# Towards a non-Intrusive Recognition of Anomalous System Behavior in Data Centers

Roberto Baldoni[1], Adriano Cerocchi[2], Claudio Ciccotelli[1], Alessandro Donno[1], Federico Lombardi[1], and Luca Montanari[1]

[1] Cyber Intelligence and Information Security Research Center,
"Sapienza" University of Rome,
Via Ariosto, 25, Rome, Italy
[2] Over Technologies, Rome, Italy
{baldoni,ciccotelli,lombardi,montanari}@dis.uniroma1.it
cerocchi@overtechnologies.it
ale.dnn@gmail.com

**Abstract.** In this paper we propose a monitoring system of a data center that is able to infer when the data center is getting into an anomalous behavior by analyzing the power consumption at each server and the data center network traffic. The monitoring system is non-intrusive in the sense that there is no need to install software on the data center servers. The monitoring architecture embeds two Elman Recurrent Networks (RNNs) to predict power consumed by each data center component starting from data center network traffic and viceversa. Results obtained along six mounts of experiments, within a data center, show that the architecture is able to classify anomalous system behaviors and normal ones by analyzing the error between the actual values of power consumption and network traffic and the ones inferred by the two RNNs.

**Keywords:** monitoring, failure prediction, dependability, critical infrastructure, data centers, power consumption, network traffic, non-intrusive, black box.

## 1 Introduction

Data centers represent the continuously-growing core infrastructure of every digital service and a basic pillar of our economy. Thus it is imperative to increase their resiliency to failures of internal components like switches, wires, servers, storage etc in order that the failure of one or a few components will not have a major degradation on the performance and on the availability of the software services hosted by the data center. Assuming that components can fail unexpectedly during service operation, to increase such resiliency there is the need of advanced monitoring system at data center scale that are able to infer if something is going wrong in order to take appropriate actions at due time. Almost all such monitoring systems are developed as intrusive software in the sense that they need to install an agent on each monitored system, sharing resources with the monitored system. Apart of the disturbance that agents can provoke to the monitored system, this approach imposes a large usage of human resource to install and to keep updated the monitoring agents with the consequent explosion of operative cost. Thus a suitable approach to datacenter monitoring should minimize the deployment and management costs being also agnostic with respect to applications running in the data center. Agnosticness can be achieved by using a black-box approach to monitoring.

For black-box monitoring we mean the monitoring system can only have access to external health indicators such as: temperature, humidity, network flows, power consumption. In a previous work [5] we considered network traffic exchanges among the data center servers and their power consumption showing the there is a sharp correlation between these two metrics. In this paper we exploit this result in order to detect an anomalous behavior of data center servers.

Thus we present NiTREC, a non-intrusive monitoring architecture that takes as input data center network traffic and the aggregate of servers' power consumption. The network traffic is used to infer data center power consumption and vice versa. Thus NiTREC is able to recognize any deviation of the data center behavior by evaluating the error between inferred values and actual values. Two Elman Recurrent Networks (RNNs) are used in Nitrec to infer the aggregate power consumption and data center network traffic.

In order to assess NiTREC capabilities, we did an extensive experimental evaluation in a real data center owned by the Italian Ministry of Economy and Finance. After an accurate training of the RNNs, we show that NiTREC is able to recognize deviations from the normal system behavior with a high level of accuracy. We also compare accuracy obtained by each of the two RNNs.

## 2 Background

To better understand the architecture functioning, some details about non-intrusive monitoring and about Artificial Neural Networks are required. After these details, NiTREC architecture is presented.

*Non-intrusive Monitoring* An intrusive approach to monitoring relies on installing software probes on each single monitored component (e.g., blade servers). The management cost of the monitoring system (installation, configuration, etc.), in terms of human and economic resources, in such complex environments can be excessive or even prohibitive for many organizations. Conversely, a non-intrusive approach does not require to install software on each server. Instead, it relies on a small number of hardware probes properly deployed leading to more affordable management costs. For this reason a non-intrusive approach is often an appealing solution, to be deployed together to legacy monitoring systems.

We considered two quantities that can be monitored without installing software on observed systems: network traffic and power consumption. Network traffic can be monitored directly at the network switches level, using network sniffers deployed in strategic positions of the data centers. Indicators like packet rate, bandwidth, message size can be in this way easily computed. Power consumption can be monitored by deploying very precise energy meters, in order to solve the problem due to the fact that blade servers-based systems aggregate the consumption. Active power, Reactive power and phase displacement can be measured.

*Artificial Neural Networks* An *Artificial Neural Network* (ANN) is a machine learning computational model capable to approximate any non-linear function of its input, widely used for pattern recognition and forecasting. ANN are structured as a weighted interconnection system of neurons spread in levels, where the input level contains the neurons corresponding to the features, the output level contains the neurons with the estimated resulted values and in the middle, in order to improve the prediction accuracy, one or more hidden levels could be insert. The weights of each neuron interconnection are tuned by a learning algorithm, generally based on gradient-descent as the Backpropagation, the most popular one [19]. Time-series forecasting, in particular for power electric load, is a well-known problem often addressed with ANNs [11, 13, 7].

Our aim is to exploit the ANN capabilities to infer real-time power consumption starting from network traffic observation and viceversa: to infer network traffic starting from power consumption measurement. The core concept that made this possible is the correlation among the two metrics found in [5].

## 3 NiTREC architecture

The architecture that we present here has been named NiTREC, Non-inTrusive deviation Recognizer Exploiting Correlation. It is designed to monitor in a non-intrusive way a single enclosure of a datacenter, to learn the correct system behavior and to recognize deviations from that. Considering the advantages of having a non-intrusive system to monitor and enhance resiliency of a critical infrastructure data center, we designed and implemented the NiTREC architecture so as to measure and to correlate network data and power consumption in real time, using artificial neural networks. NiTREC is able to recognize deviations from the correct system behavior after an initial phase of training. The architecture is depicted in Figure 1. The whole architecture lives inside a centralizer, an ordinary computer or a blade server, collecting measurement from the network traffic and the power consumption probes. It takes in input (i) $n$ streams of network packets[3], directly produced by $n$ probes (network sniffers that capture packets from the switches of the observed system) and (ii) a stream of power consumption data from the smart-PDUs (that measure with high precision the power consumption of the monitored enclosure) developed by Over [1]. The architecture produces in output alerts as soon as the monitoring system recognizes deviations from the correct enclosure behavior. Three modules compose the architecture, a description of them is now provided:

[3] In the well-known pcap format.

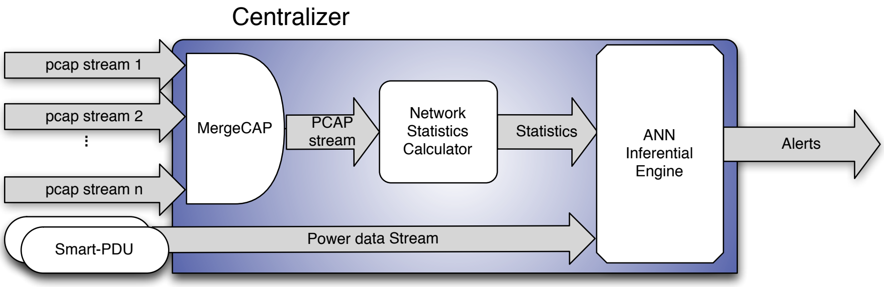

**Fig. 1.** NiTREC, a non-intrusive deviation recognizer exploiting correlation between power consumption and network traffic.

*MergeCAP* a software module that takes in input $n$ streams of captured packets and gives in output a single network stream opportunely merged[4].

*Network Statistics Calculator* a software module that takes in input the network stream and, according to a set of parameters, produces in real-time indicators (e.g., message rate, bandwidth, message size, message rate per physical machine). The indicators are grouped in tuples and produced in real-time with a given frequency, for instance, one tuple per second. This led to have a snapshot of the observed system per second, for example, if we consider message rate, bandwidth, tcp messages, average message size, we would have a tuple, like the following, per second:$< sec : 3; 4387 msg/s; 14042896 bps; 2632 tcp\_msgs; 400 byte >$ meaning that during the third second of observation there have been 4387 messages, a mean bandwidth of 14042896 bit per second, 2632 tcp messages and an average message size of 400 bytes.

*ANN Inferential Engine* a software module that using indicators tuples received from the previous module and power consumption data, correlates them and according to an implementation of artificial neural network, triggers timely alerts if it recognizes deviations from correct system behavior. The ANN Inferential Engine is a crucial part of the architecture, which requires an accurate learning phase in order to build a knowledge base regarding the observed system, more details are provided in Sec. 4.

## 4 Experimental analysis

We conducted a six months long experimental session along with Sogei s.p.a., a company of Italian Ministry of Economic and Finance (MEF) that manage the IT of the ministry. In particular we deployed the NiTREC architecture in order

[4] Merging network traces is a solved problem, several tools are available. A synchronization of the probes is required e.g., a NTP server.

to monitor a single enclosure of one of the data centers of MEF. For this initial part of the work, we collected traces for off-line processing only. Note that all the probes were passive with respect to the monitored system and connected to each other through a switch external to the data center. Therefore, the monitoring did not introduce additional traffic or delays in the monitored system.Please refer to [5] for details about probes deployment and dataset creation.

### 4.1 Testbed and Dataset

The data center is a medium-size facility, featuring 80 physical servers; 250 virtual servers; 20 network devices; 8 security devices; more than 50 different Web Applications; 2 Storage Area Network with more than 6 TB of disk space; more than 1000 internal users and more 80.000 external managed single users; We monitored a single enclosure that embeds 5 blade servers, 40 virtual machines, 4 network switches[5]. Each blade server has 24 cores and 64 GB of RAM. We recorded a mean packet rate around 2000 pps, with spikes from 10000 to 25000000 pps while the active power consumed is between 1550 and 1600 watts. The dataset created is composed by approximately 2.5 TB of pcap network traces and power consumption data, representing the behavior of the monitored servers from a network and power consumption point of view, during the period 31 July 2013 - 31 January 2014.

### 4.2 Neural Networks Implementation and details

For this work we used Encog 3.2.0 [2] as machine learning framework to employ two Elman Recurrent Networks, namely: RNN1, which is designed to infer power consumption having packet rate as input and RNN2, which is designed to infer packet rate having power consumption as input. In particular, RNN1 and RNN2 are both 4-5-1 networks: 4 inputs nodes, a single hidden layer of 5 nodes and the output node. RNN1 takes as input packet rate, day, hour and power consumption at the last-seen instant. It produces as output the inferred power consumption. RNN2 takes as input power consumption and traffic rate at the last-seen instant, day and hour. It produce as output the inferred packet rate.
Both the RNNs are trained using Resilient Backpropagation algorithm[12] as long as the choice of the input node variables is due to a hybrid approach between time-series and features, as suggested in [4].

### 4.3 Preliminary results

The idea of the experimental campaign is to evaluate the ability of the approach in recognizing deviations from normal behavior of the observed system. In particular, we evaluated two cases: estimating power consumption from packet rate and viceversa. In order to do that, we used RNN1 and RNN2 after a learning

[5] Network traffic has been monitored through 4 hardware probes attached to the switches

phase. We used a small part of the dataset (10 days) as training set and a different part (3 days) as validation set. Note that the validation phase is performed off-line, using traces, but is completely equivalent to a physical deployment of the architecture, in detection mode. Not having the possibility to inject faults in the observed system[6], during the validation, we introduced a deviation in the metric used to infer the other and we observed how the deviation reflects on values inferred by the RNN. The idea behind this approach is that a deviation (e.g., an unjustified augment of power consumption), may reveal a faulty behavior of software or hardware components. We found the percent error $\delta = 100 \times |\frac{v - v_{inferred}}{v}|$ where $v_{inferred}$ is the inferred value and $v$ is the actual value to be an effective metric to detect deviations. When the percent error exceeds a given threshold $\hat{\delta}$, we trigger an alert. The threshold $\hat{\delta}$ has been chosen in order to maximize the F-measure (see below) but more complex approaches can be considered. Figure 2 and Figure 3 graph the behavior of active power, packet rate and percent error during time. The chosen threshold of percent error has been depicted and the samples over this that have been highlighted as well. In the first case, represented in Figure 2, we deployed RNN2, which infers packet rate starting from power consumption. In the first part of the graph, until 12:00, the ability of RNN2 in its inference task can be appreciated. After that, we started to progressively increase the power consumption at 12.00 causing an increase of the percent error. Even small unattended increases of power consumption quickly cause an augment of alert, due to augments of percentage error. In the second case (see Figure 3) we deployed RNN1, in order to infer power consumption starting from network traffic. Also in this case, during the first part of the graph (until time 12:00) the ability of RNN1 its inference task can be appreciated, which is better respect the RNN2 case. After that, we started to inject spare packets incrementally. The inferred power consumption started to deviate from the measured power, thus augmenting the percent error, as soon as the packet rate reached 10000 pps. In this case a more relevant deviation is required in order to have appreciable variation in the inferred value. Note that, according to the low error obtained during the period of normal functioning, an augment of the error can fairly be assumed as an uncommon situation.

In order to better evaluate the accuracy of the proposed approach in both cases, we computed the metrics reported in Tab. 1 and Tab. 2, where $N_{tp}$ (number of true positives) indicates the number of alerts correctly produced, i.e., during a deviation from the correct system behavior; $N_{tn}$ (number of true negatives) is the number of samples of percent error that correctly are under the alert threshold, i.e., during correct system behavior; $N_{fp}$ (number of false positive) is the number of alerts incorrectly produced, i.e., during correct system behavior and finally $N_{fn}$ (number of false negatives) is the number samples that incorrectly are under the alert threshold, during a deviation from the correct system behavior.

[6] The system is not a test environment but a real Critical Infrastructure datacenter enclosure in production.

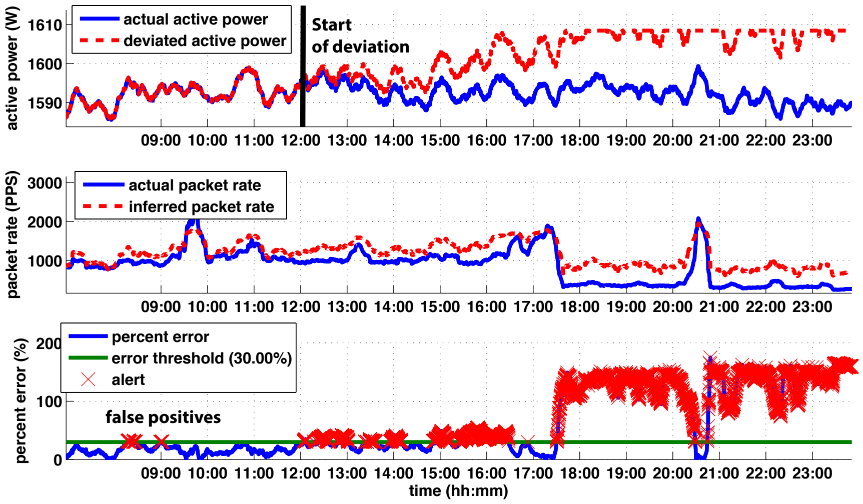

**Fig. 2.** RNN2 results. Packet rate is inferred with a good accuracy until 12:00, where the power consumption has been progressively increased causing an augment of the error. Some false positives can be seen before 9:00. The first true positive alert has been triggered at 12:00.

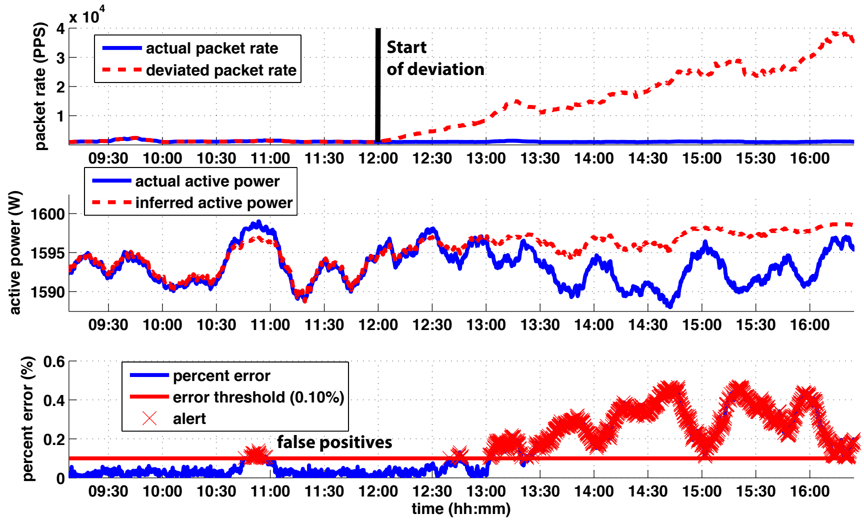

**Fig. 3.** RNN1 results. Power consumption is inferred with a better accuracy until 12:00, w.r.t. RNN2. After that, spare packets have been injected in the network trace causing an augment of the error. Some false positives can be seen before 11:00. The first true positive alert has been triggered at 12:40.

| | |
|---|---|
| Precision: $p = \frac{N_{tp}}{N_{tp}+N_{fp}}$ | 85.34% |
| Recall (TP rate): $r = \frac{N_{tp}}{N_{tp}+N_{fn}}$ | 87.45% |
| F-measure: $F = 2 \times \frac{p \times r}{p+r}$ | 86.38% |
| FP Rate: $f.p.r. = \frac{N_{fp}}{N_{fp}+N_{tn}}$ | 3.00% |

**Table 1.** RNN1 accuracy

| | |
|---|---|
| Precision: $p = \frac{N_{tp}}{N_{tp}+N_{fp}}$ | 90.67% |
| Recall (TP rate): $r = \frac{N_{tp}}{N_{tp}+N_{fn}}$ | 71.42% |
| F-measure: $F = 2 \times \frac{p \times r}{p+r}$ | 79.90% |
| FP Rate: $f.p.r. = \frac{N_{fp}}{N_{fp}+N_{tn}}$ | 1.47% |

**Table 2.** RNN2 accuracy.

In both cases we can see a very low false positive rate and a F-measure of at least of 79.9%, attesting promising future developments of the approach.

## 5 Related Work

Monitoring based only on network traffic is recognized to be non-intrusive and black-box, meaning that (i) no application-level knowledge is needed to perform the monitoring [18, 3, 6], and (ii) the monitor mechanism does not install software on the monitored system [6]. In [6] CASPER is presented, a non-intrusive and black-box approach to monitor air traffic control systems. It uses network traffic only in order to represent the system health so as to recognize deviations thus triggering failure predictions. At the best of our knowledge, this is the only work that is both non-intrusive and black-box. Other monitoring systems that adopt a black-box approach are Tiresias [18] and ALERT [14], however they are intrusive as they require monitoring software installed on the monitored system. For what concern power consumption monitoring in data centers, studies have been conducted in the context of power management and energy efficiency [10, 16, 17]. None of these works, however, concerns dependability and resiliency. In [8] and [15] network traffic is monitored with the aim of consolidating traffic flows onto a small set of links and switches so as to shut down unused network elements, thereby reducing power consumption. However, there is no attempt to correlate network traffic and power consumption. In [9] a study on correlation between power consumption data and utilization statistics (CPU load and network traffic) is presented. This work shows a strong correlation between power consumption and CPU load of desktop computers. Our previous work [5] investigates the correlation between power consumption and network traffic to support the design of a non-intrusive black-box failure prediction system for improving data center resiliency. The paper reports the results of a period of experimentation conducted in one of the data centers of the Italian Ministry of Economic and Finance (MEF) during which a large dataset of network traffic and power consumption data is collected and analyzed, thus showing that correlation between these data exists in many periods. To the best of our knowledge this was the first work that explored the possibility to exploit correlation between power consumption and network traffic to support dependability of a system. In this work we used the same dataset.

## 6 Conclusions and future work

This work is a first step in exploiting in a non-intrusive way the correlation between network data and power consumption to recognize and predict component failures in data centers. During a preliminary 6-months long experimental campaign we created a dataset (in a completely non-intrusive way) with respect to the data center's components (network and servers). The dataset allowed us to train two neural networks in order to estimate power consumption observing network traffic and vice versa. We found that the neural networks can be used to effectively detect anomalous system behavior looking at deviations from data center network traffic and an aggregate of power consumption of each data center component. A deviation from the behavior, learnt during the training phase, can be used to trigger alerts. As future work, we need to reduce the level of granularity of the study by looking at correlation on the behavior of a single data center component. In this paper we are only considering correlation between aggregate measures, namely network traffic and power consumption. We are finally developing more complex alert techniques in order to provide a more effective detection with respect to the threshold mechanism used in this work.

## Acknowledgment

This work was partially supported by the PRIN project TENACE. The authors would like to thank the Italian Ministry of Economy and Finance for allowing the experimentation in their data centers.

## References

[1] Over s.r.l. website `http://www.overtechnologies.com`.

[2] Encog Machine Learning Framework. `http://www.heatonresearch.com/encog/`, 2008.

[3] Marcos K. Aguilera, Jeffrey C. Mogul, Janet L. Wiener, Patrick Reynolds, and Athicha Muthitacharoen. Performance debugging for distributed systems of black boxes. *In SIGOPS Oper. Syst. Rev.*, 37:74–89, October 2003.

[4] L. Aniello, R. Baldoni, S. Bonomi, F. Lombardi, and A. Zelli. An Architecture for Automatic Scaling of Replicated Services. In *To appear in the Proceedings of the 2nd International Conference on NETworked sYStems (NETYS)*, 5 2014.

[5] R. Baldoni, M. Caruso, A. Cerocchi, C. Ciccotelli, L. Montanari, and L. Nicoletti. Correlating power consumption and network traffic for improving data centers resiliency. *ArXiv e-prints*, May 2014.

[6] Roberto Baldoni, Giorgia Lodi, Luca Montanari, Guido Mariotta, and Marco Rizzuto. Online black-box failure prediction for mission critical distributed systems. In *SAFECOMP*, pages 185–197, 2012.

[7] Ray J Frank, Neil Davey, and Stephen P Hunt. Time series prediction and neural networks. *Journal of Intelligent and Robotic Systems*, 31(1-3):91–103, 2001.

[8] Brandon Heller, Srini Seetharaman, Priya Mahadevan, Yiannis Yiakoumis, Puneet Sharma, Sujata Banerjee, and Nick McKeown. Elastictree: Saving energy in data center networks. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, pages 17–17, Berkeley, CA, USA, 2010. USENIX Association.

[9] Maria Kazandjieva, Brandon Heller, Philip Levis, and Christos Kozyrakis. Energy dumpster diving. In *SOSP '09: Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles*, New York, NY, USA, 2009. ACM.

[10] Charles Lefurgy, Xiaorui Wang, and Malcolm Ware. Power capping: A prelude to power shifting. *Cluster Computing*, 11(2):183–195, June 2008.

[11] Dong C Park, MA El-Sharkawi, RJ Marks, LE Atlas, MJ Damborg, et al. Electric load forecasting using an artificial neural network. *Power Systems, IEEE Transactions on*, 6(2):442–449, 1991.

[12] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *Neural Networks, 1993., IEEE International Conference on*, pages 586–591. IEEE, 1993.

[13] Tomonobu Senjyu, Hitoshi Takara, Katsumi Uezato, and Toshihisa Funabashi. One-hour-ahead load forecasting using neural network. *Power Systems, IEEE Transactions on*, 17(1):113–118, 2002.

[14] Yongmin Tan, Xiaohui Gu, and Haixun Wang. Adaptive system anomaly prediction for large-scale hosting infrastructures. In *Proc. of ACM PODC 2010*, pages 173–182, New York, NY, USA, 2010. ACM.

[15] Xiaodong Wang, Yanjun Yao, Xiaorui Wang, Kefa Lu, and Qing Cao. Carpo: Correlation-aware power optimization in data center networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 1125–1133, March 2012.

[16] Xiaorui Wang and Ming Chen. Cluster-level feedback power control for performance optimization. In *High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on*, pages 101–110, Feb 2008.

[17] Xiaorui Wang and Yefu Wang. Co-con: Coordinated control of power and application performance for virtualized server clusters. In *Quality of Service, 2009. IWQoS. 17th International Workshop on*, pages 1–9, July 2009.

[18] Andrew W. Williams, Soila M. Pertet, and Priya Narasimhan. Tiresias: Black-box failure prediction in distributed systems. In *Proceedings of IEEE International Parallel and Distributed Processing Symposium (IPDPS 2007)*, Los Alamitos, CA, USA, 2007.

[19] Guoqiang Zhang, B. Eddy Patuwo, and Michael Y. Hu. Forecasting With Artificial Neural Networks: the State of the Art. *International Journal of Forecasting*, 14(1):35 – 62, 1998.