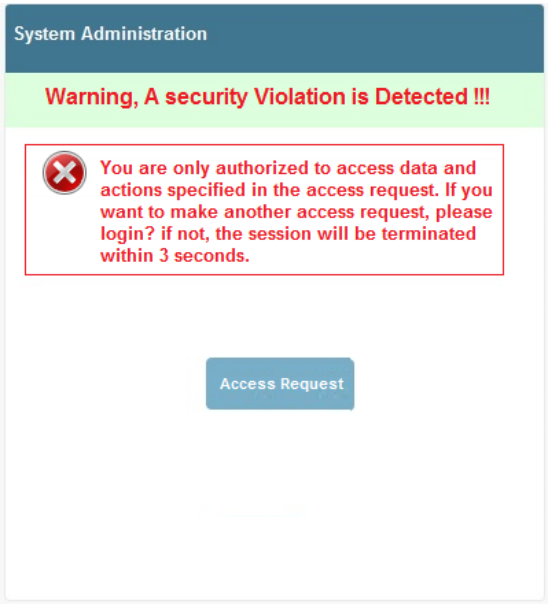


Figure 8.16: The system response when a violation was detected

8.3 Evaluation of Proposed Model

Evaluation is an essential phase to ensure the effectiveness of a research idea on real-world applications. One of the important aspects of the proposed risk-based access control model is to check its applicability in real-world IoT applications. Therefore, access control scenarios of three IoT applications including healthcare, smart home and network router were presented to show the effectiveness of implementing the proposed risk-based access control model on these applications. In addition, the proposed risk-based model was compared with existing risk-based access control models utilizing the fuzzy logic system for the risk estimation process to show major improvements in the proposed risk-based model.

In addition, one of the major issues associated with IoT devices is limited processing and storage capabilities. For the proposed risk-based model, the processing and storage capabilities of existing IoT devices cannot handle the requirements of the risk estimation technique and access monitoring using smart contracts. To solve this issue, there are three ways to implement access control models in the IoT including centralized, distributed and centralized and contextual approach, as discussed in section 2.3.1. Therefore, the centralized and contextual approach was adopted in this research to implement the proposed risk-based access control model where there is a central server connected to

IoT devices with the required processing and storage capabilities which allow IoT devices to participate in making the access decision.

The next section provides a detailed discussion of access control scenarios of three IoT applications to show access decisions based on risk values in various situations.

8.3.1 Access Scenario 1: Healthcare

Access control is a key element in healthcare information systems. Its main objective is to enforce access rules to guarantee that only authorized users can access system resources. Protecting patients' data is not the only concern in healthcare systems but providing access in unexpected situations. In crisis or emergency situations, the availability of information takes precedence over privacy and security concerns. Therefore, providing a dynamic access control model for healthcare is a significant aspect to ensure data security and adapt to unexpected situations.

This section presents applying the proposed risk-based access control model that uses contextual and real-time information to provide access decisions in a children hospital. Different access control scenarios will be presented to show the access decision with various input states.

8.3.1.1 Scenario Description

A closed world scenario involving a healthcare provider such as Mount Cedar (MC) children hospital (Ardagna et al., 2010) was utilized to show various access control scenarios. Typically, patients' information in hospitals is stored as datasets. Each dataset is characterized by a unique object identifier. Datasets can be organized in classes that can be collectively referred with a given name and associated with an object profile (metadata) that provides additional information about the dataset.

Consider the MC hospital has now received a four-years-old child called Harry, who was brought into the MC's first aid clinic by his mother, Eva, late Wednesday evening. The admitting staff observed that Harry suffered from several bruises all over his body, a fractured rib, and a distorted shoulder (Ardagna et al., 2010). Let us walk through the events that would occur in this situation. Initially, Harry's doctor in the first aid clinic, Dr Chris made an access request to the system to view or read Harry's history fills in the Electronic Patient Record (EPR). He also assigned Harry to a care team involving a set of nurses and ordered a series of examinations. The leader nurse of the care team made an access request to the system to read Harry's fills in the patient's EPR.

When the examination results are returned, Dr Chris wrote the diagnosis and the required medication for Harry and called social workers and policemen to investigate the incident as he suspected that child abuse occurred. Therefore, one of the social workers who are responsible for helping the

children in case of abuses and a police officer requested to access Harry's medical information for investigation purposes.

8.3.1.2 Scenario Actors

The MC is a children hospital. Actors involved in this scenario include:

- The child who needs treatment;
- Doctors who are responsible for providing care to the child;
- Nurses who are responsible for helping the doctors;
- Social workers who are responsible for helping the children in case of trauma or abuses;
- Policemen who are responsible for investigating and establishing possible criminal charges and responsibilities in cases of trauma or abuses.

8.3.1.3 Scenario Assumption

Applying the proposed risk-based access control model on the healthcare access control scenario requires defining values of the four risk factors for each access request. For the action severity, three actions were assumed involving; read/view, write and delete. The delete operation is not permitted for all actors involved in this scenario as the hospital keeps track of all medical history of patients, so no need to delete any data. As discussed earlier, there are various actors involved in this scenario in which each actor has a different role in the hospital. The proposed risk-based model should validate its applicability on this scenario by allowing or denying tasks for each role. Generally, only doctors have the ability to perform both read and write operations on the EPR, while other actors including nurses, social workers and policemen have the ability to only read/view the EPR. For the resource sensitivity, two sensitivity levels were assumed; sensitive and not sensitive. However, all data/resources involved in this scenario was assumed sensitive.

To define the value of the action severity, Sharma et al. (2012) formula was utilized. This formula is used to estimate the risk score of action severity in terms of various actions, risk probability, and cost regarding data availability, integrity, and confidentiality. The formula is represented as:

$$\text{Risk} = (C \times P) + (I \times P) + (A \times P) \quad (8.1)$$

Where C, I and A represents confidentiality, integrity, and availability respectively and P represents the probability. In addition, Sharma et al. (2012) have suggested some actions and corresponding values of the CIA, as shown in Table 8.2. Therefore, values of action severity of the proposed risk-based model will be estimated using this table.

Table 8.2: Risk values associated with action and data sensitivity (Sharma et al., 2012)

Action	Sensitivity	C	I	A
Create	Sensitive/Not-Sensitive	0	1	1
View	Sensitive	1	0	0
View	Not-Sensitive	0	0	1
Modify	Sensitive/Not-Sensitive	0	1	1
Delete	Sensitive/Not-Sensitive	0	1	1

For instance, if a user needs to perform a "view" operation on sensitive data and the probability of this incident was 0.4. Therefore, only confidentiality will be affected, and the risk value of the action severity will be 0.4. Healthcare data have serious importance in almost all hospitals. Several security solutions are employed to ensure the security and privacy of patients' data. Therefore, all data or resources involved in this scenario were assumed to be sensitive, and with the probability of 0.4, the value of the resource sensitivity will be 0.8.

Table 8.3: The risk value of user context of various actors involved in this scenario

Actor	On duty (Time)	Location (In Hospital)	User Context Risk level	Proposed UC value
Doctor	Yes	Yes	Low	0.25
	No	Yes	Moderate	0.5
	No	No	High	0.75
Nurse	Yes	Yes	Low	0.25
	No	Yes	Moderate	0.5
	No	No	High	0.75
Social Worker	Yes	Yes	Low	0.25
	No	Yes	Moderate	0.5
	No	No	High	0.75
Policeman	Yes	Yes	Low	0.25
	No	Yes	Moderate	0.5
	No	No	High	0.75

For the contextual and real-time attributes (user context) that are collected at the time of making the access request, the time and location features were utilized. The time refers to the time of duty for the hospital staff whether doctor or nurse in which if the doctor requested access to data during his/her time of duty (time allocation), the risk associated with the time context feature will be low otherwise it will be high. Also, the location was utilized to determine the risk associated with contextual attributes in which if the actor requested access to data from inside the hospital, the risk will be low, otherwise, the risk will be high.

The value of user context was assumed, as shown in Table 8.3. In addition, since actors involved in this scenario involving doctor, nurse, social worker and policeman are officially employed in the hospital, they are trusted users and hence their risk history was assumed to be low. Therefore, the value of the risk history will be 0.25.

8.3.1.4 Scenario Results

Determining the access decision depends on the estimated risk value associated with each access request. The estimated risk value is compared against output risk bands to decide whether granting or denying access. Access decisions bands were assumed, as depicted in Table 8.4.

Table 8.4: Proposed output risk bands for the scenarios

Risk Band	Access Decision
0.1 – 0.25	Access Granted
0.26 – 0.7	Access Granted with Monitoring
0.7 – 1.0	Access Denied

All access control scenarios of the MC children hospital were implemented and the access decision for each scenario was decided, as shown in Table 8.5. The risk value for each access request was estimated using the NFS model with the LM learning algorithm. For doctors, all their access requests to read/view patients' EPR were granted as soon as they are located inside the hospital even though they were not on duty. This is because some emergency cases came to the hospital with no available doctors, so the system should allow the doctor to read patient's EPR to help the patient until available on duty doctor be allocated for the patient.

Table 8.5: Access decisions of various scenarios of the MC children hospital

Actor	On Duty	In Hospital	Action	Risk Factors				Output Risk	Access Decision
				UC	RS	AS	RH		
Doctor	Yes	Yes	Read	0.25	0.8	0.4	0.25	0. 498	Access Granted with Monitoring
	No	Yes	Read	0.5	0.8	0.4	0.25	0. 637	Access Granted with Monitoring
	No	No	Read	0.75	0.8	0.4	0.25	0. 749	Access Denied
	Yes	Yes	Write	0.25	0.8	0.8	0.25	0. 600	Access Granted with Monitoring
	No	Yes	Write	0.5	0.8	0.8	0.25	0. 721	Access Denied
	No	No	Write	0.75	0.8	0.8	0.25	0. 822	Access Denied
Nurse	Yes	Yes	Read	0.25	0.8	0.4	0.25	0. 498	Access Granted with Monitoring
	No	Yes	Read	0.5	0.8	0.4	0.25	0. 637	Access Granted with Monitoring
	No	No	Read	0.75	0.8	0.4	0.25	0. 749	Access Denied
Social Wo.	Yes	Yes	Read	0.25	0.8	0.4	0.25	0. 498	Access Granted with Monitoring
	No	Yes	Read	0.5	0.8	0.4	0.25	0. 637	Access Granted with Monitoring
	No	No	Read	0.75	0.8	0.4	0.25	0. 750	Access Denied
Policeman	Yes	Yes	Read	0.25	0.8	0.4	0.25	0. 498	Access Granted with Monitoring
	No	Yes	Read	0.5	0.8	0.4	0.25	0. 637	Access Granted with Monitoring
	No	No	Read	0.75	0.8	0.4	0.25	0. 749	Access Denied

The read operation is denied for doctors only if they were not on duty time and outside the hospital. Since the write operation involves writing medication and ordering examinations, doctors are granted

to perform the write operation only if there are on their duty time and inside the hospital, so the risk value associated with their contextual attributes should be low.

For nurses, social workers and policemen, as they are only allowed to view/read patient's EPR based on their role, only the read scenario was discussed. They are able to read the patient's EPR whether they are on duty time or not. This gives more flexibility as they already inside the hospital and their action will not cause serious harm. Also, it can take some time until they allocate on duty person who can deal with the current case. Their access to the read operation is denied only if they were not on duty time and outside the hospital.

8.3.1.5 Scenario Discussion

Applying the proposed risk-based access control model demonstrated it can provide several advantages to the healthcare domain. Using contextual and real-time features involving time and location demonstrated it can provide dynamic and flexible access decisions that can adapt to unpredicted situations. Allowing the doctors to access the patient's EPR even after finishing their duty time allow them to help the patient until an available on duty doctor is allocated. In addition, one of the important aspects of applying the proposed risk-based model in the healthcare domain is denying access whether read or write operation for all actors involved in this scenario when they are not on duty and outside the location of the hospital. This adds more security to the healthcare system compared to the existing systems in which if one actor lost his/her credentials (for example password) through social engineering or any other type of attack, this can lead to information disclosure. Using the proposed risk-based model with contextual and real-time features, no one can access the data only if they are inside the hospital and within their duty time besides other credentials. A comparison between existing access control models and the proposed risk-based model in the healthcare scenario is shown in Table 8.6.

Applying the proposed risk-based model on access control scenarios of the MC children hospital demonstrated it can provide flexible and effective access control model that can use contextual and real-time features to provide access decisions. It solves issues associated with static policies that always give the same result in different situations. For example, it allows access if the actor located in the hospital location. It also solves issues associated with misuse and credential loss by allowing access by actors in person. In addition, using smart contracts to monitor and track the access session add another layer of security to detect and prevent malicious actions in a timely manner. In conclusion, the proposed risk-based access control model is applicable in the healthcare IoT application and it provides efficient and effective security solution.

Table 8.6: Comparison between existing access control and the proposed risk-based model in the healthcare

Item	Existing Access Control	Proposed Model
Expected behaviour	The actor is granted access to system resources only with credential information such as a password. This access is not limited by a certain operation. For example, the doctor can perform all tasks such as read and write.	The actor is granted access to system resources when he/she is located inside the hospital which provides more security. The access to the write operation for a doctor, for example, is allowed only if the doctor is inside the hospital and during his/her duty time.
	Having a website or web interface for the hospital make the actor able to access it from any location which may cause serious issues regarding data protection and privacy preservation, especially if credential information was lost or stolen.	Using location and time features allow actors to access system resources only if there are located inside the hospital. Being outside the hospital deny all access to system resources for all actors which is an advantage to prevent misuse of credentials.
Misuse	One of the serious issues to secure a system is the misuse of employees. For example, one of the actor credentials can be used to access system resources and perform malicious actions while the actor is on a vacation. Also, one can use a social engineering attack to get credential information and use it maliciously.	The use of contextual features prevents most of misuse scenarios as actors are allowed to access system resources only if they are personally existing in the hospital location. Therefore, if the credential information was lost, the actor must be in person and during his/her duty time to access system resources.
Monitoring access	Granting access without tracking the behaviour of the actor can lead to serious insider attacks. For example, a doctor can use patients' information for marketing or research purposes.	Using smart contracts to monitor actors' behaviour can prevent insider attacks. So, if an actor deceives the system, the monitoring module will detect and prevent such kinds of malicious attacks.

8.3.2 Access Scenario 2: Smart Home

Smart home has become one of the popular IoT applications that provides new digitized services to improve our quality of life. Providing an efficient and effective access control model is one of the top priorities of a smart home. With the capability of home appliances to connect and communicate together over the Internet, protecting these devices has become essential.

This section discusses applying the proposed risk-based access control model on various access control scenarios of the smart home IoT application.

8.3.2.1 Scenario Description

The IoT has the capability to connect almost all environment objects over the Internet to share their data and create new applications and services. Using a software application can control smart home appliances to enable or disable them. For example, smart thermostats can be controlled remotely to control the home temperature. This allows the device's owner to control the home's temperature for more comfortable when back home. In addition, food can be cooked while you are on your way to

home with the capability to control the Oven or Microwave remotely to turn it on or off and control the temperature.

In this scenario, some smart appliances, which are part of the smart home, were utilized to validate the applicability of the proposed risk-based model in smart home applications. These appliances were divided into two groups based on actions that can be performed on these appliances.

- The first group of appliances can be controlled using three actions: enable (ON), disable (OFF), and adjust (to set a value for a certain feature). This group of devices includes Oven, Microwave, Washing Machine, TV control, Temperature Control, and others.
- The second group of appliances can be controlled using two actions: open and close such as door and window locks, etc.

8.3.2.2 Scenario Assumption

Applying the proposed risk-based access control model on a smart home access control scenario needs specifying the four risk factors for each access request. For contextual and real-time attributes (user context), time and location were utilized. The time refers to the time of accessing a certain device. If the access was done, for example, between 9:00 AM- 17:00 PM, the risk will be high, since the owner will be at work at this time. While if the access was done outside this time, the risk will be low, since the device's owner will be at home at this time. The selected time interval can be set dynamically using the system owner. The location refers to the location of the requesting user while making the access request to access home devices. If the access was made from inside the home, then the risk will be low, while if the access was made from outside the home, the risk will be high.

The value of user context was assumed, as shown in Table 8.7. Only two risk levels were used; low and high to represent all combinations of location and time features. The risk of contextual features will be low if the device owner is accessing the device from inside the home whether within permitted time or not, as this should be the case in real-life scenarios. Also, the risk will be low, if the device owner is accessing system within permitted time whether inside or outside the home, since one of the significant features of smart devices is to operate and control it remotely. While the risk will be high if the access was made outside the permitted time and from outside the home.

Table 8.7: The value of user context for smart home access control scenario

Permitted Time	**Location (In-Home)**	**User Context Risk level**	**Proposed Value**
17:00 PM – 08:59 AM			
Yes	Yes	Low	0.25
No	Yes	Low	0.25
Yes	No	Low	0.25
No	No	High	0.75

For the resource sensitivity and action severity, since ON, OFF, and Adjust actions for the first group and Open and Close for the second group of appliances are simple actions for most devices, hence, the action severity was assumed to be low. In addition, since all smart home appliances are closely related to human life and can be used in a malicious way to literally cause people to lose their lives, all appliances/ data in this scenario were assumed to be sensitive. Values of action severity and resource sensitivity are shown in Table 8.8. Values of resource sensitivity and action severity were decided based on Sharma et al. (2012) formula that was discussed earlier in section 8.3.1.3.

Table 8.8: Values of resource sensitivity and action severity

Group	Smart home appliance	Resource Sensitivity	Action	Action Severity
First	Oven, Microwave, washing machine, TV and Temperature Control	0.8	ON	0.4
		0.8	OFF	0.4
		0.8	Adjust	0.4
Second	Door and window lock	0.8	Open	0.4
		0.8	Close	0.4

For the risk history, two values; low (0.25) and high (0.75), were assumed since the smart home device can be accessed by either the device owner who has a low-risk history or a malicious attacker who wants to perform malicious actions to gain access or steal sensitive information. Therefore, two risk history values were utilized with all smart home access control scenarios.

8.3.2.3 Scenario Results

Deciding the access decision depends on the estimated risk value associated with each access request. The estimated risk value is compared against output risk bands to decide whether granting or denying access. Access decisions bands were assumed, as depicted in Table 8.9.

Table 8.9: Access decision bands for smart home access scenarios

Risk Band	Access Decision
0.1 – 0.25	Access Granted
0.26 – 0.75	Access Granted with Monitoring
0.75 – 1.0	Access Denied

After specifying values of the four risk factors of the proposed risk-based model, the output risk value for each scenario was estimated using the NFS with the LM learning algorithm, as depicted in Table 8.10. For the first group of appliances, most access requests were granted. With 0.4 in the action severity and 0.8 in the resource sensitivity, all access requests were granted if the value of user context or the risk history was low. This is logical and reflects real-life scenarios, in which if the owner is inside the home and requesting to access the device in the permitted time interval (low contextual), the access should be permitted. Also, since one of the main features of smart devices is

the ability to access it remotely, the proposed model allows the device's owner to access various devices remotely. The device's owner can perform ON, OFF and Adjust actions if the value of user context or risk history was low.

Table 8.10: Applying the proposed model on access control scenarios of a smart home

Smart home appliance	Context Features	Risk History	Action	Risk Factors				Output Risk	Access Decision
				UC	RS	AS	RH		
Oven, Microwave, washing machine, TV Control, Temperature Control and other	Low	Low	ON	0.25	0.8	0.4	0.25	0.498	Access Granted with monitoring
	High	Low	ON	0.75	0.8	0.4	0.25	0.749	Access Granted with monitoring
	Low	Low	OFF	0.25	0.8	0.4	0.25	0.498	Access Granted with monitoring
	High	Low	OFF	0.75	0.8	0.4	0.25	0.749	Access Granted with monitoring
	Low	Low	Adjust	0.25	0.8	0.4	0.25	0.498	Access Granted with monitoring
	High	Low	Adjust	0.75	0.8	0.4	0.25	0.749	Access Granted with monitoring
	Low	High	ON	0.25	0.8	0.4	0.75	0.582	Access Granted with monitoring
	High	High	ON	0.75	0.8	0.4	0.75	0.808	Access Denied
	Low	High	OFF	0.25	0.8	0.4	0.75	0.582	Access Granted with monitoring
	High	High	OFF	0.75	0.8	0.4	0.75	0.808	Access Denied
	Low	High	Adjust	0.25	0.8	0.4	0.75	0.582	Access Granted with monitoring
	High	High	Adjust	0.75	0.8	0.4	0.75	0.808	Access Denied
Door and window lock	Low	Low	Open	0.25	0.8	0.4	0.25	0.498	Access Granted with monitoring
	High	Low	Open	0.75	0.8	0.4	0.25	0.582	Access Granted with monitoring
	Low	Low	Close	0.25	0.8	0.4	0.25	0.498	Access Granted with monitoring
	High	Low	Close	0.75	0.8	0.4	0.25	0.749	Access Granted with monitoring
	Low	High	Open	0.25	0.8	0.4	0.75	0.582	Access Granted with monitoring
	High	High	Open	0.75	0.8	0.4	0.75	0.808	Access Denied
	Low	High	Close	0.25	0.8	0.4	0.75	0.582	Access Granted with monitoring
	High	High	Close	0.75	0.8	0.4	0.75	0.808	Access Denied

On the other hand, the access was denied for this group of devices only if values of both user context and risk history were high. This is also logical as it reflects the fact that the malicious user with a high risk history who requested to access the device from outside the home and outside the permitted time interval should not able to access the device.

For the second group of devices, in the same way, most access requests were granted. The access was granted for Open and Close actions if the value of user context or risk history was low. This allows the device's owner to access various devices either from inside or outside the home easily and securely. In addition, the access was denied to perform Open and Close actions only if values of both user context and risk history were high.

8.3.2.4 Scenario Discussion

Applying the proposed risk-based model on smart home access scenarios demonstrated it can provide several advantages over existing access control models. Using the contextual and real-time features involving time and location demonstrated it can provide dynamic and flexible access decisions.

The proposed risk-based model provides the expected functionality like existing access control models in which it allows the owners to perform all actions on various devices as soon as they are at the home. In addition, it allows the device's owner to access various appliances and perform various actions on different appliances remotely and securely.

For both groups of appliances, the access was granted with monitoring access sessions to detect and prevent malicious actions. Monitoring access sessions using smart contracts to detect and prevent malicious actions in the smart home provides an effective solution to control access to smart appliances. In addition, one of the important features of the proposed risk-model is the flexibility of selecting risk decision bands, which make the device' owner has full control of granting or denying access. As a conclusion, the results of the access control scenarios in the smart home demonstrated that the proposed risk-based access control can be applied efficiently and effectively in smart home applications.

8.3.3 Access Scenario 3: Network Router

To validate the applicability of the proposed risk-based access control model in real-world access applications, access control scenarios of the network router will be presented to show different access decisions in various situations based on the estimated risk value for each access request. The network router is an electronic device designed to connect at least two networks and forwards packets among them based on the information existed in the packet header and the routing table. The router is a fundamental element to the operation of the Internet and other complex networks (Kim et al., 2014; Shuzhao & Zhaohui, 2014).

8.3.3.1 Scenario Description

The network router is one of the significant elements to set up a network. There are two methods to access a network router; console and telnet connection. Router console connection is used to connect end devices, such as PC to the router to manage its configurations using a rollover cable connection. While telnet connection is used to configure the router remotely through a router virtual terminal.

To provide access control scenarios of the network router, three parameters need to be specified:

- **Router data to be accessed**: a user will access the router only to perform certain operations on certain data. Therefore, different router data and operations should be specified.
- **Values of four risk factors of the proposed risk model**: to calculate the risk value for each access request, values of user context, resource sensitivity, action severity and risk history need be specified.
- **Router acceptable risk values**: after estimating the risk value associated with the access request, the access decision should be decided to either grant or deny the access.

8.3.3.2 Scenario Assumption

To determine values of resource sensitivity and action severity of the network router, different types of data of the router need to be specified. Data that can be accessed through the router can be categorized as follows:

- **Non-Volatile Random-Access Memory (NVRAM)**: is used to store start-up configuration files of the router.
- **Dynamic Host Configuration Protocol (DHCP)**: allocates IP address information to various devices in the network dynamically.
- **Flash Memory**: is used to store the router internetworking operating system.
- **Configuration Passwords**: are router passwords that are required to enter different router configuration modes to add or edit configuration commands.
- **Routing Table**: is used by the router to determine the best path to forward packets to its destination. Without routing table, all router packets will be discarded.

The router data were classified in terms of actions severity and data sensitivity, as shown in Table 8.11. The data sensitivity level is based on the action to be performed. For instance, “View” operation is not sensitive while “Delete” operation is sensitive on the same NVRAM data.

Table 8.11: Data sensitivity with different actions regarding router data

Router Data	Action	Sensitivity
NVRAM data	View	Not Sensitive
	Delete	Sensitive
	Modify	Sensitive
DHCP data	View	Not Sensitive
	Modify	Sensitive
	Create	Sensitive
Flash Data	Delete	Not Sensitive
Configuration passwords	View	Sensitive
	Modify	Sensitive
Routing table	View	Not Sensitive
	Delete	Sensitive

The values of action severity and data sensitivity were decided using Sharma et al. (2012) formula and Table 8.2 that were discussed earlier in section 8.3.1.3. For the user context, the location of the requester was utilized to estimate the risk value associated with the user context. Only two values; low and high, was used to indicate whether the requester is at the router location or not at the time of making the access request. So, if the requesting user is at the physical location of the router, the risk of user context will be 0.25, otherwise, the risk of the user context will be 0.75.

8.3.3.3 Scenario Results

Deciding the access decision depends on the estimated risk value associated with each access request. The estimated risk value is compared against output risk bands to decide whether granting or denying access. The same risk decision bands employed in the healthcare scenario was utilized in this scenario, as depicted in Table 8.4. After specifying values of the four risk factors of the proposed risk-based model, the output risk value for each access scenario was estimated using the NFS with the LM learning algorithm.

8.3.3.3.1 Console Connection

Suppose a user wants to manage configurations of the router through the console connection. The router was initially configured but the user wants to access the router to perform other operations. Since the user has the ability to reach the physical location of the router and attach the rollover cable to connect the router to his/her end device, so he/she will be considered as a trusted user with low risk history and low user context value. Therefore, values of risk history and user context were assumed 0.25. In addition, values of resource sensitivity and action severity were calculated using Sharma et al. (2012) formula with a risk probability of 0.4. Table 8.12 shows different access control scenarios of the router through the console connection.

Table 8.12: Access control scenarios of the network router through the console connection

Router Data	Action	Risk Factors				Output Risk	Access Decision
		UC	RS	AS	RH		
NVRAM data	View	0.25	0.4	0.4	0.25	0.358	Access Granted with Monitoring
	Delete	0.25	0.8	0.8	0.25	0.600	Access Granted with Monitoring
	Modify	0.25	0.8	0.8	0.25	0.600	Access Granted with Monitoring
DHCP data	View	0.25	0.4	0.4	0.25	0.358	Access Granted with Monitoring
	Modify	0.25	0.8	0.8	0.25	0.600	Access Granted with Monitoring
	Create	0.25	0.8	0.8	0.25	0.600	Access Granted with Monitoring
Flash Data	Delete	0.25	0.8	0.8	0.25	0.600	Access Granted with Monitoring
Configuration passwords	View	0.25	0.4	0.4	0.25	0.358	Access Granted with Monitoring
	Modify	0.25	0.8	0.8	0.25	0.600	Access Granted with Monitoring
Routing table	View	0.25	0.4	0.4	0.25	0.358	Access Granted with Monitoring
	Delete	0.25	0.8	0.8	0.25	0.600	Access Granted with Monitoring

The estimated risk value for different scenarios to access the router through the console connection was small, so all access requests were granted with monitoring. This seems logical to the real-world scenarios as if the user has the ability to reach the router location and attach the cross over cable to the router, the user should be able to perform all various actions on various data whether sensitive or not sensitive.

8.3.3.3.2 Telnet Connection

Consider a user who wants to manage configurations of the router remotely. The router was initially configured but the user wants to access the router to perform other operations. Since the user is accessing the router from a remote location, the user context will be high. Also, the user risk history was assumed to have two values (high and low) as the router owner can access the router remotely and the malicious user as well. Therefore, the risk history has two values 0.25 and 0.75. The access to the router via the telnet connection will be similar to console connection in terms of values of actions severity and data sensitivity on the router data. Also, the risk probability was assumed to be 0.4.

Table 8.13: Access control scenarios of the network router through the telnet connection

Router Data	Action	Risk Factors				Output Risk	Access Decision
		UC	RS	AS	RH		
NVRAM data	View	0.75	0.4	0.4	0.25	0.609	Access Granted with Monitoring
		0.75	0.4	0.4	0.75	0.668	Access Granted with Monitoring
	Delete	0.75	0.8	0.8	0.25	0.822	Access Denied
		0.75	0.8	0.8	0.75	0.880	Access Denied
	Modify	0.75	0.8	0.8	0.25	0.822	Access Denied
		0.75	0.8	0.8	0.75	0.880	Access Denied
DHCP data	View	0.75	0.4	0.4	0.25	0.609	Access Granted with Monitoring
		0.75	0.4	0.4	0.75	0.668	Access Granted with Monitoring
	Modify	0.75	0.8	0.8	0.25	0.822	Access Denied
		0.75	0.8	0.8	0.75	0.880	Access Denied
	Create	0.75	0.8	0.8	0.25	0.822	Access Denied
		0.75	0.8	0.8	0.75	0.880	Access Denied
Flash Data	Delete	0.75	0.8	0.8	0.25	0.822	Access Denied
		0.75	0.8	0.8	0.75	0.880	Access Denied
Configuration passwords	View	0.75	0.4	0.4	0.25	0.609	Access Granted with Monitoring
		0.75	0.4	0.4	0.75	0.668	Access Granted with Monitoring
	Modify	0.75	0.8	0.8	0.25	0.822	Access Denied
		0.75	0.8	0.8	0.75	0.880	Access Denied
Routing table	View	0.75	0.4	0.4	0.25	0.609	Access Granted with Monitoring
		0.75	0.4	0.4	0.75	0.668	Access Granted with Monitoring
	Delete	0.75	0.8	0.8	0.25	0.822	Access Denied
		0.75	0.8	0.8	0.75	0.880	Access Denied

Table 8.13 shows access control scenarios of the network router through the telnet connection. Most access requests were denied. This is because values of user context and risk history were assumed to be high. Only view action was granted for different router data when the risk history either high or low except configuration passwords. This seems to be fine in terms of security as it is a view/read operation, which will not cause any harm especially these data have been categorized as not sensitive.

For configuration password, since viewing these passwords will allow the user to access most system resources, it has been categorized as sensitive, so the resource sensitivity and action severity were assumed to be 0.8. This makes the output risk to be high, and the access was denied.

The access was granted only when values of risk history, action severity, and resource sensitivity were low. In other words, the access was granted only when the value of the risk history was 0.25 and values of both action severity and resource sensitivity were 0.4. Most access via the telnet connection was denied. This is due to the fact that Telnet as a protocol has several drawbacks in term of security. For instance, it uses a plaintext to send or receive data without any encryption. This leads to several security attacks such as eavesdropping and snooping which are easier to employed by malicious attackers.

8.3.3.4 Scenario Discussion

Applying the proposed risk-based model on the network router access scenarios demonstrated it can provide the expected functionality like other access control system with adding new advantages. One of these advantages is the use of contextual and real-time features to provide access decisions. The contextual information plays a significant role to decide access decisions. For example, when the user requested to perform “Delete” action on the NVRAM data and the risk metric of contextual features was 0.25, the output risk value was 0.6 and the access was granted. While when the user requested to perform the same action on the same data but with a high risk metric for contextual features, which was assumed 0.75, the output risk value was 0.8217, and the access was denied.

The proposed risk-based model provided the expected functionality that allows the owner of the router to perform all actions on various data as soon as he/she can reach to the location of the router and attach the rollover cable to it. In addition, all access was granted with monitoring access sessions to detect and prevent malicious actions. This monitoring feature adds another layer of security to secure access to various data of the network router. In conclusion, the results demonstrated that proposed risk-based access control is applicable to the access control scenarios of the network router and it provides efficient and effective security solution

8.3.4 Comparison with Current Risk Models

Current access control models are based on static and predetermined policies that cannot satisfy the required flexibility needed in various IoT applications. While the proposed risk-based access control model provides a dynamic approach by using real-time and contextual features collected from the IoT environment while making the access request to make the access decision. Reviewing the existing and related risk-based access control models demonstrated that no previous research has employed real-time features collected from the IoT environment while making the access request in IoT applications, as depicted in Table 8.14.

Table 8.14: Risk factors utilized to build risk-based access control models

Related Model	Subject clearance	Object clearance	Resource Sensitivity	Action Severity	Risk History	Subject Trust	Risk Policies	Real-time Features
Zhang et al. (2006)				✓				
Britton & Brown (2007)	✓	✓			✓	✓		
Chen et al. (2007)	✓	✓						
Lee et al. (2007)				✓				
Bertino & Lobo (2010)	✓	✓						
Rajbhandari & Snekkenes (2011)	✓			✓				
Wang & Jin (2011)			✓					
Shaikh et al. (2012)					✓			
Sharma et al. (2012)			✓	✓	✓			
Khambhammettu et al. (2013)			✓			✓		
Li et al. (2013)			✓	✓	✓			
Namitha et al. (2015)	✓							
Choi et al. (2015)				✓				
Chen et al., (2016)					✓	✓		
Dos Santos et al., (2016)							✓	
Abomhara et al., (2018)		✓				✓		
Proposed Model			✓	✓	✓			✓

In addition, the proposed risk-based access control model was compared with related risk-based access control models that utilized the fuzzy logic system in the risk estimation process, as depicted in Table 8.15. The proposed risk-based model provides a dynamic and context-aware approach by using real-time and contextual features associated with the user at the time of making the access request as a risk factor besides resource sensitivity, action severity and risk history to estimate the risk value associated with each access request to decide the access decision.

In addition, fuzzy rules are the core of the fuzzy logic system which need to be built accurately to yield a precise risk value for each access request. Although fuzzy rules are built on expert knowledge, there is no evidence or any details about using security experts to build fuzzy rules in related fuzzy risk-based access control models discussed in the literature. In this research, Twenty IoT security experts from inside and outside the UK were interviewed to build fuzzy rules. This number of experts adds more robustness and accuracy to the research. In addition, interviewing IoT security experts

reduced the subjectivity of the risk estimation process. Indeed, the subjectivity was not completely eliminated. However, it is unlikely that a method with no subjectivity will ever exist for risk analysis.

Table 8.15: Comparison between the proposed model with existing fuzzy-based risk models

Items	**Chen et al. (2007)**	**Ni et al. (2010)**	**Li et al. (2013)**	**Proposed Model**
Risk factors	Difference between subject security level and object security level	Object security level and subject security level	Data sensitivity, action severity, and user risk history	Contextual features of the user, resource sensitivity, action severity and risk history
Context-awareness	Not Context-aware	Not Context-aware	Not Context-aware	Context-aware
Fuzzy rules	Fuzzy rules were built by authors	Fuzzy rules were built by authors	Fuzzy rules were built by authors	Twenty IoT security experts were interviewed to build fuzzy rules
Subjectivity	High subjectivity	High subjectivity	High subjectivity	Less subjectivity
Validation	No proof of validation	No proof of validation	No proof of validation	Validated by Twenty IoT security experts
Scalability	Not Tested	Not Tested	Not Tested	Tested with a large number of access requests
Solution to cold start	N/A	N/A	Not Exist	Exist, a solution to cold start was provided
Learning Capability	Not Exist	Not Exist	Not Exist	Exist as ANFIS and NFS were applied
Monitoring Capability	Not Exist	Not Exist	Not Exist	Smart contracts were utilized to monitor access sessions

In addition, to overcome the time overhead of the fuzzy logic system, the proposed risk estimation technique was implemented using ANFIS and NFS. After comparing processing time with a large number of access requests, the resultant NFS model demonstrated it provides fast and scalable risk estimation technique. Also, the use of ANFIS and NFS with the proposed risk-based model adds the learning capability that allows the proposed risk-based model to adapt to new changes of various IoT applications.

Lastly, utilizing smart contracts to monitor access sessions provided significant improvements over existing access control models. Reviewing related risk-based access control models demonstrated that no previous research has employed smart contracts in this context. The ability to detect and prevent malicious attacks in a timely manner provided another layer of security. In addition, smart contracts added a new dimension to the next research about integrating the blockchain technology with existing centralized models to provide better and effective security solutions.

8.4 Summary

This chapter has presented the simulation of smart contracts and the evaluation of the proposed risk-based model. The chapter was divided into two main parts. The first part discussed smart contracts in access monitoring. Existing access control models do not provide a way to detect malicious actions and protect system resources after granting access. While the proposed risk-based model added abnormality detection capability by utilizing smart contracts to track and monitor user's activities during access sessions to detect and prevent malicious actions. MATLAB Simulink was utilized to simulate the operation of smart contracts to validate its efficiency and effectiveness to monitor access sessions. After discussing different scenarios, the results demonstrated that smart contracts can provide an effective and efficient way to monitor user activities and prevent malicious actions in a timely manner. The second part of the chapter discussed the evaluation of the proposed risk-based model using access control scenarios of three IoT applications including healthcare, smart home and network router. The results demonstrated that the proposed risk-based model is applicable to various IoT application and it provides efficient and effective security solution. The next chapter concludes the thesis and present future work.

Chapter 9: Conclusion and Future Work

This chapter summarises the results and findings reached to answer the research questions. It also discusses the contributions made by this research. Future research directions are also explored.

9.1 Conclusion

Currently, the IoT becomes a broadly examined subject among researchers, specialists and experts. It is considered as the next stage of the evolution of the Internet. Although the Internet has passed several stages since it was invented, as it switched from a couple of PCs communicating with each other to billions of computational devices and billions of cell phones over time. With the IoT, we are moving towards a phase where almost all items in our environment will be connected and communicate together with minimum human efforts. The IoT is considered as a universal presence in the environment that contains a variety of things that can be connected whether using wireless and wired connections. These things have a unique addressing scheme that allow them to interact and cooperate with others to create new IoT applications and services such as smart homes, smart cities, smart energy and the smart grids, smart transportation and traffic management and control and others.

The IoT has several benefits in various domains, but it also creates multiple issues that need to be addressed to continue adopting IoT applications. One of these issues is security that has a major impact literally on people lives. This is due to the fact that the IoT is a dynamic system in nature in which every poorly secured object can disturb the security and resilience of the entire system, as they are connected like a chain. The ease of connection and access of IoT devices open doors for severe security issues especially with the large-scale distribution of heterogamous devices, their ability to connect to other devices without requesting permissions or even notifying their owners and probability of flooding these devices with severe security threats.

One of the solutions to address security issues in the IoT system is to build an efficient and effective access control model. This model not only limits access to authorised users but also prevents authorised users from accessing system resources in an unauthorised way. However, the existing access control models are rigid and use static policies to provide access decisions. This static

approach gives the same result in different situations, which cannot provide an adequate level of security in a dynamic system like the IoT. Therefore, there is a need to adopt dynamic access control models for the IoT. These models use not only access policies but also real-time information to provide access decisions. One of the dynamic models is risk-based access control. This model uses the security risk value associated with the access request to determine the access decision. This model solves the issue regarding the flexibility in accessing system resources. In addition, it provides a dynamic and efficient solution to unpredicted situations, especially in healthcare and military applications, where granting access can save thousands of lives.

The research questions were addressed through this research study. The findings for each research question are briefly presented as follows:

The main research question was:

RQ: What is the appropriate adaptive risk-based access control model for the IoT system?

The major target of this research was to develop a dynamic and adaptive risk-based access control model that can provide an effective security solution for various IoT applications. Therefore, a dynamic and adaptive risk-based access control model was proposed, as discussed in section 4.3. This model utilizes contextual and real-time features collected from the IoT environment while making the access request to determine access decisions. It has four inputs; user contextual features, resource sensitivity, action severity and risk history. In addition, to detect and prevent abnormal misuse from authorized users during the access session, the proposed model utilized smart contracts to monitor user's activities and adjust their risk values adaptively based on their actions.

The main research question was divided into six sub-questions as follows:

SRQ1: What is the appropriate risk estimation technique to estimate the risk associated with the access request?

Specifying the optimal risk estimation technique to assess security risks of access control operations in IoT systems faces several issues. For example, the core drive of the risk estimation process is to expect the future likelihood of information disclosure that is corresponded to the current access. Identifying this likelihood in the absence of a dataset is a very difficult task. In addition, the IoT system requires a flexible and scalable risk estimation technique that can adapt to growing rates of the number of IoT devices and changing conditions during making access decisions. After reviewing related risk estimation techniques in the literature, the fuzzy logic approach with expert judgment was selected as the appropriate risk estimation technique, as discussed in chapter 3. In addition, Twenty IoT security experts from inside and outside the UK were interviewed to validate the proposed risk-based access control model and build fuzzy rules. The proposed risk estimation technique was implemented using MATLAB, as discussed in chapter 5. The results demonstrated

that the fuzzy logic system with expert judgment generates accurate and realistic risk values for access control operations.

SRQ2: What are acceptable risk values to make the access decision in IoT applications?

The risk-based access control model works by estimating the security risk value associated with each access request. Then, the estimated risk value is compared with a threshold risk value to decide the access decision. After reviewing the literature, most presented risk-based access control models suggested using a threshold risk value to grant or deny access without providing any details about how to decide this threshold risk value in different applications. Therefore, in this research, three risk decision bands were proposed involving allow, allow with risk monitoring, and deny. Then, twenty IoT security experts were interviewed to decide acceptable risk values for risk decisions bands, as discussed in section 5.4.

SRQ3: How to provide plug and play risk-based model that can work when first used or connected?

As discussed earlier in section 4.2, some related risk-based access control models used the risk history as a risk factor to determine access decisions. However, values of risk history will not be available at the start of setting up the new risk-based model, which will make the system unusable until collecting these values. To overcome this cold start problem, a solution was provided by adding additional twenty-seven fuzzy rules that use only three risk factors (user context, resource sensitivity and action severity). To validate these rules, ten IoT security research fellow from the University of Southampton were interviewed, as discussed in section 5.6. The results demonstrated that the proposed risk-based model can work properly when first used or connected without reconfiguration or adjustment.

SRQ4: How to provide fast and scalable risk estimation technique to handle the constant increase in the number of IoT devices?

The IoT system grows significantly. So, the risk estimation technique should be able to handle the growing rate of the number of IoT devices. As discussed in section 5.7, a set of experiments was introduced to evaluate the efficiency of the proposed fuzzy risk estimation technique. These experiments were utilized to measure the response time with different number of access requests and determine the most efficient MF, defuzzification method, and rule aggregation operator. The results of these experiments demonstrated that the scalability of the proposed risk estimation technique is questionable. In addition, it lacks the ability to learn and cannot be adjusted to new IoT environments. To solve this issue, ANFIS and NFS were utilized to implement the risk estimation technique, as discussed in chapter 6 and chapter 7 respectively. Several experiments were carried out to train the ANFIS model using hybrid and backpropagation learning methods at three different number of epochs; 20, 100, and 300. The results demonstrated that the TrapMF with the hybrid learning method at 20 epochs is the optimal combination to implement the ANFIS model of the proposed risk

estimation technique. In addition, several experiments were carried out to train the NFS model of the proposed risk estimation technique using four different learning methods. The results demonstrated that the NFS model with the LM learning method is the best approach to implement the proposed risk estimation technique to increase the accuracy, reduce the processing time needed to provide access decisions in IoT applications and adapt to new changes of various real-world IoT applications.

SRQ5: How will the user/agent activities be monitored during the access session?

Most existing access control models did not employ a method to detect malicious actions after granting access. Therefore, the proposed risk-based model added abnormality detection capability by utilizing smart contracts to track and monitor user's activities to detect and prevent malicious actions during access sessions. MATLAB Simulink was utilized to simulate the operation of smart contracts to validate its efficiency and effectiveness to monitor access sessions, as discussed in section 8.2. After discussing different scenarios, the results demonstrated that smart contracts provide an effective and efficient way to monitor user activities and prevent malicious actions in a timely manner.

SRQ6: To what extent is the proposed risk-based model applicable to real IoT scenarios?

The ultimate target of any new approach is to guarantee that it is applicable in real-world scenarios. Hence, the proposed risk-based model was evaluated using access control scenarios of three IoT applications including healthcare, smart home and network router, as discussed in section 8.3. The results demonstrated that the proposed risk-based model is applicable to various IoT application and it provides efficient and effective security solution.

9.2 Contributions

This research made the following contributions that can be beneficial to the research community:

- This research provided a novel dynamic and adaptive risk-based access control model that uses contextual and real-time information collected from the IoT environment while making the access request to estimate the risk and determine the access decision. This model can be adapted to unexpected situations and provide a flexible way to determine access decisions in various IoT application.

- Providing a clear and accurate risk estimation technique for obtaining a quantitative risk value for each access request is one of the main contributions of this research. Integrating the fuzzy logic system with expert judgment has demonstrated that it can provide accurate and realistic risk values for access control operations. In the absence of a dataset that represents risk probabilities of access control scenarios and its impact, IoT security domain experts were interviewed to provide predicted measures of risk values according to their

knowledge and experience in the form of linguistic variables. A clear and detailed implementation of the risk estimation technique using the fuzzy logic system with expert judgment was presented in this research.

- This research proposed three risk decision bands to grant or deny access. The first band grants access without monitoring, the second band grants access with monitoring, while the third band denies access. In this research, twenty IoT security experts from inside and outside the UK were interviewed to provide acceptable risk values for the three risk decision bands.

- Providing a risk-based model that can work when first used or connected without adjustments was one of the contributions of this research. One of the issues associated with existing risk-based access control models was the use of risk history as one of the risk factors. So, the risk-based model cannot operate immediately as previous risk values are needed. This research resolved this issue by presenting a solution that is based on running the proposed risk-based model immediately before collecting previous risk values.

- This research integrated the ANN with the fuzzy logic system to tune fuzzy variables and use parallel computation and learning abilities of the ANN to provide a scalable and fast risk estimation technique that can cope with the constant increase of the number of IoT devices and provide access decisions in a timely manner. The ANFIS and NFS were utilized to implement the risk estimation technique. The results demonstrated that combining ANN with the fuzzy logic system have outperformed results produced by the fuzzy logic system in which it takes only one-sixth of the time taken by the fuzzy logic system to process an access request. It also added the learning capability that allows the risk estimation technique to adapt to new changes of various IoT applications.

- This research provided abnormality detection capability by using smart contracts to track and monitor user activities during the access session to detect and prevent malicious actions. MATLAB Simulink was utilized to simulate the operation of smart contracts to validate its efficiency and effectiveness to monitor access sessions. The results demonstrated that smart contracts can be used to provide an effective monitoring technique.

9.3 Future Work

The work provided by this research can be used as a foundation for future research to develop dynamic and adaptive risk-based access control models. Below are some of the proposed research directions.

9.3.1 Deep Learning Techniques

One of the major stages to build a risk-based access control model for the IoT is the risk estimation process. This process is based on estimating the possibility of information leakage and the value of that information. For the access control context, quantitative risk estimation approaches are only needed to provide a numeric value to determine the access decision. Typically, there is no universal and best method for conducting risk analysis. In addition, providing an accurate and realistic risk value for each access request for the dynamic IoT system is a very difficult process. Although combining the ANN with the fuzzy logic system has provided an efficient and fast way to estimate security risks in the IoT, deep learning techniques can be utilized to provide more improvements in terms of accuracy and performance. Deep learning provides a scalable and efficient way to teach the system by example to perform automatic feature extraction from raw data (Aziz & Dowling, 2019). It is the main technology for enabling a variety of applications such as speech recognition, social network filtering, driverless cars, bioinformatics, and audio recognition.

Like all learning approaches, deep learning techniques provide better results with large datasets. With the availability of a dataset containing more than two million data records in this research, deep learning algorithms are expected to provide better results in terms of accuracy, performance and scalability.

9.3.2 Comparative Study of Risk Estimation Techniques

Determining a suitable risk estimation technique to implement an efficient and scalable risk-based access control model for the IoT system is not easy. In this research, the fuzzy logic system with expert judgment, ANFIS and NFS were utilized to implement the risk estimation process. However, there are other machine learning techniques that can be adopted. Also, there are several deep learning algorithms that can provide better results. With the availability of a dataset in this research, a comparative study between different approaches can be utilized to determine the best approach for each context of various IoT applications in terms of accuracy, performance and scalability.

9.3.3 Integration with Standard Access Model

Integrating the proposed adaptive risk-based access control model with existing standards is one of the main objectives of future work. One of the popular standard access control frameworks is the eXtensible Access Control Markup Language (XACML) (OASIS, 2003). It is considered as one of the most promising policy languages dealing with dynamic and complex systems. It is broadly accepted by the majority of experts, communities and organizations since it is compatible with most access control models such as ACL, RBAC, and ABAC (Chen et al., 2013). Implementing the proposed risk-based access control model with attribute-based XACML model will add more advantages by utilizing both risk values associated with access request and user attributes to make

the access decision, as shown in Figure 9.1. It will also facilitate the integration of the proposed risk-based model with existing access control approaches such as ABAC and RBAC.

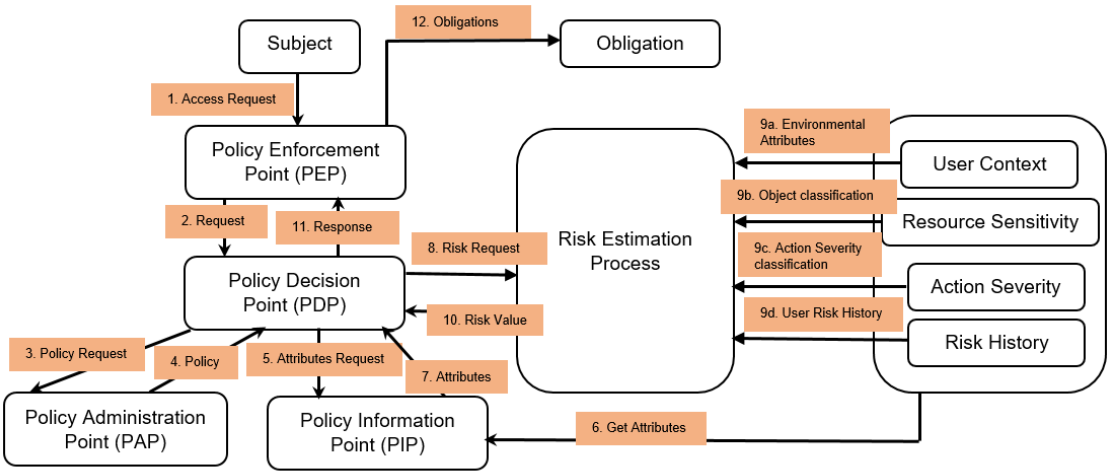


Figure 9.1: Flow of the XACML model of the proposed risk-based access control model

9.3.4 IoT Testbed for Practical Scenarios

The testbed is one of the best ways to test the applicability of new and innovative solutions on real-world operating conditions. It provides a good tool to perform various experiments to discover new technologies for generating ground-breaking products or techniques with the potential to produce new international standards (Adjih et al., 2015). Although twenty IoT security experts validated the proposed risk-based access control model, IoT testbed can provide an additional tool to carry out practical experiments on how the proposed risk-based model can provide access decisions dynamically based on contextual and real-time features collected from the IoT environment.

9.3.5 Formal Methods for Smart Contracts

This research introduced smart contracts to monitor user activities during the access session to detect and prevent malicious actions. To the best of the researcher's knowledge, no study has been used smart contracts in this context. Although MATLAB Simulink provided a simulation of the smart contract to test the system response needed to detect abnormal and malicious activities, more evaluation metrics to test the applicability of smart contracts in this context are needed. Therefore, formal methods can be utilized to evaluate the effectiveness of smart contracts in monitoring user activities in the IoT context. Formal methods are used to model complex systems, software or hardware, as mathematical entities and provide a mathematical proof to evaluate the system performance (Gaudel, 2017).

9.3.6 Privacy-aware Risk and Trust Model

Risk-based access control model provides a dynamic way to determine access decisions by utilizing the security risk value associated with each access request as the primary criterion. However, very little attention was given to privacy, which is very essential especially in the IoT context. A privacy-aware risk and trust model can be used to utilize the privacy risk value and the user trust to decide access decisions. The privacy risk refers to the impact of violating the privacy of data to be accessed by the requester. There are two main approaches to estimate data privacy: differential privacy and syntactic approaches. So, the goal is to determine the suitable approaches to estimate the privacy risk value for each access request. On the other hand, trust plays an important role to grant or deny access. In this access control model, contextual and real-time features associated with the access request will be used to estimate a trust value for each user. Then, the privacy risk value will be compared against the trust value to determine the access decision for each access request.

References

Abdur, M., Habib, S., Ali, M., & Ullah, S. (2017). Security Issues in the Internet of Things (IoT): A Comprehensive Study. *International Journal of Advanced Computer Science and Applications*, 8(6), 383-388.

Abie, H., & Balasingham, I. (2012). Risk-Based Adaptive Security for Smart IoT in eHealth. In: *Proceedings of the 7th International Conference on Body Area Networks*, (SeTTIT), pp 269-275.

Abomhara, M., M. Køien, G., Oleshchuk, V. A., & Hamid, M. (2018). Towards Risk-aware Access Control Framework for Healthcare Information Sharing. In*: Proceedings Ofthe 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, pp 312-321.

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggles, P. (1999). Towards a Better Understanding of Context and Context-Awareness. In: *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, pp 304–307.

Abraham, A. (2005). Artificial neural networks. In: *Handbook of measuring system design*, pp 97-129.

Abraham, Ajith. (2001). Neuro-Fuzzy Systems : State-of-the-art Modeling Techniques. *International Work-Conference on Artificial Neural Networks*, pp 269–276.

Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., Watteyne, T. (2015). FIT IoT-LAB: A large scale open experimental IoT testbed. In: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pp 459–464.

Ahmed, A. A. M., & Shah, S. M. A. (2017). Application of adaptive neuro-fuzzy inference system (ANFIS) to estimate the biochemical oxygen demand (BOD) of Surma River. *Journal of King Saud University - Engineering Sciences*, *29*(3), 237–243.

Airehrour, D., Gutierrez, J., & Ray, S. K. (2016). Secure routing for internet of things: A survey. *Journal of Network and Computer Applications*, *66*(April), 98–213.

Akkaş, M. A., & Sokullu, R. (2017). An IoT-based greenhouse monitoring system with Micaz motes. In: *International Workshop on IoT, M2M and Healthcare (IMH 2017)*, *113*, pp 603–608.

Al-Hmouz, A., Jun Shen, Al-Hmouz, R., & Jun Yan. (2012). Modeling and Simulation of an Adaptive Neuro-Fuzzy Inference System (ANFIS) for Mobile Learning. *IEEE Transactions on Learning Technologies*, *5*(3), 226–237.

Alberts, C. J., & Dorofee., A. (2002). Managing Information Security Risks: The Octave Approach. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Alharby, M., & Moorsel, A. van. (2017). Blockchain Based Smart Contracts: A Systematic Mapping Study. *Computer Science & Information Technology*, 125–140.

Alur, R., Berger, E., Drobnis, A. W., Fix, L., Fu, K., Hager, G. D., Zorn, B. (2015). Systems Computing Challenges in the Internet of Things. *ArXiv Preprint ArXiv:1604.02980*, (June), 1-15.

Anand Narayan. (2017). A Brief Intro To Smart Contracts And Their Endless Possibilities. Retrieved [February 2, 2019], from https://codebrahma.com/brief-intro-smart-contracts-endless-possibilities/

Ardagna, C. A., De Capitani Di Vimercati, S., Foresti, S., Grandison, T. W., Jajodia, S., & Samarati, P. (2010). Access control for smarter healthcare using policy spaces. *Computers and Security*, 29(8), 848–858.

Ashton, K. (2009). That “Internet of Things” Thing. *RFID Journal*, 4986. Retrieved from http://www.rfidjournal.com/articles/pdf?4986

Asogbon, M. G., Olabode, O., Agbonifo, O. C., Samuel, O. W., & Yemi-peters, V. I. (2016). Adaptive Neuro-Fuzzy Inference System for Mortgage Loan Risk Assessment. *International Journal of Intelligent Information Systems*, *5*(1), 17–24.

Atlam, H. F., Alenezi, A., Walters, R. J., Wills, G. B., & Daniel, J. (2017). Developing an adaptive Risk-based access control model for the Internet of Things. In: *2017 IEEE International Conference on Internet of Things (IThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, (June), pp 655–661.

Atlam, H. F., Walters, R. J., & Wills, G. B. (2018). Internet of Things : State-of-the-art, Challenges, Applications, and Open Issues. *International Journal of Intelligent Computing Research*, 9(3), 928–938.

Atlam, H. F., & Wills, G. B. (2019). Intersection Between IoT and Distributed Ledger. In: *Role of Blockchain Technology in IoT Applications, advances in computers,* pp 1-35

Aung, W. P. (2007). Analysis on Modeling and Simulink of DC Motor and its Driving System Used for Wheeled Mobile Robot. *International Journal of Electrical and Computer Engineering*, 1(8), 299–306.

Aziz, S., & Dowling, M. (2019). Machine Learning and AI for Risk Management. In: Lynn T., Mooney J., Rosati P., Cummins M. (eds) Disrupting Finance. Palgrave Studies in Digital Business & Enabling Technologies. Palgrave Pivot, Cham.

Bahga, A., Madisetti, V. K., Bahga, A., & Madisetti, V. K. (2016). Blockchain Platform for Industrial Internet of Things. *Journal of Software Engineering and Applications*, 09(10), 533–546.

Bai, Y., & Wang, D. (1982). Fundamentals of Fuzzy Logic Control – Fuzzy Sets, Fuzzy Rules and Defuzzifications. *Advanced Fuzzy Logic Technologies in Industrial Applications*, 17–36.

Balci, O. (1994). Validation, verification, and testing techniques throughout the life cycle of a simulation study. In: *Proceedings of the 26th Conference on Winter Simulation WSC ’94*, pp 215–220.

Baracaldo, N., & Joshi, J. (2012). A trust-and-risk aware RBAC framework: tackling insider threat. In: *Proceedings of the 17th ACM Symposium on Access Control Models and Technologies - SACMAT ’12*, pp 167–176.

Benini, A. P., Chataigner, N., Noumri, N., Parham, J., Sweeney, J., & Tax, L. (2017). The Use of Expert Judgment in Humanitarian Analysis - Theory, Methods and Applications. In *Geneva, Assessment Capacities Project.*

Berleant, D., & Kuipers, B. J. (1997). Qualitative and quantitative simulation: bridging the gap. *Artificial Intelligence*, *95*, 215–255.

Bhattacherjee, A. (2012). Social Science Research: principles, methods, and practices. Global Text Project publisher.

Bijon, K. Z., Krishnan, R., & Sandhu, R. (2013). A framework for risk-aware role based access control. In: *2013 IEEE Conference on Communications and Network Security (CNS)*, pp 462-469.

Binmore, K., & Vulkan, N. (1999). Applying game theory to automated negotiation. *Economic Research and Electronic Networking*, 1, 1–9.

Bishop, C. M., & Tipping, M. E. (1998). A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(3), 281–293.

Black, M. (1937). Vagueness. An exercise in logical analysis. *JSTOR, The University of Chicago Press*, 4(4), 427–455.

Boc, K. (2012). Fuzzy approach to risk analysis and its advantages against the qualitative approach.

In: *Proceedings of the 12th International Conference "Reliability and Statistics in Transportation and Communication*, (12), pp 234–239.

Boström, P., Grönblom, R., Huotari, T., & Wiik, J. (2010). An approach to contract-based verification of Simulink models.

Britten, N. (1995). Qualitative interviews in medical research. *BMJ: British Medical Journal*, 311(6999), 251–264.

Britton, D., & I. Brown. (2007). A security risk measurement for the RAdAC model, Naval Postgraduate School (U.S.) Retrieved from https://apps.dtic.mil/dtic/tr/fulltext/u2/a467180.pdf

Bugiel, S., Heuser, S., & Sadeghi, A.R. (2013). Flexible and fine-grained mandatory access control on Android for diverse security and privacy policies. In: *Proceedings of the 22nd USENIX Security Symposium*, pp 131–146.

Burnett, C., Chen, L., Edwards, P., & Norman, T. J. (2014). TRAAC: Trust and risk aware access control. In: *2014 Twelfth Annual International Conference on Privacy, Security and Trust*, pp 371–378.

Castiglione, A., De Santis, A., Masucci, B., Palmieri, F., Castiglione, A., Li, J., & Huang, X. (2016). Hierarchical and shared access control. *IEEE Transactions on Information Forensics and Security*, 11(4), 850–865.

Cerezuela-Escudero, E., Rios-Navarro, A., Dominguez-Morales, J. P., Tapiador-Morales, R., Gutierrez-Galan, D., Martín-Cañal, C., & Linares-Barranco, A. (2016). Performance Evaluation of Neural Networks for Animal Behaviors Classification: Horse Gaits Case Study. In: Omatu S. et al. (eds) Distributed Computing and Artificial Intelligence, 13th International Conference. Advances in Intelligent Systems and Computing.

Chen, A., Xing, H., She, K., & Duan, G. (2016). A Dynamic Risk-Based Access Control Model for Cloud Computing. In: *2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom)*, pp 579–584.

Chen, L., Gasparini, L., Norman, T., & Liang Chen, Luca Gasparini, and T. J. N. (2013). XACML and risk-aware access control. In: *Proceedings of the 10th International Workshop on Security in Information*, pp 66–75.

Chen, P., Pankaj, C., Karger, P. A., Wagner, G. M., & Schuett, A. (2007). Fuzzy Multi-Level Security : An Experiment on Quantified Risk – Adaptive Access Control. In: *2007 IEEE Symposium on Security and Privacy(SP'07)*, pp 222–227.

Cheng, T., Wen, P., & Li, Y. (2016). Research Status of Artificial Neural Network and Its Application Assumption in Aviation. In: *2016 12th International Conference on Computational Intelligence and Security (CIS)*, pp 407–410.

Choi, D., Kim, D., & Park, S. (2015). A Framework for Context Sensitive Risk-Based Access Control in Medical Information Systems. *Computational and Mathematical Methods in Medicine*, vol. 2015, 1-9.

Cirani, S., & Picone, M. (2015). Wearable Computing for the Internet of Things. *IEEE Computer Society*, 35–41.

Cisco. (2014). The Internet of Things Reference Model. White Paper, pp 1–12.

Covington, M. J., Moyer, M. J., & M. Ahamad. (2000). Generalized role-based access control for securing future applications. In: *Proceedings of the 23rd National Information Systems Security Conference (NISSC)*, pp 40–51.

Creswell, J. W. (2003). Research Design Qualitative, Quantitative, and Mixed Methods Approaches. Second Edition edition; SAGE Publications Inc.

Da Ruan. (2000). Fuzzy Sets And Fuzzy Information Granulation Theory. In: Bejing Normal Univeristy Press.

De Myttenaere, A., Golden, B., Le Grand, B., & Rossi, F. (2015). Mean Absolute Percentage Error for regression models. *Neurocomputing*, 192, 38–48.

Demuth, H. B., & Beale, M. (1998). Neural Network Toolbox for Use with MATLAB, Users Guide. The Mathworks, Inc., Massachusetts, USA.

DiCicco-Bloom, B., & Crabtree, B. F. (2006). The qualitative research interview. *Medical Education*, 40(4), 314–321.

Diep, N. N., Hung, L. X., Zhung, Y., Lee, S., Lee, Y., & Lee, H. (2007). Enforcing Access Control Using Risk Assessment. In: *The Fourth European Conference on Universal Multiservice Networks*, pp 419–424.

Dooley, K. (2002). Simulation research methods. In: The Blackwell Companion to Organizations, Blackwell, London, pp 829–848.

Dos Santos, D., Marinho, R., Schmitt, G. R., Westphall, C. M., & Westphall, C. B. (2016). A framework and risk assessment approaches for risk-based access control in the cloud. *Journal of Network and Computer Applications*, 74, 86–97.

Dos Santos, D. R., Westphall, C. M., & Westphall, C. B. (2014). A dynamic risk-based access control architecture for cloud computing. In: 2014 IEEE Network Operations and Management Symposium (NOMS), pp 1–9.

Dubois, D., & Yager, R. R. (1992). Fuzzy Set Connectives as Combination of Belief Structures. *Information Sciences*, 66, 245–275.

Eldabi, T., Irani, Z., Paul, R. J., Love, P. (2002). Quantitative and qualitative decision-making methods in simulation modelling. *Management Decision*, 40(1), 64–73.

Elkhodr, M., Shahrestani, S., & Cheung, H. (2013a). A contextual-adaptive Location Disclosure Agent for general devices in the Internet of Things. In: *1st IEEE International Workshop on Machine to Machine Communications Interfaces and Platforms*, pp 848–855.

Elkhodr, M., Shahrestani, S., & Cheung, H. (2013b). The Internet of Things: Vision & challenges. In: *Proceedings of IEEE 2013 Tencon - Spring Conference*, pp 218–222.

Elky, S. (2006). An Introduction to Information System Risk Management. In SANS Institute.

Ellah, A. R. A., Yahya, A., & Essai, M. H. (2015). Comparison of Different Backpropagation Training Algorithms Using Robust M- Estimators Performance Functions. In: *2015 Tenth International Conference on Computer Engineering & Systems (ICCES)*, pp 384–388.

Evans, D. (2011). The Internet of Things - How the Next Evolution of the Internet is Changing Everything. Cisco White Paper, (April), pp 1–11.

Fabiano, N. (2018). Internet of things and blockchain: legal issues and privacy. The challenge for a privacy standard. In: *Proceedings of 2017 IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical and Social Computing, IEEE Smart Data, IThings-GreenCom-CPSCom-SmartData*, pp 727–734.

Farooq, M. U., & Waseem, M. (2015). A Review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT). *International Journal of Computer Applications*, 113(1), 1–7.

Feitosa, E. L. O. (2014). Security Information Architecture for Automation and Control Networks. In: *VIII Brazilian Symposium on Information Security and Computational Systems*, pp 17-30.

Fletcher, R., & Reeves, C. M. (1964). Function minimization by conjugate gradients. *Computer Journal*, 7, 149–154.

Gao, P., Xue, L., Lu, Q., & Dong, C. (2015). Effects of alkali and alkaline earth metals on N-containing species release during rice straw pyrolysis. *Energies*, 8(11), 13021–13032.

Garcia-Morchon, O., & Wehrle, K. (2010). Modular context-aware access control for medical sensor networks. In: *Proceeding of the 15th ACM Symposium on Access Control Models and Technologies - SACMAT '10*, pp 129–138.

Gaudel, M.-C. (2017). Formal methods for software testing. In: *2017 International Symposium on Theoretical Aspects of Software Engineering (TASE)*, pp 1–3.

Gershenfeld, N. A. (1999). When things start to think. Henry Holt and Co.

Ghorbanzadeh, O., Rostamzadeh, H., Blaschke, T., Gholaminia, K., & Aryal, J. (2018). A new GIS-based data mining technique using an adaptive neuro-fuzzy inference system (ANFIS) and k-fold cross-validation approach for land subsidence susceptibility mapping. *Natural Hazards*, 94(2), 497–517.

Gill, P., Stewart, K., Treasure, E., & Chadwick, B. (2008). Methods of data collection in qualitative research: interviews and focus groups. *British Dental Journal*, 204(6), 291–295.

Gomez, L., & Trabelsi, S. (2014). Obligation Based Access Control. In: Meersman R. et al. (eds) On the Move to Meaningful Internet Systems: OTM 2014 Workshops. OTM 2014. Lecture Notes in Computer Science, vol 8842. Springer, Berlin, Heidelberg.

Gray, A. R., & MacDonell, S. G. (1997). A comparison of techniques for developing predictive models of software metrics. *Information and Software Technology*, 39(6), 425–437.

Guest, G., Bunce, A., & Johnson, L. (2006). How Many Interviews Are Enough ? An Experiment with Data Saturation and Variability. *Family Health International*, 18(1), 23–27.

Guney, K. (2008). Concurrent neuro-fuzzy systems for resonant frequency computation of rectangular, circular, and triangular microstrip antennas. *Progress In Electromagnetics Research*, 84, 253- 277.

Guo, Z., Zhang, Z., & Li, W. (2012). Establishment of intelligent identification management platform in railway logistics system by means of the Internet of Things. *Procedia Engineering*, 29, 726-730.

Gupta, A., Christie, R., & Manjula, P. R. (2017). Scalability in Internet of Things : Features, Techniques and Research Challenges. *International Journal of Computational Intelligence Research*, 13(7), 1617–1627.

H.-J. Zimmermann. (2000). Practical Applications of Fuzzy Technologies. In The Handbooks of Fuzzy Sets. Springer Berlin Heidelberg.

Habib, K., & Leister, W. (2015). Context-Aware Authentication for the Internet of Things. In: *The Eleventh International Conference on Autonomic and Autonomous Systems Fined*, pp 134–139.

Hamdi, M., & Abie, H. (2014). Game-based adaptive security in the Internet of Things for eHealth. In: *2014 IEEE International Conference on Communications (ICC 2014)*, pp 920–925.

Haverkamp, N., & Beauducel, A. (2017). Violation of the Sphericity Assumption and Its Effect on Type-I Error Rates in Repeated Measures ANOVA and Multi-Level Linear Models (MLM). *Frontiers in Psychology*, 8, 1841.

Haykin, S. (2004). Neural Networks – A Comprehensive foundation. 2nd Ed., Pearson Education.

Hernández-Ramos, J., & Jara, A. (2013). Distributed Capability-based Access Control for the Internet of Things. *Journal of Internet Services and Information Security (JISIS)*, *3*, 1–16.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Hu, V. C. V., Ferraiolo, D. F., Kuhn, D. R., V., H., Ferraiolo, D. F., & D., K. (2006). Assessment of access control systems. In NIST Interagency Report.

Hulsebosch, R. J., Bargh, M. S., Lenzini, G., Ebben, P. W. G., & Iacob, S. M. (2007). Context Sensitive Adaptive Authentication. In: Kortuem G., Finney J., Lea R., Sundramoorthy V. (eds) Smart Sensing and Context. EuroSSC 2007. Lecture Notes in Computer Science, vol 4793. Springer, Berlin, Heidelberg.

Hulsebosch, R. J., Salden, A. H., Bargh, M. S., Ebben, P. W. G., & Reitsma, J. (2005). Context sensitive access control. In: *Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies*, pp 111–119.

Iacono, V. Lo, Symonds, P., & Brown, D. H. K. (2016). Skype as a tool for qualitative research interviews. *Sociological Research*, 21(2), 1-12.

Ibrahim, M., Elgamri, A., Babiker, S., & Mohamed, A. (2015). Internet of things based smart environmental monitoring using the Raspberry-Pi computer. In: *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, pp 159–164.

Iqbal, M. A., Olaleye, O. G., & Bayoumi, M. A. (2016). A Review on Internet of Things (IoT): Security and Privacy Requirements and the Solution Approaches. *Global Journal of Computer Science and Technology: E Network, Web & Security*, 16(7), 1-9.

Iranmanesh, S. H., Alem, S. M., & Berneti, E. M. (2009). Project Risk Assessment for Customer Relationship Management Using Adaptive Nero Fuzzy Inference System (ANFIS). In *Proceedings of 2nd International Conference Computer Science and Its Applications*, pp 23-29.

ITU. (2005). The Internet of Things. ITU Internet Report 2005, 212.

ITU. (2012). Overview of the Internet of things. Series Y: Global Information Infrastructure, Internet Protocol Aspects and next-Generation Networks - Frameworks and Functional Architecture Models, 22. Retrieved from https://www.itu.int/rec/T-REC-Y.2060-201206-I

J. Fernández, P. C., J.Muñuzuri, & J. Guadix. (2014). Dynamic Fuzzy Logic Elevator Group Control System With Relative Waiting Time Consideration. *IEEE Transactions on industrial electronics*, 61(9), 4912–4919.

Jang, J. R., Sun, C., & Mizutani, E. (1997). Neuro-Fuzzy and Soft-Computing. Prentice Hall, Upper Saddle River, NJ.

Jang, J. S. R. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man and Cybernetics*, *23*(3), 665–685.

Jason, C. (2004). Horizontal integration: Broader Access Models for Realizing Information Dominance. In MITRE Corporation, Tech. Report.

Jeff Desjardins. (2017). The Power of Smart Contracts on the Blockchain. [Online, accessed May 22, 2018] http://www.visualcapitalist.com/smart-contracts-blockchain/

Jin, X., Krishnan, R., & Sandhu, R. S. (2012). A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7371 LNCS*, 41-55.

Kalmeshwar, M., & Prasad, N. (2017). Internet Of Things: Architecture, Issues and Applications. *International Journal of Engineering Research and Applications*, 07(06), 85–88.

Kandala, S., Sandhu, R., & Bhamidipati, V. (2011). An Attribute Based Framework for Risk-Adaptive Access Control Models. In: *Proc. of the 6th International Conference on Availability, Reliability and Security*, pp 236–241.

Kar, S., Das, S., & Ghosh, P. K. (2014). Applications of neuro fuzzy systems: A brief review and future outline. *Applied Soft Computing Journal*, *15*, 243–259.

Keller, J. M., Liu, D., & Fogel, D. B. (2016). Fuzzy Relations and Fuzzy Logic Inference. John Wiley & Sons, Inc.

Khambhammettu, H., Boulares, S., Adi, K., & Logrippo, L. (2013). A framework for risk assessment in access control systems. *Computers & Security*, 39, 86–103.

Khatoun, R., & Zeadally, S. (2017). Cybersecurity and privacy solutions in smart cities. *IEEE Communications Magazine*, 55(3), 51–59.

Kim, J., Ryu, M., & Cha, S. (2014). A Traffic Aware Routing Protocol for Congestion Avoidance in Content-Centric Network. *International Journal of Multimedia and Ubiquitous Engineering*, 9(9), 69–80.

Kitchin, R., & Dodge, M. (2017). The (In)Security of Smart Cities: Vulnerabilities, Risks, Mitigation, and Prevention. *Journal of Urban Technology*, 1–19.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1), 59–69.

Konaté, A. A., Pan, H., Khan, N., & Yang, J. H. (2015). Generalized regression and feed-forward back propagation neural networks in modelling porosity from geophysical well logs. *Journal of Petroleum Exploration and Production Technology*, 5(2), 157–166.

Korol, T., & Korodi, A. (2011). An Evaluation of Effectiveness of Fuzzy Logic Model in Predicting the Business Bankruptcy. *Romanian Journal of Economic Forecasting*, 3, 92–107.

Kose, U. (2012). Fundamentals of Fuzzy Logic with an Easy-to-use, Interactive Fuzzy Control Application. *International Journal of Modern Engineering Research (IJMER)*, 2(3), 1198-1203.

Krishna, K. L., Silver, O., Malende, W. F., & Anuradha, K. (2017). Internet of Things application for implementation of smart agriculture system. In: *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 25(15), pp 54–59.

Kumar, A., Karnik, N. M., & Chafle, G. (2002). Context sensitivity in role-based access control. *Operating Systems Review*, *36*(3), 53–66.

Langaliya, C., & Aluvalu, R. (2015). Enhancing Cloud Security through Access Control Models : A Survey. *International Journal of Computer Applications*, 112(7), 8–12.

Lee, S., Lee, Y. W., Diep, N., Lee, Y., & Lee, H. (2007). Contextual Risk-based access control. In: *Proceedings of the 2007 International Conference on Security & Management*, pp 406–412.

Lee, Y. (2015). Technology Trends of Access Control in IoT and Requirements Analysis. In: *2015 International Conference of Information and Communication Technology Convergence*, pp 1031–1033.

Lei, Z., Brodsky, A., & Jajodia, S. (2006). Toward information sharing: Benefit and Risk Access Control (BARAC). In: *Proceedings - Seventh IEEE International Workshop on Policies for Distributed Systems and Networks*, pp 45–53.

Leloglu, E. (2017). A Review of Security Concerns in Internet of Things. *Journal of Computer and Communications*, 05(01), 121–136.

Leung, K., & Verga, S. (2007). Expert Judgement in Risk Assessment Expert Judgement in Risk Assessment. *Defence R&D Canada Centre for Operational Research & Analysis*, 321–354.

Li, Jing, Cheng, J., Shi, J., & Huang, F. (2012). Brief Introduction of Back Propagation (BP) Neural Description of BP Algorithm in Mathematics. *Advances in Computer Science and Information Engineering*, *2*, 553–558.

Li, Juan, Bai, Y., & Zaman, N. (2013). A fuzzy modeling approach for risk-based access control in eHealth cloud. In *Proceedings of 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom*, pp 17–23.

Li, Y., Sun, H., Chen, Z., Ren, J., & Luo, H. (2008). Using Trust and Risk in Access Control for Grid. In*: International Conference On Environment and Security Technology*, pp 13–16.

Liaw, C. M. (1994). A Fuzzy Controller Improving a Linear Model Following Controller for Motor Drives. *IEEE Transactions on Fuzzy Systems*, 2(3), 194–202.

Liu, C., Peng, Z., & Wu, L. (2016). Role of Time-Domain Based Access Control Model. Journal of Software Engineering and Applications, 9, 57-62.

Liu, J. K., Au, M. H., Huang, X., Lu, R., & Li, J. (2016). Fine-Grained Two-Factor Access Control for Web-Based Cloud Computing Services. *IEEE Transactions on Information Forensics and Security*, 11(3), 484–497.

Lourakis, M. I. A., & Argyros, A. A. (2005). Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment?. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp 1526–1531.

Luo, J., Ni, X., & Yong, J. (2009). A trust degree based access control in grid environments.

Information Sciences, *179*(15), 2618–2628.

Maheshwari, N., & Dagale, H. (2018). Secure communication and firewall architecture for IoT applications. In: *2018 10th International Conference on Communication Systems and Networks (COMSNETS 2018)*, January, pp 328–335.

Maksimovic, M., Vujovic, V., & Kosmajac, D. (2013). Fuzzy rule reduction influence on system's accuracy. In: *2013 21st Telecommunications Forum Telfor (TELFOR)*, pp 920–923.

Mamdani, E. H., & Assilian, S. (1975). An experimental in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1), 1–13.

Maple, C. (2017). Security and privacy in the internet of things. *Journal of Cyber Policy*, 2(2), 155-184.

Mathworks, C. (2016). Fuzzy Logic Toolbox User' s Guide. Natick, Massachusetts: The MathWorks, Inc.

Maw, H. A., Xiao, H., & Christianson, B. (2012). An adaptive access control model with privileges overriding and behaviour monitoring in wireless sensor networks. In: *Proceedings of the 8h ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pp 81-86.

McGraw, R. (2009). Risk-Adaptable Access Control (RAdAC): Access Control and the Information Sharing Problem. In: *Proceedings of NIST & NSA Privilege Management Workshop*.

Metoui, N. (2018). Privacy-Aware Risk-Based Access Control Systems. PhD thesis, University of Trento.

Mohanta, B. K., Panda, S. S., & Jena, D. (2018). An Overview of Smart Contract and Use Cases in Blockchain Technology. In: *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp 1–4.

Moller, M. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6, 525–533.

Moore, D., & McCabe, D. (1993). Introduction to the practice of statistics. New York: Freeman.

Morabito, V. (2017). Smart Contracts and Licensing. In *Business Innovation Through Blockchain*, pp 101–124.

Musaddiq, A., Zikria, Y. Bin, Hahm, O., Yu, H., Bashir, A. K., & Kim, S. W. (2018). A Survey on Resource Management in IoT Operating Systems. *IEEE Access*, 6, 8459–8482.

N.Mahalle, P., Anggorojati, B., Prasad, N. R., & Prasad, R. (2013). Identity Authentication and Capability Based Access Control (IACAC) for the Internet of Things. *Journal of Cyber Security and Mobility*, 1, 309–348.

Naidu, D. S., & Sun, C. T. (1997). Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. *IEEE Transactions on automatic control*, 1520 - 1533.

Namitha, S., Gopalan, S., Sanjay, H. N., & Chandrashekaran, K. (2015). Risk Based Access Control In Cloud Computing. In: *International Conference on Green Computing and Internet of Things (ICGCloT)*, pp 1502–1505.

Natarajan, H., Krause, S. K., & Gradstein, H. L. (2017). Distributed Ledger Technology (DLT) and Blockchain. FinTech Note, Washington, D.C, World Bank Group.

Nawi, N. M., Ransing, R. S., Salleh, M. N. M., Ghazali, R., & Hamid, N. A. (2010). An Improved Back Propagation Neural Network Algorithm on Classification Problems. In: Database Theory and Application, Bio-Science and Bio-Technology, Vol 118. Springer, Berlin, Heidelberg.

Negnevitsky, M. (2010). Artificial Intelligence (Second). Pearson Education Limited 2002.

Ni, Q., Bertino, E., & Lobo, J. (2010). Risk-based access control systems built on fuzzy inferences. In*: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, New York, USA*, pp 250–260.

OASIS. (2003). eXtensible Access Control Markup Language (XACML). *OASIS Standard*,

(January), pp 1–154.

Okut, H. (2016). Bayesian Regularized Neural Networks for Small n Big p Data. *Artificial Neural Networks - Models and Applications*, pp. 28–48.

Omar, A. S., Waweru, M., & Rimiru, R. (2015). Application of Fuzzy Logic in Qualitative Performance Measurement of Supply Chain Management. *International Journal of Information and Communication Technology Research*, 5(6), 42–55.

Otwayl, H., & Winterfeldt, D. Von. (1992). Expert Judgment in Risk Analysis and Management : Process, Context, and Pitfalls. *Risk Analysis*, 12(I), 82–93.

P. Guillemin and P. Friess. (2009). Internet of Things Strategic Research Roadmap. European Commission Information Society and Media, Luxembourg.

P.J. Werbos. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behaviour Sciences. Harward University, Cambridge.

Pang, W., & Coghill, G. M. (2015). Qualitative, semi-quantitative, and quantitative simulation of the osmoregulation system in yeast. *BioSystems*, 131, 40–50.

Patel, K. K., & Patel, S. M. (2016). Internet of Things-IoT : Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. *International Journal of Engineering Science and Computing*, 6(5), 6122–6131.

Pătru, I. I., Carabaş, M., Bărbulescu, M., & Gheorghe, L. (2016). Smart home IoT system. *Proceedings of 15th International Conference of Networking in Education and Research*, pp 365–370.

Peleg, M., Beimel, D., Dori, D., & Denekamp, Y. (2008). Situation-Based Access Control: Privacy management via modeling of patient data access scenarios. *Journal of Biomedical Informatics*, 41(6), 1028–1040.

Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *IEEE Communications Surveys and Tutorials*, *16*(1), 414–454.

Peter Houlis. (2018). The history and future of access control credentials. [Online, Accessed March 9, 2019] https://www.ifsecglobal.com/global/history-future-access-control-credentials/

Phinyomark, A., Thongpanj, S., Hu, H., & Pornchai, Phukpattaranont Chusak, L. (2012). The Usefulness of Mean and Median Frequencies in Electromyography Analysis. In *Computational Intelligence in Electromyography Analysis - A Perspective on Current Applications and Future Challenges*, pp 196–218

Pluess, D., Groso, A., & Meyer, T. (2013). Expert Judegement in Risk Analysis: A Strategy to Overcome Uncertainities. *Chemical Engineering Transactions*, 31, 307–312.

Pramanik, N., & Panda, R. K. (2009). Application of neural network and adaptive neuro-fuzzy inference systems for river flow prediction. *Hydrological Sciences Journal*, *54*(2), 247–260.

Quijano-Sánchez, L., Bridge, D., Díaz-Agudo, B., & Recio-García, J. A. (2012). A Case-Based Solution to the Cold-Start Problem in Group Recommenders. In: Agudo B.D., Watson I. (eds) Case-Based Reasoning Research and Development. ICCBR 2012. Lecture Notes in Computer Science, vol 7466. Springer, Berlin, Heidelberg.

Radionovs, A., & Uzhga-rebrov (2014). Application of Fuzzy Logic for Risk Assessment. *Information Technology and Management Science*, 50–54.

Rahbari, O., Mayet, C., Omar, N., Mierlo, J. Van, Rahbari, O., Mayet, C., Van Mierlo, J. (2018). Battery Aging Prediction Using Input-Time-Delayed Based on an Adaptive Neuro-Fuzzy Inference System and a Group Method of Data Handling Techniques. *Applied Sciences*, 8(8), 1301.

Rajbhandari, L., & Snekkenes, E. A. (2011). Using game theory to analyze risk to privacy: An initial insight. In: Privacy and Identity Management for Life, Springer Berlin Heidelberg, pp 41–51.

Ramesh, C. S., Jain, V. K. S., Keshavamurthy, R., Khan, Z. A., & Hadfield, M. (2013). Prediction

of slurry erosive wear behaviour of Al6061 alloy using a fuzzy logic approach. *WIT Transactions on Engineering Sciences*, 78, 109–119.

Ramona, S. E. (2011). Advantages and Disadvantages of Quantitative and Qualitative Information Risk Approaches. *Chinese Business Review*, 10(12), 1106–1110.

Rezaei, K., Hosseini, R., & Mazinani, M. (2014). A Fuzzy Inference System for Assessment of the Severity of the Peptic Ulcers. *Computer Science & Information Technology*, 263–271.

Roberto, M., Abyi, B., & Domenico, R. (2015). Towards a definition of the Internet of Things (IoT). *IEEE Internet of Things*, 1–86.

Rochette, S., Lobry, J., Lepage, M., & Boët, P. (2009). Dealing with uncertainty in qualitative models with a semi-quantitative approach based on simulations. Application to the Gironde estuarine food web. *Ecological Modelling*, 220, 122–132.

Rogers, Y., Sharp, H., & Preece, J. (2011). Interaction design: beyond human-computer interaction. John Wiley & Sons, Ltd.

Roman, R., Zhou, J., & Lopez, J. (2013). On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10), 2266–2279.

Ross, T. J. (2010). Fuzzy Logic with Engineering Applications. John Wiley & Sons, Ltd.

Saduf, & Wani, M. A. (2013). Comparative Study of Back Propagation Learning Algorithms for Neural Networks. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(12), 1151–1156.

Saini, L. M. (2008). Peak load forecasting using Bayesian regularization, Resilient and adaptive backpropagation learning based artificial neural networks. *Electric Power Systems Research*, 78(7), 1302–1310.

Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. *Computer*, 29(2), 38–47.

Seguí, X., Pujolasus, E., Betrò, S., Àgueda, A., Casal, J., Ocampo-Duque, W., Darbra, R. M. (2013). Fuzzy model for risk assessment of persistent organic pollutants in aquatic ecosystems. *Environmental Pollution*, 178, 23–32.

Shaf, J., Angelov, P., & Umair, M. (2016). Prediction of the Attention Area in Ambient Intelligence Tasks. In: Sgurev V., Yager R., Kacprzyk J., Jotsov V. (eds) Innovative Issues in Intelligent Systems. Studies in Computational Intelligence, vol 623. Springer, Cham.

Shaikh, R. A., Adi, K., & Logrippo, L. (2012). Dynamic risk-based decision methods for access control systems. *Computers and Security*, 31(4), 447–464.

Shanbhag, R., & Shankarmani, R. (2015). Architecture for Internet of Things to minimize human intervention. In: *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI 2015)*, pp 2348–2353.

Shang, K., & Hossen, Z. (2013). Applying Fuzzy Logic to Risk Assessment and Decision-Making. *Casualty Actuarial Society, Canadian Institute of Actuaries, Society of Actuaries*, pp 1–59.

Shapiro, A., & Koissi, M. (2015). Risk Assessment Applications of Fuzzy Logic. Casualty Actuarial Society, Canadian Institute of Actuaries.

Sharma, M., Bai, Y., Chung, S., & Dai, L. (2012). Using risk in access control for cloud-assisted ehealth. In: *High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS)*, pp 1047–1052.

Shen, J., Zhou, T., Chen, X., Li, J., & Susilo, W. (2018). Anonymous and Traceable Group Data Sharing in Cloud Computing. *IEEE Transactions on Information Forensics and Security*, 13(4), 912–925.

Shuzhao, A., & Zhaohui, W. (2014). Based on the Part of Routing Information Congestion Modeling Research with the Large-Scale Network. *International Journal of Future Generation Communication and Networking*, 7(2), 173–182.

Singh, H., Gupta, M. M., Meitzler, T., Hou, Z.-G., Garg, K. K., Solo, A. M. G., & Zadeh, L. A. (2013). Real-Life Applications of Fuzzy Logic. *Advances in Fuzzy* Systems, 1-3.

Singh, V., Rana, R. K., & Singhal, R. (2013). Analysis of repeated measurement data in the clinical trials. *Journal of Ayurveda and Integrative Medicine*, 4(2), 77–81.

Singhala, P., Shah, D. N., & Patel, B. (2014). Temperature Control using Fuzzy Logic. *International Journal of Instrumentation and Control Systems (IJICS)*, 4(1), 1–10.

Smart Contracts Alliance. (2016). Smart Contracts: 12 Use Cases for Business & Beyond. Chamber of Digital Commerce, (April), 56.

Sorich, M. J., Miners, J. O., McKinnon, R. A., Winkler, D. A., Burden, F. R., & Smith, P. A. (2003). Comparison of Linear and Nonlinear Classification Algorithms for the Prediction of Drug and Chemical Metabolism by Human UDP-Glucuronosyltransferase Isoforms. *Journal of Chemical Information and Computer Sciences*, 43(6), 2019–2024.

Stallings, W. (2015). The Internet of Things: Network and Security Architecture. *The Internet Protocol Journal*, 18(4), 2–24.

Statista. (2018). Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025. [Online, Accessed October 15, 2018] https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

Stoneburner, G., Goguen, A., & Feringa, A. (2002). Risk Management Guide for Information Technology Systems. *NIST Special Publication Sp*, 19(30).

Suhendra, V. (2011). A Survey on Access Control Deployment. *Communications in Computer and Information Science*, 11–20.

Sultan, K., Ruhi, U., & Lakhani, R. (2018). Conceptualizing Blockchain: Characteristics & Applications. In: *11th IADIS International Conference Information Systems*, pp 49–57.

Suo, H., Wan, J., Zou, C., & Liu, J. (2012). Security in the internet of things: A review. In *International Conference on Computer Science and Electronics Engineering (CCSEE 2012)*, *3*, pp 648–651.

Suparta, W., & Alhasa, K. M. (2016). Adaptive Neuro-Fuzzy Interference System. In Modeling of Tropospheric Delays Using ANFIS, Springer International Publishing Inc, pp 5–19.

Szabo, N. (1994). Smart Contracts. [Online, Accessed May 22, 2018] http://szabo.best.vwh.net/smart.contracts.html

Takagi, T., & Sugeno, M. (1985). Fuzzy Identification of Systems and Its Applications to Modeling and Control. *IEEE Transactions on Systems, Man and Cybernetics*, 15(1), 116–132.

Taylor, G. R. (2005). Integrating Quantitative and Qualitative Methods in Research. University Press of America, 2nd revise edition.

Tessmer, M. (1993). Planning and conducting formative evaluations. Psychology Press.

Tiwari, S., Babbar, R., & Kaur, G. (2018). Performance Evaluation of Two ANFIS Models for Predicting Water Quality Index of River Satluj (India). *Advances in Civil Engineering*, Vol. 2018, 1-10.

Tóth-laufer, E., & Takács, M. (2012). The Effect of Aggregation and Defuzzification Method Selection on the Risk Level Calculation. In: *10th IEEE Jubilee International Symposium on Applied Machine Intelligence and Informatics*, pp 131–136.

Turisová, R., Mihok, J., & Kádárová, J. (2012). Verification of the Risk Assessment Model through An Expert Judgment. *Kvalita Inovacia Prosperita/ Quality Innovation Prosperity*, 37–48.

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: heuristics and biases. *Science*, 185(4157), 1124–1131.

Tyson Brooks, Caicedo, C., & Park, J. S. (2012). Security Vulnerability Analysis in Virtualized Computing Environments. *International Journal of Intelligent Computing Research (IJICR)*, 3(4), 263–277.

Verma, S., Singh, M., & Kumar, S. (2012). Comparative analysis of Role Base and Attribute Base Access Control Model in Semantic Web General Terms Access Control Model in Semantic Web. *International Journal of Computer Applications*, Vol. 46, 54-66

Vieira, J., Dias, F. M., & Mota, A. (2004). Neuro-Fuzzy Systems : A Survey. In: *5th WSEAS NNA International Conference on Neural Networks and Applications*, pp 1–6.

Viharos, Z. J., & Kis, K. B. (2015). Survey on Neuro-Fuzzy systems and their applications in technical diagnostics and measurement. *Journal of the International Measurement Confederation*, 67, 126–136.

Vijayakumar, H., Jakka, G., Rueda, S., Schiffman, J., & Jaeger, T. (2012). Integrity walls: Finding attack surfaces from mandatory access control policies. In: *7th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2012)*, pp 75–76.

Wan, K., & Alagar, V. (2013). Integrating context-awareness and trustworthiness in IoT descriptions. In: *Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing (GreenCom-IThings-CPSCom 2013)*, pp 1168–1174.

Wang, C. (2015). A Study of Membership Functions on Mamdani- Type Fuzzy Inference System for Industrial. PhD thesis, Lehigh University.

Wang, Q., & Jin, H. (2011). Quantified risk-adaptive access control for patient privacy protection in health information systems. In: *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security (ASIACCS '11)*, pp 406–410.

Wang, S., Fan, C., Hsu, C. H., Sun, Q., & Yang, F. (2016). A Vertical Handoff Method via Self-Selection Decision Tree for Internet of Vehicles. *IEEE Systems Journal*, *10*(3), 1183–1192.

Wang, Y. M., & Elhag, T. M. S. (2008). An adaptive neuro-fuzzy inference system for bridge risk assessment. *Expert Systems with Applications*, 34(4), 3099–3106.

Waqas Aman. (2013). Modeling Adaptive Security in IoT Driven eHealth. In: *Norwegian Information Security Conference (NISK 2013)*, pp 61–69.

Watanabe, H., Fujimura, S., Nakadaira, A., Miyazaki, Y., Akutsu, A., & Kishigami, J. (2016). Blockchain contract: Securing a blockchain applied to smart contracts. In: *2016 IEEE International Conference on Consumer Electronics (ICCE)*, pp 467–468.

White, H. (1992). Artificial neural networks : approximation and learning theory. Cambridge, USA : Blackwell.

Wu, Y., Zhang, B., Lu, J., & Du, K.-L. (2011). Fuzzy Logic and Neuro-fuzzy Systems: A Systematic Introduction. *International Journal of Artificial Intelligence and Expert Systems*, 2(2), 47–80.

Xu, D., & Zhang, Y. (2014). Specification and analysis of attribute-based access control policies: An overview.In: *Proceedings of 8th International Conference on Software Security and Reliability (SERE-C 2014)*, pp 41–49.

Xu, L. Da, He, W., & Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, 10(4), 2233–2243.

Ye, N., Zhu, Y., Wang, R. C., Malekian, R., & Lin, Q. M. (2014). An efficient authentication and access control scheme for perception layer of internet of things. *Applied Mathematics and Information Sciences*, 8(4), 1617–1624.

Yin, J., Tang, C., Zhang, X., & McIntosh, M. (2006). On estimating the security risks of composite software services. *In First Program Analysis for Security and Safety Workshop Discussion (PASSWORD 2006)*, pp 1-10.

Yu, Y., Kaiya, H., Yoshioka, N., Hu, Z., & Washizaki, H. (2016). Goal Modelling for Security Problem Matching and Pattern Enforcement. *International Journal of Secure Software Engineering*, 8(3), 42–57.

Zadeh, L. A. (1965). Fuzzy Sets. *International Journal of Information Control*, 8, 338–353.

Zanchettin, C., Mimku, L. ., & Ludermir, T. b. (2010). Design of Experiments in Neuro-Fuzzy Systems. *International Journal of Computational Intelligence and Applications*, 09(02), 137-152.

Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, 1(1), 22–32.

Zhang, Z.-K., Wang, M. C. Y. C. C.-W., Hsu, C.-W., & Chong-Kuan Chen. (2014). IoT Security: Ongoing Challenges and Research Opportunities. In: *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, pp 230–234.

Zhou, L., Varadharajan, V., & Hitchens, M. (2013). Achieving Secure Role-Based Access Control on Encrypted Data in Cloud Storage. *IEEE Transactions on Information Forensics and Security*, 8(12), 1947–1960.

Zhu, H., & Jin, R. (2007). A Practical Mandatory Access Control Model for XML Databases A Practical Mandatory Access Control Model for XML Databases. In: *2nd International Conference: Scalable Information Systems*, pp 1–4.

Zhu, Z., & Xu, R. (2008). A Context-Aware Access Control Model for Pervasive Computing in Enterprise Environments. In: *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, pp 1–6.

Zia, T., Liu, P., & Han, W. (2017). Application-Specific Digital Forensics Investigative Model in Internet of Things (IoT). In: *Proceedings of the 12th International Conference on Availability, Reliability and Security (ARES '17)*, pp 1–7.

Appendix A Validation of Proposed Model and Fuzzy Rules

This appendix contains all material of the expert interview that was carried out to validate the proposed risk-based access control model, create fuzzy rules and determine values of access decision bands. It involves contacting experts, information sheet of the interview, and consent form. This followed by presenting interview questions.

A.1 Contacting Experts

Dear xxx.

My name is Hany Atlam; I am a PhD student in Computer Science at the University of Southampton. I am working on developing an adaptive risk-based access control model for the Internet of Things (IoT). I am writing to invite you to participate in an expert interview to validate the proposed risk-based model and confirm a set of fuzzy rules based on your experience and knowledge of security and IoT applications. The interview will be via Skype and it will be about 40 minutes and it can be arranged at your convenient time. The results will help in building a dynamic risk-based access control model that can adapt to unpredicted situations. I sincerely hope that you will consider participating in this interview. I will be contacting you in the near future to confirm your interest in being interviewed. Please feel free to contact me with any questions. Please find the attached information sheet about the interview for your reference.

Sincerely,

Participant Information Sheet

Ethics reference number: **ERGO/FPSE/25091**	Version: **1**	Date: **27/06/2017**
Study Title: **Developing an Adaptive Model for Security Risk-based Access Control in the Internet of Things**		
Investigator: **Hany Atlam**		

Please read this information carefully before deciding to take part in this research. If you are happy to participate, you will be asked to sign a consent form. Your participation is completely voluntary.

What is the research about?

This research is for my PhD. I have created a risk-based access control model for the Internet of Things (IoT) system. Risk estimation process is one of the major tasks to implement this model. I will use the fuzzy logic system to estimate the risk value associated with each access request. By the means of this interview, I wish to validate the proposed risk-based model and confirm fuzzy rules that will be used to implement the risk estimation process.

Why have I been chosen?

You have been chosen because of your knowledge and experience in cyber security and IoT applications.

What will happen to me if I take part?

If you decided to take part in this research, you will spend about 40 minutes for completing the questionnaire or answering the questions in an interview format.

Are there any benefits in my taking part?

There are no benefits for you to take part in this work. Your participation is completely voluntary

Are there any risks involved?

No risks are involved in this research.

Will my data be confidential?

All data collected will be anonymous and will be used only for the purposes of research. Data will be held on a password-protected computer so nobody except the researcher has access to it. The collection of data complies with the University of Southampton policy under the data protection Act.

What happens if I change my mind?

You may withdraw at any time and for any reason. You may access, change, or withdraw your data at any time and for any reason prior to its destruction.

What happens if something goes wrong? Should you have any concern or complaint, contact me (Hany Atlam, hfa1g15@soton.ac.uk), otherwise please contact my supervisor prof. Gary Wills (gbw@ecs.soton.ac.uk). Otherwise please contact the FPSE Office (ergopse@soton.ac.uk) or any other authoritative body such as the Research Integrity & Governance Team (rgoinfo@soton.ac.uk).

Consent Form

Ethics reference number: **ERGO/FPSE/25091**	Version: **1**	Date: **27/06/2017**
Study Title: **Developing an Adaptive Model for Security Risk-based Access Control in the Internet of Things**		
Investigator: **Hany Atlam**		

Please initial each statement if you agree:

I have read and understood the Participant Information (version 1 dated 12/01/2017) and have had the opportunity to ask questions about the study.	
I agree to take part in this study.	
I understand my participation is voluntary and I may withdraw at any time and for any reason.	

Data Protection

I understand that information collected during my participation in this study is completely anonymous / will be stored on a password protected computer/secure University server and that this information will only be used in accordance with the Data Protection Act (1998). The DPA (1998) requires data to be processed fairly and lawfully in accordance with the rights of participants and protected by appropriate security. In addition, the DPA (1998) makes provision for an appropriate authority, such as the Police, to access data held by the study for the purpose of…

Name of participant (print name)……

Signature of participant……

Date……

A.2 Interview Questions

The main purpose of this research is to develop a risk-based access control model for the Internet of Things (IoT) applications. This model has the ability to permit or deny access requests dynamically based on the estimated risk value of each access request. One of the major tasks of implementing our model is the risk estimation process. We decided to utilize the fuzzy logic system with expert judgment as the appropriate risk estimation technique. One of the essential steps to implement fuzzy logic is to set the appropriate fuzzy rules. Your response and expertise will help us to validate the proposed model, confirm fuzzy rules and decide acceptable risk values to provide the access decision. All provided information will be used for research purposes only. Your participation is greatly appreciated.

Part1: Background Questions

1.1 What is your level of education?

- ☐ Bachelor degree
- ☐ Master degree
- ☐ Doctoral degree
- ☐ Others, please specify……

1.2 Which of the following describe your job role?

- ☐ Security Administrator
- ☐ Security Analyst
- ☐ Security Specialist
- ☐ Senior Cybersecurity Engineer
- ☐ Security researcher
- ☐ Others, please specify……

1.3 Which of the following IoT applications are you familiar with?

- ☐ Connected industry
- ☐ Smart city
- ☐ Smart energy
- ☐ Connected car

- ☐ Smart home
- ☐ Smart agriculture
- ☐ Healthcare
- ☐ Smart retail
- ☐ Smart supply chain
- ☐ Others, please specify……

1.4 How long have you been working in the field of cyber security?

- ☐ Less than 2 years
- ☐ 2 – 5 years
- ☐ 6 – 10 years
- ☐ More than 10 years

1.5 Do you have background knowledge about the fuzzy logic system?

- ☐ Yes
- ☐ No
- ☐ Other, please specify ……

Part 2: Validation of Proposed Model

We proposed a dynamic risk-based access control model. This model uses real-time and contextual features associated with the user while making the access request with resource sensitivity, action severity, and risk history as inputs/risk factors to estimate the security risk value associated with each access request. Then, the estimated risk value is compared against risk policies to provide the access decision, as shown in Figure A.1.

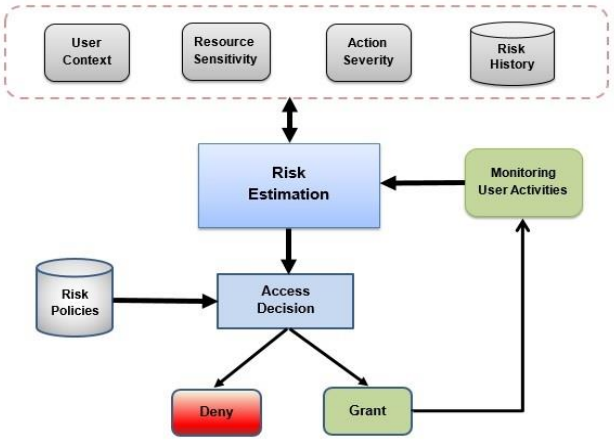


Figure A.1: Proposed risk-based access control model

We provide a summary of components of the proposed risk-based model as follow:

- **User Context**: It represents the environmental attributes collected from the IoT environment at the time of making the access request like user location, current time, user profile, etc.
- **Resource Sensitivity**: It represents how valuable the resource/data is to the owner or to the service provider. Each resource or data have a different sensitivity level. So, the higher the data sensitivity, the higher the risk associated with the data.
- **Action Severity**: It represents the consequences of a certain action on a particular resource in terms of security requirements of confidentiality, integrity and availability. For example, the risk of a view operation is lower than the risk of a delete operation.
- **Risk History**: It represents user previous risk values. It used to identify good and bad authorized users and predict the user future behaviour.
- **Risk Policies**: They are mainly used by the risk estimation module to make access decisions. These policies are created by the resource owner or security system administrator to identify terms and conditions of granting or denying access to a particular resource.
- **Risk Estimation Module**: It is responsible for taking the input features/ risk factors to quantify the risk value associated with each access request.

Please, use your knowledge and experience to answer these questions.

2.1 What is your feedback about the proposed risk-based access control model?

2.2 Are the proposed risk factors appropriate for different IoT applications?

2.3 In term of importance for IoT applications, what is the ranking of the proposed four risk factors?

Part 3: Validation of Fuzzy Rules

One of the major tasks to build a risk-based access control model is the risk estimation process. We decided to use the fuzzy logic system to estimate the risk value. However, to build an efficient fuzzy model, fuzzy rules should be specified by domain experts. We decided to use three fuzzy sets for each risk factors/input and five fuzzy sets for the output risk, as shown in Table A.1.

Table A.1: Linguistic variables of input and output

Linguistic value	Notation	Range
Input variable: User Context (UC)		
Low	L	0 - 0.4
Moderate	M	0.3 - 0.7
High	H	0.6 - 1
Input variable: Resource Sensitivity (RS)		
Not Sensitive	NS	0 - 0.35
Sensitive	S	0.2 - 0.5
Highly Sensitive	HS	0.45 - 1
Input variable: Action Severity (AS)		
Low	L	0 - 0.4
Moderate	M	0.35 - 0.7
High	H	0.6 - 1
Input variable: Risk History (RH)		
Low	L	0 - 0.4
Moderate	M	0.3 - 0.7
High	H	0.6 - 1
Output variable: Risk (R)		
Negligible	N	0 - 0.3
Low	L	0.1 - 0.4
Moderate	M	0.2 - 0.6
High	H	0.4 - 0.8
Unacceptable High	UH	0.7 - 1

Fuzzy rules are built as IF-THEN statements to describe how the output risk varies as a function of the four risk factors. For example, if (action severity is Low & resource sensitivity is Not Sensitive & user context is Low & risk history is Low) then (the output risk will be Negligible). Fuzzy rules were built using information collected from the literature with researcher experience. The relation between action severity and resource sensitivity that is shown in Figure A.2 was utilized with the following logical rules:

- If the risk history increased, the output risk will not decrease.
- If the resource sensitivity increased, the output risk will not decrease.
- If the Action severity increased, the output risk will not decrease.
- If any two inputs are high, the lowest output will be high.

- If the resource sensitivity is Highly Sensitive (HS) or Sensitive (S), the output risk cannot be Negligible (N).

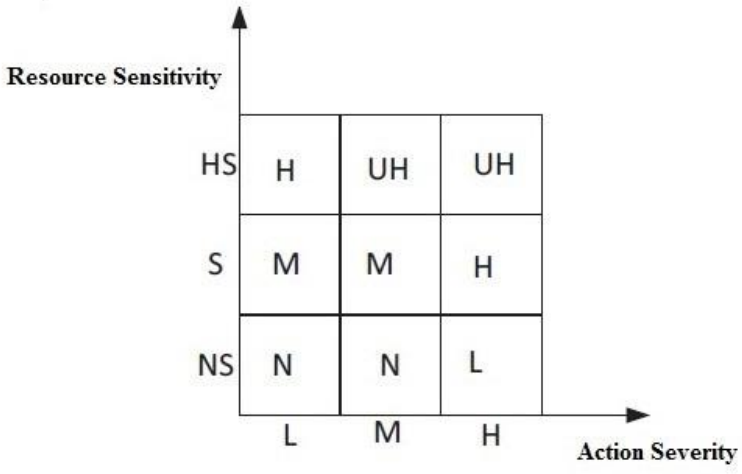


Figure A.2: Risk value regarding resource sensitivity and action severity

Please, confirm these fuzzy rules using your knowledge and experience.

Input Notations: L: Low; **M:** Moderate; **H:** High; **NS:** Not Sensitive; **S:** Sensitive; **HS:** Highly Sensitive

Output Notations: N: Negligible; **L:** Low; **M:** Moderate; **H:** High; **UH:** Unacceptable High

Table A.2: Fuzzy rules when output is Negligible

Rule No	Risk Factors				Output Risk	Expert Validation		
	Action Severity	Resource Sensitivity	User context	Risk History		Yes	No	Output if No
1	L	NS	L	L	N			
2	M	NS	L	L	N			
3	H	NS	L	L	N			
4	L	NS	M	L	N			
5	M	NS	M	L	N			
6	H	NS	M	L	N			
7	L	NS	H	L	N			
8	L	S	L	L	N			
9	M	S	L	L	N			
10	L	NS	L	M	N			
11	M	NS	L	M	N			
12	L	NS	M	M	N			
13	M	NS	M	M	N			

Input Notations: L: Low; **M:** Moderate; **H:** High; **NS:** Not Sensitive; **S:** Sensitive; **HS:** Highly Sensitive

Output Notations: N: Negligible; **L:** Low; **M:** Moderate; **H:** High; **UH:** Unacceptable High

Table A.3: Fuzzy rules when output is Low

Rule No	Risk Factors				Output Risk	Expert Validation		
	Action Severity	Resource Sensitivity	User context	Risk History		Yes	No	Output if No
14	H	S	L	L	L			
15	L	HS	L	L	L			
16	M	HS	L	L	L			
17	H	HS	L	L	L			
18	L	S	M	L	L			
19	M	S	M	L	L			
20	M	NS	H	L	L			
21	H	NS	L	M	L			
22	L	S	L	M	L			
23	H	NS	M	M	L			
24	L	NS	H	M	L			
25	L	NS	L	H	L			
26	M	NS	L	H	L			
27	L	NS	M	H	L			
28	M	NS	M	H	L			

Table A.4: Fuzzy rules when output is Moderate

Rule No	Risk Factors				Output Risk	Expert Validation		
	Action Severity	Resource Sensitivity	User context	Risk History		Yes	No	Output if No
29	H	S	M	L	M			
30	L	HS	M	L	M			
31	M	HS	M	L	M			
32	H	HS	M	L	M			
33	H	NS	H	L	M			
34	L	S	H	L	M			
35	M	S	H	L	M			
36	M	S	L	M	M			
37	H	S	L	M	M			
38	L	HS	L	M	M			
39	M	HS	L	M	M			
40	L	S	M	M	M			
41	M	S	M	M	M			
42	M	NS	H	M	M			
43	H	NS	H	M	M			
44	H	NS	L	H	M			
45	L	S	L	H	M			
46	H	NS	M	H	M			
47	L	NS	H	H	M			

Input Notations: L: Low; **M:** Moderate; **H:** High; **NS:** Not Sensitive; **S:** Sensitive; **HS:** Highly Sensitive

Output Notations: N: Negligible; **L:** Low; **M:** Moderate; **H:** High; **UH:** Unacceptable High

Table A.5: Fuzzy rules when output is High

Rule No	Risk Factors				Output Risk	Expert Validation		
	Action Severity	Resource Sensitivity	User context	Risk History		Yes	No	Output if No
48	H	S	H	L	H			
49	L	HS	H	L	H			
50	H	HS	L	M	H			
51	H	S	M	M	H			
52	L	HS	M	M	H			
53	M	HS	M	M	H			
54	H	HS	M	M	H			
55	L	S	H	M	H			
56	M	S	H	M	H			
57	L	HS	H	M	H			
58	M	S	L	H	H			
59	L	S	M	H	H			
60	M	S	M	H	H			

Table A.6: Fuzzy rules when output is Unacceptable High

Rule No	Risk Factors				Output Risk	Expert Validation		
	Action Severity	Resource Sensitivity	User context	Risk History		Yes	No	Output if No
61	M	HS	H	L	UH			
62	H	HS	H	L	UH			
63	H	S	H	M	UH			
64	M	HS	H	M	UH			
65	H	HS	H	M	UH			
66	H	S	L	H	UH			
67	L	HS	L	H	UH			
68	M	HS	L	H	UH			
69	H	HS	L	H	UH			
70	H	S	M	H	UH			
71	L	HS	M	H	UH			
72	M	HS	M	H	UH			
73	H	HS	M	H	UH			
74	M	NS	H	H	UH			
75	H	NS	H	H	UH			
76	L	S	H	H	UH			
77	M	S	H	H	UH			
78	H	S	H	H	UH			
79	L	HS	H	H	UH			
80	M	HS	H	H	UH			
81	H	HS	H	H	UH			

Part 4: Acceptable Risk Bands

To determine the access decision, the estimated risk value is compared against acceptable risk values, which are specified in risk policies, to determine access decisions. This research proposed three risk decision bands, as follow:

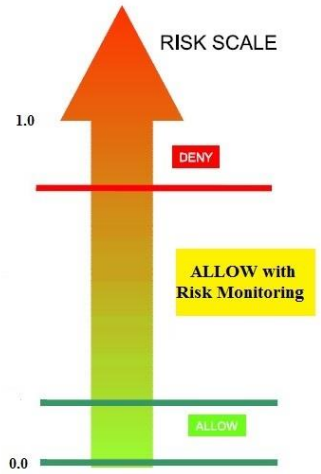


Figure A.3: Proposed access decision bands

- **Allow band**: This band is used to grant access without monitoring user activities during access sessions to keep the user's privacy.
- **Allow with Risk Monitoring band**: This band is used to grant access with monitoring all users' behaviours and activities during the access session to detect any malicious behaviour.
- **Deny band**: Due to the high-risk value associated with the user requesting the access, the access will be denied through this band.

Please use your knowledge and experience to decide the best values for each risk decision band. these risk bands. Please use **Yes** to confirm the suggested values by the researcher for each band, and **No** to suggest other values.

4.1 Do you think the range from 0.0 – 0.25 is appropriate for the allow band?

☐ Yes

☐ No, please specify ……...

4.2 Do you think the range from 0.26 – 0.7 is appropriate for the allow with risk monitoring band?

☐ Yes

☐ No, please specify ……...

4.3 Do you think the range from 0.71 – 1.0 is appropriate for the deny band?

☐ Yes

☐ No, please specify ……...

4.4 Do you suggest any other decision bands?

Appendix B Validating Fuzzy Rules of Cold Start

This appendix contains all material of the expert interview that was carried out to validate fuzzy rules of the cold start problem. Since the information sheet and consent form are the same as in Appendix A, this section will only present interview questions.

B.1 Interview Question

The main purpose of this research is to develop a risk-based access control model for IoT applications. This model has the ability to permit or deny access requests dynamically based on the estimated risk value of each access request. All provided information would be used for research purposes only. Your participation is greatly appreciated.

Part 1: Background Questions

1.1 Which of the following describe your job role?

- ☐ Security Administrator
- ☐ Security Analyst
- ☐ Security Specialist
- ☐ Senior Cybersecurity Engineer
- ☐ Security researcher
- ☐ Others, please specify……

1.2 Which of the following IoT applications are you familiar with?

- ☐ Connected industry
- ☐ Smart city
- ☐ Smart energy
- ☐ Connected car
- ☐ Smart home
- ☐ Smart agriculture
- ☐ Healthcare
- ☐ Smart retail
- ☐ Smart supply chain
- ☐ Others, please specify……

1.3 How long have you been working in the field of IoT security?

- ☐ Less than 2 years
- ☐ 2 – 5 years
- ☐ 6 – 10 years

☐ More than 10 years

1.4 Do you have background knowledge about the fuzzy logic system?

☐ Yes

☐ No

☐ Other, please specify ……

Part 2: Validation of Fuzzy Rules

The proposed risk-based access control model uses real-time user context, resource sensitivity, action severity, and risk history as inputs to estimate the security risk value for each access request, as shown in Figure B.1.

- **User Context**: it represents the environmental attributes associated with the user at the time of making the access request such as user location, current time, and user profile.
- **Resource Sensitivity**: It represents how valuable the resource/data is to the owner or to the service provider. For instance, the higher the data sensitivity, the higher the risk associated with the data.
- **Action Severity**: It represents the consequences of a certain action on a particular resource in terms of security requirements of confidentiality, integrity, and availability. For example, the risk of a view operation is lower than the risk of a delete operation.
- **Risk History**: It represents user previous risk values. It used to identify good and bad authorized users and predict the user future behaviour.

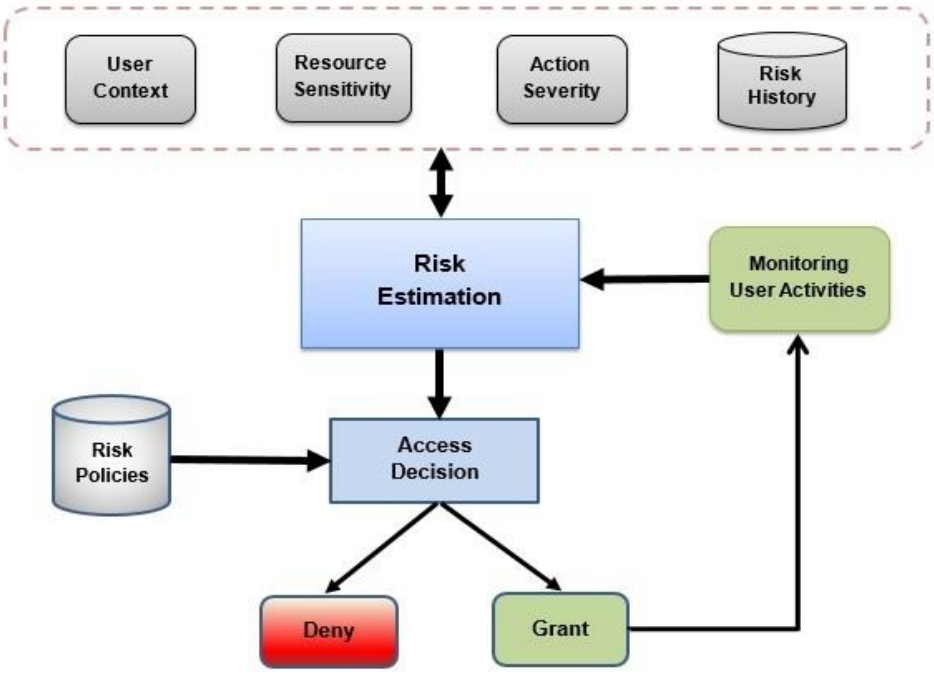


Figure B.1: Proposed risk-based access control model

- **Cold Start**

The proposed risk-based access control model has four risk factors; user context, resource sensitivity, action severity and risk history. Each risk factor is used to estimate the overall risk score associated with the access request to make the access decision. However, the user requesting access for the first time will not have a risk history value. Therefore, the risk estimation module of the proposed risk-based model can not estimate the overall risk value without the risk history value. To solve this issue, we have to add new fuzzy rules containing only three risk factors including user context, resource sensitivity and action severity without risk history. So, if there is no risk history, the other three risk factors will be used to estimate the risk value to make the access decision

- **Fuzzy Rules**

Fuzzy rules are built as IF-THEN statements to describe how the output risk varies as a function of the three risk factors. Fuzzy rules were built using information collected from the literature with researcher experience. The relation between action severity and resource sensitivity that is shown in Figure B. was utilized with the following logical rules:

- If the resource sensitivity increased, the output risk will not decrease.
- If the Action severity increased, the output risk will not decrease.
- If any two inputs are high, the lowest output will be high.
- If the resource sensitivity is Highly Sensitive (HS) or Sensitive (S), the output risk cannot be Negligible (N).

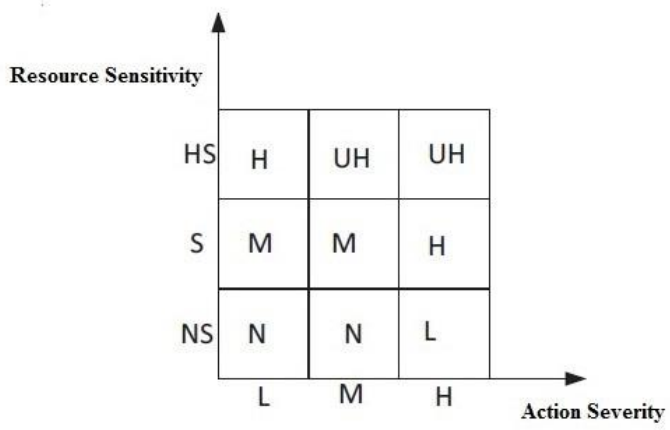


Figure B.2: Risk value regarding resource sensitivity and action severity

In addition, three fuzzy sets for each risk factors/input and five fuzzy sets for the output risk were proposed, as shown in Figure B.3.

Input variable: User Context (UC)		
Linguistic value	**Notation**	**Range**
Low	L	0 - 0.4
Moderate	M	0.3 - 0.7
High	H	0.6 - 1
Input variable: Resource Sensitivity (RS)		
Linguistic value	**Notation**	**Range**
Not Sensitive	NS	0 - 0.35
Sensitive	S	0.2 - 0.5
Highly Sensitive	HS	0.45 - 1
Input variable: Action Severity (AS)		
Linguistic value	**Notation**	**Range**
Low	L	0 - 0.4
Moderate	M	0.35 - 0.7
High	H	0.6 - 1
Output variable: Risk (R)		
Linguistic value	**Notation**	**Range**
Negligible	N	0 - 0.3
Low	L	0.1 - 0.4
Moderate	M	0.2 - 0.6
High	H	0.4 - 0.8
Unacceptable High	UH	0.7 - 1

Figure B.3: Risk value regarding resource sensitivity and action severity

Please, use your knowledge and experience to validate these rules.

Input Notations: L: Low; **M:** Moderate; **H:** High; **NS:** Not Sensitive; **S:** Sensitive; **HS:** Highly Sensitive

Output Notations: N: Negligible; **L:** Low; **M:** Moderate; **H:** High; **UH:** Unacceptable High

Table B.1: Fuzzy rules with the output of cold start

Rule No	Risk Factors			Output Risk	Expert Validation		
	Action Severity	Resource Sensitivity	User context		Yes	No	Output if No
82	L	NS	L	N			
83	M	NS	L	N			
84	H	NS	L	N			
85	L	S	L	L			
86	M	S	L	L			
87	H	S	L	L			
88	L	HS	L	M			
89	M	HS	L	M			
90	H	HS	L	M			
91	L	NS	M	N			
92	M	NS	M	L			
93	H	NS	M	L			
94	L	S	M	M			
95	M	S	M	M			

Table B.1: Fuzzy rules with the output of cold start (Cont.)

Rule No	Risk Factors			Output Risk	Expert Validation		
	Action Severity	Resource Sensitivity	User context		Yes	No	Output if No
96	H	S	M	H			
97	L	HS	M	H			
98	M	HS	M	UH			
99	H	HS	M	UH			
100	L	NS	H	L			
101	M	NS	H	M			
102	H	NS	H	H			
103	L	S	H	UH			
104	M	S	H	UH			
105	H	S	H	UH			
106	L	HS	H	UH			
107	M	HS	H	UH			
108	H	HS	H	UH			