Decision Support Systems xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Isolation-based conditional anomaly detection on mixed-attribute data to uncover workers' compensation fraud

Eugen Stripling[a,*], Bart Baesens[a,b], Barak Chizi[c], Seppe vanden Broucke[a]

[a] Department of Decision Sciences and Information Management, KU Leuven, Leuven, Belgium
[b] School of Management, University of Southampton, Southampton, UK
[c] Department of Information Systems and Software Engineering, Ben-Gurion University of the Negev, Beer-Sheva, Israel

ARTICLE INFO

Keywords:
Workers' compensation insurance fraud
Fraud detection
Conditional anomaly detection
Isolation forest

ABSTRACT

The development of new data analytical methods remains a crucial factor in the combat against insurance fraud. Methods rooted in the research field of anomaly detection are considered as promising candidates for this purpose. Commonly, a fraud data set contains both numeric and nominal attributes, where, due to the ease of expressiveness, the latter often encodes valuable expert knowledge. For this reason, an anomaly detection method should be able to handle a mixture of different data types, returning an anomaly score meaningful in the context of the business application.

We propose the *i*Forest$_{CAD}$ approach that computes conditional anomaly scores, useful for fraud detection. More specifically, anomaly detection is performed conditionally on well-defined data partitions that are created on the basis of selected numeric attributes and distinct combinations of values of selected nominal attributes. In this way, the resulting anomaly scores are computed with respect to a reference group of interest, thus representing a meaningful score for domain experts. Given that anomaly detection is performed conditionally, this approach allows detecting anomalies that would otherwise remain undiscovered in unconditional anomaly detection.

Moreover, we present a case study in which we demonstrate the usefulness of our proposed approach on real-world workers' compensation claims received from a large European insurance organization. As a result, the *i*Forest$_{CAD}$ approach is greatly accepted by domain experts for its effective detection of fraudulent claims.

1. Introduction

Across all lines of insurance, it is conservatively estimated that fraud causes a monetary damage of $80 billion a year [1]. Given this estimate, it is self-evident that insurance fraud is a major problem that adversely affects our society [2]. Among the various insurance lines, the Insurance Information Institute [3] (or short III), reported that the majority of industry experts (69%) believes in an increase of workers' compensation (WC) insurance fraud. WC is an insurance policy to cover costs that emerge when employees sustain an injury or become ill on the job. Fraudsters view the deprivation of money from insurance organization as a *low-risk*, *high-reward game*, since it is far safer than other money earning, serious crimes such as armed robbery or drug trafficking [1, 4]. It should therefore be of no surprise that even when considering WC insurance alone, the total loss caused by WC fraud can easily reach tens of millions of dollars [5]. Taking these points into consideration, it is therefore strongly advised by the III [6] to invest in the advances of analytical technology to protect the insurance organization and their honest clients against the ever-changing nature of increasingly intricate fraud practices.

In this paper, we propose a novel analytical approach, called *i*Forest$_{CAD}$, that performs isolation-based anomaly detection *conditionally* on reference groups (i.e., data partitions) meaningful to domain experts. The resulting *i*Forest$_{CAD}$ anomaly scores are then leveraged for fraud detection. Data partitions are defined by distinct combinations of values of selected nominal attributes, thereby integrating a mixture of nominal and numeric attributes in a meaningful way. Based on the observation that fraud data sets usually consist of both nominal and numeric attributes [7], our proposed *i*Forest$_{CAD}$ approach aims to fulfill the strong desire to make use of all available information in the combat against fraud.

Moreover, we present a case study in which we apply our *i*Forest$_{CAD}$ approach on a data set of real-world WC claims received from a large European insurance organization. For the study, we collaborated with

* Corresponding author at: Department of Decision Sciences and Information Management, Naamsestraat 69 - box 3555, Leuven 3000, Belgium.
E-mail addresses: eugen.stripling@kuleuven.be (E. Stripling), bart.baesens@kuleuven.be (B. Baesens), chizba@bgu.ac.il (B. Chizi), seppe.vandenbroucke@kuleuven.be (S. vanden Broucke).

https://doi.org/10.1016/j.dss.2018.04.001
Received 12 November 2017; Received in revised form 17 April 2018; Accepted 17 April 2018

the insurer's special investigation unit (SIU) to fruitfully incorporate valuable expert knowledge of the private investigators (PIs) to enhance the automatic detection of fraudulent WC claims. One of the most important project goals in order to reach acceptance among the PIs is that the fraud detection approach satisfies the following major requirement:

Interestingness: It is to be ensured that suspected fraud cases reported to the PIs are *interesting*. This implies that the reported claims should conform with the PIs' expert knowledge (i.e, detection of fraudulent claims in line with known fraud patterns). Yet, the reported claims should also have some element of novelty and surprise (i.e., discovery of previously unseen fraud patterns).

The implications for the data scientist of the main requirement can be stated as follows:

1. **Integration of expert knowledge**: The inclusion of accumulated expert knowledge into the fraud detection mechanisms plays an essential role in order for the data science application to be accepted by the PIs.
2. **Accuracy**: Due to scarce resources, an efficient deployment of PIs to check and invest suspicious claims is required. Hence, the fraud detection model should be accurate in its predictions so that the PIs first focus their attention to the truly fraudulent claims.
3. **Explainability**: The data scientist *must* be able to explain to the PIs *why* the classification model predicts a claim as fraudulent.
4. **Novelty**: To reach acceptance among the PIs, the fraud detection approach should return fraudulent claims according to known patterns, but also detect novel ones that comply with the expertise of the PIs.

Since the PIs ultimately decide whether or not an in-depth investigation has to be conducted, it is crucial that the fraud detection approach fulfills the aforementioned criteria. In particular, special attention needs to be dedicated to the first criterion, because it is often beneficial to inject expert knowledge into the data analytical approach (see, e.g., [8-10]).

According to the PIs, given the type of injury and other information, an “unusual long” recovery time (or, equivalently, disproportional duration of incapacity), is a strong indicator of a WC claim being fraudulent. To capture this insight in a data-driven manner, one needs to answer the following questions: *How to decide when a recovery time is too long (without requiring human judgment)? How can valuable expert knowledge be integrated into the decision model construction?*

With an interesting real-world case study on WC fraud, we present a fraud detection approach (Fig. 1) that allows detecting claims with a disproportional recovery time in a *fully data-driven* manner by which information of mixed type attributes is processed in a way meaningful to the PIs.

We thereby leverage the well-established anomaly detection algorithm called isolation forest (*i*Forest), introduced by [11, 12]. The application of anomaly detection plays a crucial role as it allows for the *automatic* detection of disproportional recovery times. The *i*Forest algorithm is a key component of our proposed approach which we favor over other anomaly detectors for reasons we elaborate on in Section 2.2. It is important to note that the anomaly scores of the observations are computed conditionally on data partitions, which are defined based on the distinct combination of values of selected nominal attributes. Thus, the name of our proposed approach, $i\text{Forest}_{\text{CAD}}$, roots in the fact that a conditional anomaly detection (CAD) is performed with the aid of the *i*Forest. The created $i\text{Forest}_{\text{CAD}}$ scores are combined with the remaining attributes which then serve as an input for training a supervised classification model. In this way, we exploit the benefits of both supervised and unsupervised learning. The $i\text{Forest}_{\text{CAD}}$ scores proved to be indeed of high importance for the detection of suspicious claims in our case study.

Our research contributions can be summarized as follows:

- We propose the $i\text{Forest}_{\text{CAD}}$ method that computes anomaly scores conditionally on given reference groups (i.e., well-defined data partitions). In this way, $i\text{Forest}_{\text{CAD}}$ is capable of identifying “hidden” anomalies, which we demonstrate in Section 3.4.
- Our approach allows processing a mixture of nominal and numeric attributes, returning a condensed score that is meaningful in the context of the business application. The scores produced by $i\text{Forest}_{\text{CAD}}$ can be used not only for conditional anomaly detection but also as a new numeric input attribute for a predictive model.
- We demonstrate the application of anomaly detection and predictive analytics in the scope of a real-world case study on WC insurance fraud. To the best of our knowledge, the practical application of primarily machine learning techniques to combat WC fraud has not yet been presented in the literature.

The remainder of this paper is structured as follows. The next section provides more background information on WC insurance fraud, discusses anomaly detection and methods, as well as explains the inner workings of the *i*Forest algorithm. Section 3 formally introduces the concepts of our proposed approach with particular focus on the creation of $i\text{Forest}_{\text{CAD}}$ scores. Additionally, in Section 3.4, we provide an example that showcases the detection of hidden anomalies. Section 4 presents the case study in which we demonstrate the usefulness of the $i\text{Forest}_{\text{CAD}}$ scores for the detecting of fraudulent WC claims. In the same section, we elaborate on the effectiveness of applying conditional anomaly detection and how our proposed approach is applied to meet the most important requirement of interestingness. Section 5 summarizes the main findings of our work and highlights potential research directions.

2. Preliminaries

2.1. Workers' compensation fraud

Workers' compensation (WC) insurance provides a cost coverage in case employees sustain a work-related injury or disease that occur as a result of performing their occupational duties [3, 13]. For example, in the USA, coverage may be required for costs such as wage replacement, medical care and rehabilitation, and death benefits for the dependents if the employee deceased in work-related accidents (including terrorist attacks) [3].

According to the latest issue update on insurance fraud [6], it is believed that WC is one of the most vulnerable insurance lines to fraud. Further, III reported that 69% of industry experts forecast an increase in WC fraud. This strongly suggests initiating appropriate measures in order to protect insurance organizations and their honest clients against fraudsters. To do so, III pointed out that advances in analytical technology are a crucial factor in order to be able to keep up with the ever-changing nature of increasingly complex and sophisticated fraud schemes.

Viaene and Dedene [4] characterized insurance fraud by the presence of (at least) the following elements: (1) Misrepresentation of circumstances or material facts in the form of lie, falsification, or concealment, (2) deliberate plan of deception, and (3) purpose to gain unauthorized benefits. The authors further classified insurance fraud into three broad categories: (1) internal versus external, (2) underwriting versus claim, and (3) soft versus hard.

The first category (*internal versus external*) attempts to distinguish between the various types of perpetrators. Internal fraud is committed from within the insurance organization, e.g., by insurers, agents, and insurer employee. External fraud is perpetrated by individuals outside the organization, e.g., by applicants, policyholders, and claimants. The distinction sometimes becomes blurry in situations that involve a

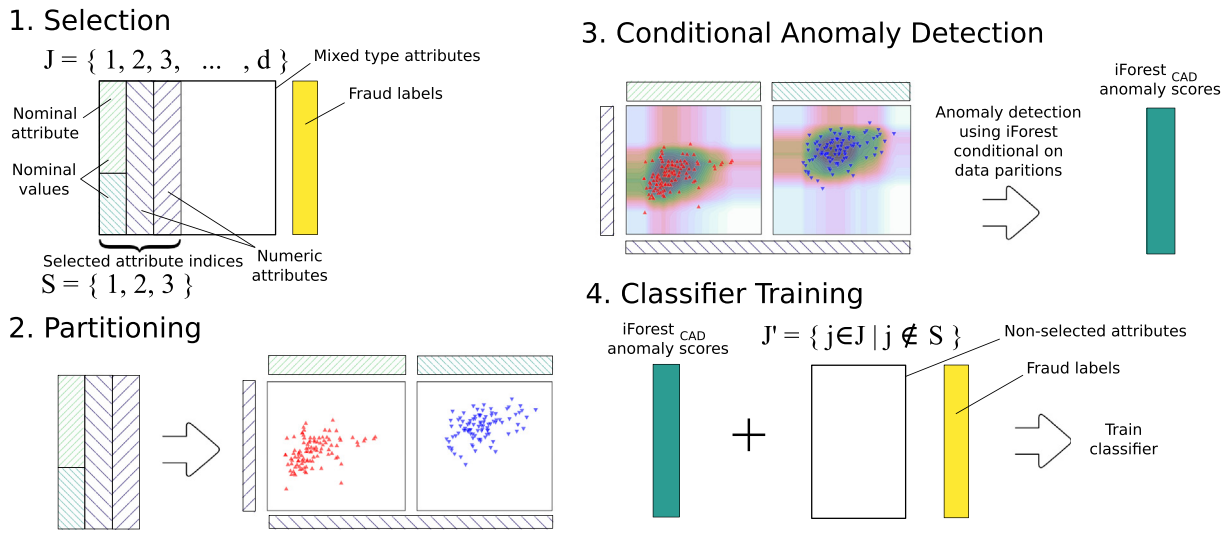

Fig. 1. Operating principle of the proposed *i*Forest$_{CAD}$ approach to perform conditional anomaly detection and leveraging anomaly scores for fraud detection. Steps 1 to 3 involve the creation of the *i*Forest$_{CAD}$ anomaly scores based on a selection of mixed type attributes. Thus, the *i*Forest$_{CAD}$ approach is capable of processing a mixture of nominal and numeric attributes in a meaningful way. Step 4 encompasses the training of a binary classifier in which the scores created in the previous step serve as an input.

collusion between internal and external parties.

The second category (*underwriting versus claim*) aims to address the various types of fraud, where it is of particular importance to distinguish between perpetrating fraud at underwriting and at claim time. The former refers, for example, to fraudulent activities at the time of the renewal of the insurance contract or the misrepresentation of information during the application (application fraud) with the aim to attain either coverage or a lower premium (premium fraud). On the other hand, the latter type of insurance fraud is typically more prominent and refers to claim fraud in which claims are deliberately inflated, false, or fictitious.

The final category (*soft versus hard*) aims to provide an indication of the degree of intent by assigning labels to the severity of the committed fraud. Soft fraud, often also referred to as opportunistic fraud, describes the cases in which, for example, the claimant seizes the opportunity to exaggerate the damage of an otherwise legitimate claim (claim padding). In contrast, hard fraud is typically associated with carefully planned and well-executed scams. Clearly, hard fraud refers to well-organized crime executed by cunning individuals with malicious intent or sophisticated fraud rings (e.g., deliberately filing bogus claims).

Information asymmetry is the natural fertilizer for fraud [4]. The party with the information advantage has the upper hand in the business relationship which fraudsters leverage to their advantage to receive a more beneficial business deal. In case of WC, claimants are naturally in an advantageous position when filing the claim, since the insurer has often no other option than to trust the provided information filed in the claim. Here, fraud can range, for example, from the exaggeration of a minor injury (i.e., opportunistic fraud) to more severe scenarios such as staging an accident (i.e., hard fraud).

Unlike other social insurances, WC benefits essentially compensate individuals for not working [13]. Research studies have shown that the number of filed claims will generally increase as benefits increase [13-16], as well as that economic incentives significantly affect the claim duration [16-18]. In this sense, fraudsters will constantly try to find ways to outwit the system. Once, they are in the position of receiving benefits, they likely attempt to unduly prolong the period of compensation [19].

Additionally, Bolduc et al. [19] showed that, under certain assumptions, the level of WC benefits has a stronger impact on the probability of reporting a hard-to-diagnose injury (e.g., back-related injuries, sprains, strains, and stress-related problems) than on the probability of reporting an easy-to-diagnose injury (e.g., contusion, fracture, and friction burn). Lower back pain in particular is a common medical problem that is hard to diagnose. Its challenging characteristics are, for example, discussed by [20]. Hard-to-diagnose injuries makes it easier for fraudsters, as insurers have more difficulties to verify the true nature of the injury [19].

A conviction requires hard evidence that proves fraudulent behavior "beyond reasonable doubt," but this comes at a high price since much effort and time have to be put into the procurement of definitive proof [4]. Taken all together, this makes WC fraud an interesting and challenging problem for both business and data science.

2.2. Anomaly detection and related work

Anomaly detection is widely used in a large variety of research and application domains such as health care, security, law enforcement, image processing, and text mining. It is, for example, utilized for the detection of network intrusion, malware, industrial damage, novelty, or fraud. Note that the terms *anomaly* and *outlier* are often used interchangeably, as well as that anomaly and outlier detection is closely related to novelty detection. In Hawkins's often quoted words [21]: "An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism." Thus, outliers or anomalies are generally considered to be *exceptions* or *peculiarities* in the data that do not conform to the normal or expected behavior of the majority. Therefore, the task of an anomaly detection method is to learn the characteristics of a data set in order to be sufficiently capable of distinguishing anomalous data points from normal ones [7]. The choice of the anomaly detection method thereby strongly depends on the type of data. An important distinction to make is between *record data* or *point data*, where no relationship is assumed among the data instances, and *complex data* such as sequence, spatial, or graph data that express the interrelation between observations in some way [7]. It comes with no surprise that there exists an abundance of proposed methods for various application domains and types of data. Accordingly, many scientific works have been published in the past years that summarize methods for anomaly detection from very general to very specific applications and types of data (see, e.g., [7] and[22-33]).

In this paper, we apply anomaly detection in order to detect fraudulent WC claims. To this end, we make use of an anomaly detection

Table 1
Overview of common, state-of-the-art, and related anomaly detection methods.

Method	Measure	Description
*k*NN distance [48]	Distance	Computes a *k*NN outlier score for each observation, which has an intuitive interpretation. That is, data points with large distances to their nearest neighbors are more likely to be anomalous. Being a *k*NN-based method, it can handle multidimensional data.
Method by [49]	Distance	Is a *k*NN-based method combined with a partition-based algorithm, which makes the algorithm computationally more efficient and scalable. *Note*: Study performed only on data with numeric attributes.
ORCA [50]	Distance	Is a *k*NN-based method that uses sample randomization together with a pruning rule. The calculation of *k*NN distances is computationally intensive with non-linear complexity. ORCA reduces it to near linear time. Next to *k*, an additional parameter, which determines how many anomalies to report, needs to be set manually.
Method by [51]	Distance	Computes a proposed distance measure for data with mixed type attributes in which links are defined between two observations by the distance for nominal and numeric attributes separately. It can handle a mixture of nominal and numeric attributes, by maintaining a covariance matrix for each itemset.
LOF [52]	Density	Computes a LOF value for each observation, which in turn indicates the sparseness of a data point with regard to its *k* nearest neighbors. In this way, it accounts for local density variation among data points. Observations with a high LOF value are viewed as anomalies.
COF [53]	Density	A variation of LOF in which the *k* neighborhood for each observation is computed differently. More specifically, the *k* neighborhood (i.e., a set of instances) of a given observation is created by incrementally including the instance that has the minimal distance to any member of the given set of instances (until *k* is reached). On the basis this *k* neighborhood, the outlier score is computed as in LOF.
DBSCAN [54, 55]	Clustering	Is a density-based clustering algorithm in which instances are grouped together that have many nearby neighbors. This allows the algorithm to find clusters of arbitrarily shape. By design, it accounts for noise and is robust to outliers. Hence, an instance with its nearest neighbors being too far away so that it does not belong to a cluster (i.e., lies in a low-density region) is regarded as an outlier. It does not require to set the number of clusters *a priori* as, for example, in the *k*-means algorithm.
HDBSCAN* [56, 57]	Clustering	Inspired by DBSCAN, it generates a complete density-based clustering hierarchy, a composition of all possible density-based clusters, that can effectively be used for cluster analysis and outlier detection. As for outlier detection, the method computes the so-called GLOSH (Global-Local Outlier Scores from Hierarchies) scores, which allows the simultaneous detection of both global and local types of outliers.
OC-SVM [58, 59]	ML model	Constructs a decision boundary around the normal observations. Any instance falling outside the boundary is regarded as an anomaly. The incorporation of nominal attributes is straightforward (e.g., through dummy variables). Yet, it has one or more hyperparameter(s) that require(s) tuning. Also, it is computationally much more demanding than other methods as, for example, *i*Forest.
RFPM [60]	ML model	Since RF is a supervised learner, synthetic data are generated as the alternative class by uniformly sampling in the domain of each attribute. Next, the learned RF is used to construct a matrix that contains a proximity measure for every pair of instances. An observation is viewed as anomalous when its proximities to all other instances are small. Thanks to the use of RF, mixed type attributes can be handled in a straightforward manner. However, maintaining the proximity matrices comes with a large memory requirement.
EXPoSE [61, 62]	ML model	Applies a kernel embedding of distributions that maps a probability measure into a reproducing kernel Hilbert space in order to manipulated it there efficiently. This allows estimating the similarity between a new unseen instance and the distribution data under normal conditions. Thanks to the kernel embedding representation, no parametric assumptions or explicit description of the probability measure are made. The algorithm is designed for large-scale anomaly detection problems. However, as of the time of writing, no publicly accessible software implementation is available.
Method by [63]	Statistical model	Is a GMM-based algorithm for conditional anomaly detection in which two sets of multivariate Gaussians are modeled that correspond to the *environmental* and *indicator* attributes, as well as a dependence structure between the two sets of Gaussians. It computes the likelihood of the occurrence of an observation's indicator values *given* the values of its environmental attributes. Main disadvantage of using a GMM is that as the dimensionality increases calculations become intractable. *Note*: Study performed only on data with numeric attributes.
*i*Forest [11, 12]	Isolation	First to propose an unsupervised, tree-based ensemble method that applies the novel concept of isolation. Authors showed that isolation is a better indicator for anomaly detection than distance and density. *i*Forest is designed to directly model anomalies. It is a computationally very efficient and competitive algorithm. The recommended default values perform well on a multitude of data sets. *i*Forest is less effective in the presence of local and/or clustered anomalies. *Note*: Study performed only on data with numeric attributes.
$i\text{Forest}_{\text{ext}}$ [44]	Isolation	As *i*Forest but maps values of nominal attributes to numeric ones. Mapping of nominal values to numeric ones is arbitrary, hence conceptionally inadequate. Refer to the discussion surrounding Fig. 4.
SC*i*Forest [64]	Isolation	An anomaly detector that works in a similar manner as *i*Forest but is capable of detecting local clustered anomalies through the use of hyperplanes. It detects local clustered anomalies effectively, even when close to normal points. It is less computationally efficient than *i*Forest due to the computation of hyperplanes. *Note*: Study performed only on data with numeric attributes.
*i*NNE [65]	Isolation	An anomaly detector that operates in a similar way as *i*Forest, yet applies an efficient isolation-based *k*NN method. It is computationally more efficient than distance or density-based anomaly detection algorithms. It can detect local anomalies. Yet, it is less computationally efficient than *i*Forest due to the computation of hyperspheres. *Note*: Study performed only on data with numeric attributes.

Abbreviations: Connectivity-based Outlier Factor (COF), EXPected Similarity Estimation (EXPoSE), Gaussian Mixture Model (GMM), isolation Forest (*i*Forest), extended *i*Forest ($i\text{Forest}_{\text{ext}}$), isolation Forest with Split-selection Criterion (SC*i*Forest), isolation using Nearest Neighbour Ensemble (*i*NNE), *k* Nearest Neighbors (*k*NN), Local Outlier Factor (LOF), Machine Learning (ML), One-Class Support Vector Machine (OC-SVM), Random Forest (RF), Random Forest with Proximity Matrices (RFPM).

method that can operate on record data. Yet, the sheer volume of proposed anomaly detection methods makes it impossible to survey all techniques, as it is also not the purpose of this work. Nevertheless, we view it as necessary to briefly describe some state-of-the-art as well as some well-established anomaly detection methods in academic literature (Table 1). The table also lists techniques related to our work.

Ultimately, an anomaly detection method aims to capture a notion of similarity or dissimilarity between data points, which in turn allows it to identify anomalies. There are several ways to capture that notion. Popular approaches for anomaly detection operate via a distance or density measure (e.g., ORCA or LOF), use a data analytical model (e.g., one-class SVM), a clustering technique (e.g., DBSCAN), or apply the state-of-the-art concept of isolation (e.g., isolation forest). In what follows, we compare the anomaly detection methods merely on the high level, rather than on the level of the individual techniques, and motivate our choice for the isolation forest (*i*Forest) [11, 12].

The advantage of distance and density measures is that they have an intuitive interpretation. Anomalous data points are more likely to have large distances to their neighbors, as they are generated—according to Hawkins—by a different mechanism. Density measures are often derived from distances, indicating whether an instance is surrounded by many neighbors (high-density region) or by a few to none neighbors (low-density region). However, it immediately becomes clear that pairwise calculations between instances have to be carried out, and hence the computational complexity increases nonlinearly with the sample size. Additional clever algorithmic mechanisms are required

(e.g., sampling, pruning, or partitioning) in order to achieve a near linear complexity.

Another strategy to detect anomalies is by first modeling the normal data, then identifying those data points that are not regarded as normal. Most model-based methods that were first designed for other purposes (e.g., classification) operate in this way. This is especially also true for clustering-based methods. These methods detect anomalies as a by-product, and hence are usually not optimized for anomaly detection [7]. Consider cluster analysis, for example. A clustering algorithm is designed to find groups of data points that lie relatively close to each other. If anomalies form clusters by themselves, however, a clustering algorithm will not be able to detect such anomalies [7]. On top of it, several clustering algorithms, as for example *k*-means, force every instance to belong to some cluster. Hence, they are more likely to overlook anomalies.

In contrast to the other approaches, isolation-based methods aim to *directly* model anomalies. These methods apply the notion of isolation in which the anomaly detector attempts to separate instances in a specific way. Under the assumption that anomalies are *few and different*, instances that are easier to isolate from the rest are more likely to be anomalies. In fact, in a simple example, [11] and[12] showed that isolation is a better indicator for anomaly detection than distance and density. That is, in the example, normal data points close to the dense anomaly cluster exhibit a larger distance or lower density than the anomalous points, which is the opposite of the desired outcome, whereas the isolation-based method consistently assigned reasonable values to the anomalous and normal points, effectively identifying the anomalies. It is important to note that, in the isolation-based methods, no assumptions are made about the distribution of the data.

Compared to most other anomaly detection algorithm, the isolation-based methods are computationally very efficient and scalable, which is an important property for an anomaly detector, in particular when working with large data sets. Consider *i*Forest for example, it has been empirically proven that its detection performance, especially when the number of instances is larger than 1000, is superior to state-of-the-art anomaly detectors [11, 12]. With *i*Forest being a highly competitive and scalable anomaly detection algorithm, it is no surprise that, within the last years, it has quickly established itself well in the academic literature and has often become the anomaly detector of choice (see, e.g., [34])

We favor the *i*Forest over other anomaly detection methods as it is a scalable algorithm designed for the sole purpose of anomaly detection. Moreover, *i*Forest is publicly available through the well-known Python library for machine learning: `scikit-learn` [35, 36]. Since we make *i*Forest our anomaly detector of choice, we next describe the *i*Forest algorithm in greater detail.

2.3. Isolation forest

The isolation forest (*i*Forest) is an unsupervised, tree-based ensemble method that applies the novel concept of *isolation* to anomaly detection [11, 12]. Isolation refers to the separation of each instance from the rest. Solely the tree structures of the learned ensemble are required to generate anomaly scores, hence this method avoids calculating computationally expensive distance or density measures.

The base learning algorithm of *i*Forest is called isolation tree (*i*Tree), which constructs a *proper binary tree* in a completely random manner based on a subsample of size $\psi \in \mathbb{N}_{\geq 2}$ taken from the training data without replacement (proposed default value: $\psi = 256$). In a divide-and-conquer fashion, *i*Tree recursively splits the input space into progressively smaller, axis-parallel rectangles with the aim to isolate instances. Ideally, there remains only one instance in each leaf node. Given their properties of being few and different, anomalies are thereby more susceptible to isolation, and therefore tend to be closer to the root of an *i*Tree than normal instances (Fig. 2).

An *i*Tree node is created by randomly selecting an attribute along with a randomly drawn split value, which lies between the minimum and maximum of the selected attribute. Note that the application of *i*Trees is therefore only meaningful on numeric attributes. The fact that an *i*Tree is built in a random fashion on a subsample makes *i*Forest so computationally and memory efficient. When a test instance passes through an *i*Tree, at each non-leaf node, the respective attribute value is retrieved and tested against the split value in order to decide its traversal to either the left or right child node.

To determine the anomaly score for a given instance $\mathbf{x} \in \mathbb{R}^d$ with $d \in \mathbb{N}_1$ representing the number of attribute measurements, *i*Forest solely leverages the learned tree structures of the $n_{\text{trees}} \in \mathbb{N}_1$ *i*Trees. Since anomalies are more susceptible to isolation, an anomalous instance is expected to have a shorter path length than a normal instance when it traverses an *i*Tree from root to leaf. Given an isolation tree h_t, the path length $h_t(\mathbf{x}) = e + c(n_{\text{leaf}}) \in [1, \psi - 1]$ for instance $\mathbf{x}$ is derived by counting the number of edges $e \in \{1, 2, \ldots, \psi - 1\}$ from the root to the leaf node $\mathbf{x}$ falls into. To account for the possibility that the isolation of a set of instances at the leaf node did not fully succeed, the following adjustment is added to e as a function of the leaf node size $n_{\text{leaf}} \in \mathbb{N}_1$ [12, 37]:

$$c(n_{\text{leaf}}) = \begin{cases} 2H(n_{\text{leaf}} - 1) - 2(n_{\text{leaf}} - 1)/n_{\text{leaf}} & \text{if } n_{\text{leaf}} > 2, \\ 1 & \text{if } n_{\text{leaf}} = 2, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $H(\cdot)$ is the harmonic number that can be approximated by

$H(a) \approx \ln(a) + 0.5772156649$ (Euler's constant).

Since an *i*Tree is structurally equivalent to a Binary Search Tree (BST), the adjustment is derived from unsuccessful searches in BST and aims to account for the average path length of a random sub-tree that could be built given the leaf node size [11, 12]. The average path length of instance $\mathbf{x}$ can be computed by utilizing the collection of n_{trees} constructed *i*Trees:

$$E(h(\mathbf{x})) = \frac{1}{n_{\text{trees}}} \sum_{t=1}^{n_{\text{trees}}} h_t(\mathbf{x}), \tag{2}$$

where $h_t(\mathbf{x})$ is the path length of $\mathbf{x}$ derived from the tth isolation tree. Liu et al. [11, 12] empirically showed that already at a moderate ensemble size (proposed default value: $n_{\text{trees}} = 100$), the average path length stabilizes quickly and tends to be much lower for anomalous instances.

Finally, the anomaly score $s(\mathbf{x},\psi)$ for instance $\mathbf{x}$ can be computed as follows [11, 12]:

$$s(\mathbf{x}, \psi) = 2^{-\frac{E(h(\mathbf{x}))}{c(\psi)}}, \tag{3}$$

where $E(h(\mathbf{x}))$ is defined as in Eq.(2) and $c(\psi)$ serves as a normalization factor to make a suitable comparison of models with different subsample sizes ψ. The latter is regarded to be the average path length of traversing a random tree that was constructed based on a sample of size ψ [38]. The final mapping step in Eq.(3) ensures that the anomaly score lies in the interval (0,1), which allows for a more intuitive interpretation [11, 12]:

- If $E(h(\mathbf{x})) \longrightarrow 0, s \longrightarrow 1$. If the average path length of $\mathbf{x}$ is close to zero, the anomaly score will be close to one, hence $\mathbf{x}$ can be regarded as an anomalous instance.
- If $E(h(\mathbf{x})) \longrightarrow \psi - 1, s \longrightarrow 0$. If the average path length of $\mathbf{x}$ is close to the absolute maximum depth of a binary tree given ψ, the anomaly score will be close to zero, hence $\mathbf{x}$ can be regarded as a normal instance.
- If $E(h(\mathbf{x})) \longrightarrow c(\psi), s \longrightarrow 0.5$. If the average path length of $\mathbf{x}$ is close to the average path length of a random tree given ψ, the anomaly score will be close to 0.5. If all instances have an anomaly score close to 0.5, then there are no distinct anomalies in the data.

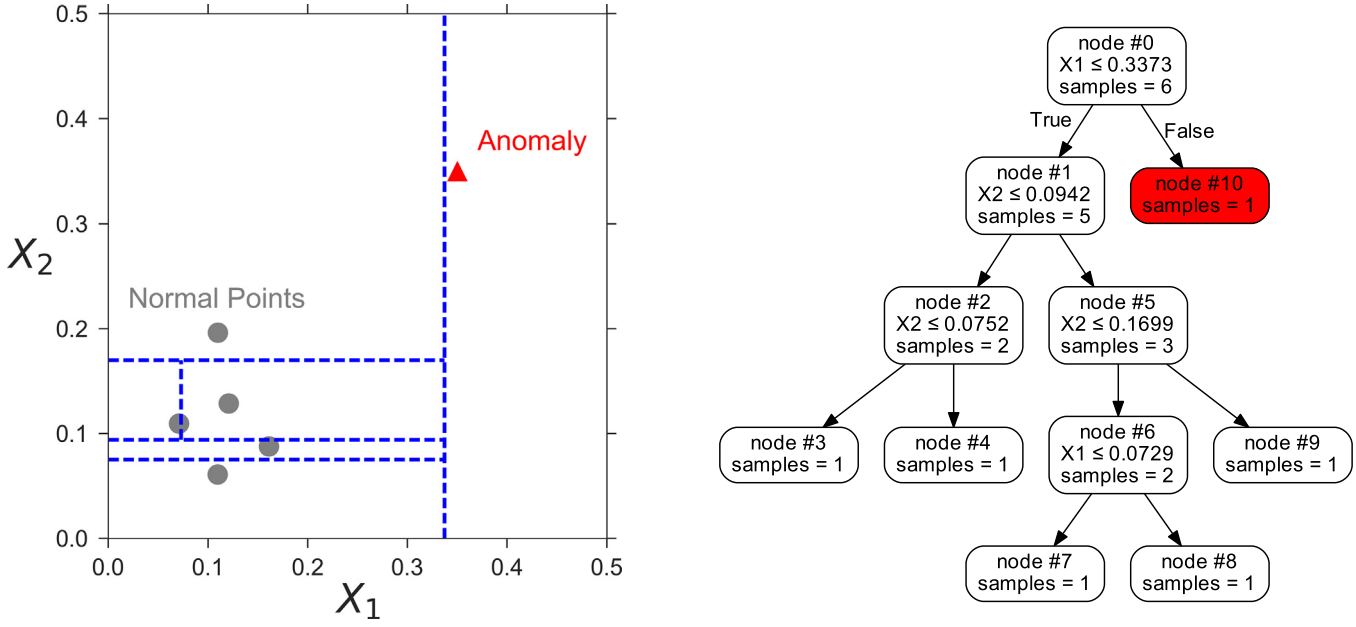

Fig. 2. Example of an isolation tree (*i*Tree) on fabricated data. The red triangle represents an anomalous data point. Blue dashed lines correspond to the data partitions made by the *i*Tree shown on the right. The anomaly (▲) falls into the leaf node directly under the root, thus it is separated (isolated) faster than the normal data points (●). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In this paper, we utilize the Python implementation of *i*Forest available in the `scikit-learn` library [35, 36], and, throughout the paper, anomaly scores are computed as follows:

$$s_{skl}(\mathbf{x}, \psi) = s(\mathbf{x}, \psi) - 0.5 = 2^{-\frac{E(h(\mathbf{x}))}{c(\psi)}} - 0.5. \quad (4)$$

Consequently, the anomaly scores become centered around zero with interval $(-0.5, 0.5)$. It still applies that an instance with a score close to the upper bound is regarded as an anomaly.

*i*Forest is a *nonparametric* anomaly detection method since it does not make any assumptions about the data distribution. Despite the very simple design, the *i*Forest algorithm is very competitive both in detection performance and time efficiency. The creators of *i*Forest showed that their algorithm outperforms several other state-of-the-art anomaly detectors on various real-world data sets. Regarding *i*Forest's scalability, complexity analysis performed by [12] revealed that its worst-case time complexities for training and evaluation are $O(\psi^2 n_{\text{trees}})$ and $O(n\psi n_{\text{trees}})$ respectively, which amounts to a total time complexity of $O(\psi(n + \psi)n_{\text{trees}})$. It is important to note that the training complexity does not involve any data-dependent variables, since the training complexity of *i*Forest solely depends on its invariant and known input parameters. The space complexity of the isolation forest equals $O(\psi n_{\text{trees}})$. In summary, *i*Forest is a very scalable algorithm, and even when dealing with large data sets, it possesses a low linear time complexity (i.e., $\psi^2 n_{\text{trees}} \ll n$) with a low memory requirement [12].

3. Methodology

3.1. Proposed approach: $iForest_{CAD}$

As fraud data sets typically contain mixed type attributes [7], it is often desired to include all available information in the construction of the analytical fraud detection model. The proposed $iForest_{CAD}$ approach aims to achieve this by combining information of nominal and numeric attributes into anomaly scores which in turn are leveraged in the construction of a fraud classifier.[1] In this respect, $iForest_{CAD}$ can be used for conditional anomaly detection and for the semi-automatic creation of new attributes. For the former, the scores produced by $iForest_{CAD}$ can be utilized for ranking observations and detecting anomalies. As for the latter, in machine learning, feature engineering is a difficult and *domain-specific* task that is often key for a successful data science application, and it is therefore considered to be more important than the choice of the classification model [9]. With $iForest_{CAD}$, new numeric attributes can be created that integrate expert knowledge into the predictive model. For this reason, expert knowledge drives the selection of attributes in $iForest_{CAD}$ in order to obtain scores that are meaningful to the domain experts. From a high-level perspective, the proposed approach performs the following steps:

1. **Selection**: Select nominal and numeric attributes that should undergo the transformation. The selection is driven by expert knowledge in order to obtain meaningful scores.
2. **Partitioning**: Determine all distinct combinations of values of selected nominal attributes and split the data set according to these combinations.
3. **Conditional anomaly detection**: Train *i*Forest on each data partition and compute the anomaly scores of all instances across all partitions.
4. **Classifier training**: Replace the selected attributes with the anomaly scores and train a binary classifier.

A more rigorous and formal description of the $iForest_{CAD}$ approach is given in the following paragraphs. Table 2 provides an overview of the mathematical notation.

3.1.1. Learning phase

3.1.1.1. Data definition. Suppose a sample of $n \in \mathbb{N}_{>1}$ instance-label pairs $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ is given with $y_i \in \{0,1\}$ being the observed class label of instance i and its mixed type attribute measurements are bundled in a d-tuple $\mathbf{x}_i = (x_{ij})_{\forall j \in J} = (x_{i1}, \ldots, x_{id})$, where $d \in \mathbb{N}_1$ is the number of attributes and $J = \{1, 2, \ldots, d\}$ is the index set over the attributes. Additionally, let $I = \{1, 2, \ldots, n\}$ be the index set over the observations. Note that x_{ij} denotes the value of the attribute j of observation i. Likewise, for example, $(x_{ij})_{\forall j \in \{1,2,3\}} = (x_{i1}, x_{i2}, x_{i3})$ would be a triple holding the values of the 1st, 2nd, and 3rd attribute of observation i.

[1] A software implementation of $iForest_{CAD}$ is available at https://github.com/estripling/iForestCAD.

Table 2
Overview of mathematical notation.

Symbol	Description
$\{a_r\}_{r\in R}$	Abbreviated notation for a set in which the elements a_r are indexed by r given some index set R. Similarly for a tuple: $(a_r)_{r\in R}$
d	Number of attributes, $d \in \mathbb{N}_1$
J	Index set over attributes, $J = \{1, 2, ..., d\}$
j	Attributes are indexed by j, where $j \in J$
n	Number of observations or instances, $n \in \mathbb{N}_{>1}$
I	Index set over observations, $I = \{1, 2, ..., n\}$
i	Observations are indexed by i, where $i \in I$
x_{ij}	Value of the jth attribute of observation i
$\mathbf{x}_i$	Observation or instance i: $\mathbf{x}_i = (x_{ij})_{\forall j\in J} = (x_{i1}, x_{i2}, ..., x_{id})$
y_i	Observed class label of observation i
$\hat{y}_i$	Predicted value of observation i
V_j	Value set of the jth attribute, $V_j = \{x_{ij}\}_{\forall i\in I}$ with $j \in J$
D	Data set: sample of n instance-label pairs, $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$
S	Set of indices of selected mixed type attributes, $\emptyset \neq S \subseteq J$: $S = S_{nom} \cup S_{num}$
S_{nom}	Set of indices of selected *nominal* attributes, $\emptyset \neq S_{nom} \subsetneqq S$
S_{num}	Set of indices of selected *numeric* attributes, $\emptyset \neq S_{num} \subsetneqq S$
K	Set of distinct combinations of values of selected nominal attributes
k	One distinct combination: distinct tuple of nominal attribute values, $k \in K$
$\mathscr{A}$	Binary classification learning algorithm

3.1.1.2. Selection. Attributes are selected by their indices which are contained in the index set $\emptyset \neq S \subseteq J$. The selection of attributes is driven by expert knowledge. S must comprise indices that correspond to nominal and numeric attributes, i.e., $S = S_{nom} \cup S_{num}$ and $S_{nom} \cap S_{num} = \emptyset$, where the two subsets $\emptyset \neq S_{nom} \subsetneqq S$ and $\emptyset \neq S_{num} \subsetneqq S$ hold the indices of the selected nominal and numeric attributes, respectively.

3.1.1.3. Partitioning. Denote $V_s = \{x_{is}\}_{\forall i\in I}$ as the value set of the attribute corresponding to index $s \in S$. The number of distinct values of attribute s is defined by the cardinality of its value set $|V_s|$. The set K of distinct combinations of values of selected nominal attributes is obtained by computing the Cartesian product. The outcome of the Cartesian product is a set of ordered $|S_{nom}|$-tuples. To formally define this operation, an additional index set $R = \{1, 2, ..., |S_{nom}|\}$ over the elements of S_{nom} is introduced, since S_{nom} is a set that can contain any elements of J. Hence, the Cartesian product of $|R|$ value sets is defined as

$$K = \prod_{r=1}^{|R|} V_{s_r} = V_{s_1} \times \cdots \times V_{s_{|R|}} = \{(v_1, ..., v_{|R|}) \mid (\forall r \in R)[s_r \in S_{nom} \wedge v_r \in V_{s_r}]\}. \quad (5)$$

The number of distinct combinations of values of selected nominal attributes is equal to the cardinality of K, i.e., $|K| = \prod_{s\in S_{nom}} |V_s|$. It is important to note that the number of distinct combinations must be smaller than the sample size, i.e., $|K| < n$, in order to have a sufficiently high number of instances in each partition when training the anomaly detector in the next step. As *i*Forest is the anomaly detector of choice, "sufficiently high" refers to its subsample size parameter ψ (a formal description is given in the next paragraph). If one or more selected nominal attributes have too many distinct values, binning or grouping of the nominal attribute values should be performed either through the use of categorization methods (see, e.g., [39, p. 60 ff.], [40, Chapter 9.2.4]) or under the guidance of expert knowledge.

The data set D is then split into nonempty, disjoint subsets according to all distinct combinations $k \in K$, and only the selected numeric attributes of the instances are elected. Formally, denote such subset as

$$D_k = \{(x_{is})_{\forall s\in S_{num}} \mid (\exists\, !k \in K)[k = (x_{is})_{\forall s\in S_{nom}}]\} \quad (\forall i \in I). \quad (6)$$

[2]

In other words, for a given distinct combination $k \in K$, D_k is a set of $|S_{num}|$-tuples, where each tuple contains the values of the selected numeric attributes, of those observations in the data set for which the combination of their values of the selected nominal attribute equals k. Moreover, let $\emptyset \neq I_k \subsetneqq I$ be the set of observation indices that are members of D_k for a given $k \in K$. Recall it is important for training the anomaly detector in the next step that each data partition D_k has a sufficiently high number of instances, i.e., $\forall\, k \in K$, $|D_k| \gtrsim \psi$, where ψ is the subsample size parameter of *i*Forest. If this condition is not met, a categorization method as mentioned previously must be applied. In any case, it must hold that $|D_k| > 0$ for all $k \in K$.

3.1.1.4. Conditional anomaly detection. Train *i*Forest on each data partition D_k to obtain $M = \{m_k = i\text{Forest}(D_k)\}_{\forall k\in K}$. Once all anomaly detectors are trained, compute the anomaly scores with the corresponding trained *i*Forest m_k:

$$A_k = \{a_i = m_k(t_i) \in \mathbb{R} \mid m_k \in M \wedge (\forall i \in I_k)[t_i = (x_{is})_{\forall s\in S_{num}}]\} \quad (\forall k \in K) \quad (7)$$

such that $\bigcup_{k\in K} A_k = \{a_i\}_{\forall i\in I}$.

3.1.1.5. Classifier training. Remove the values of the selected attributes, $(x_{is})_{\forall s\in S}$, from the observation tuple i and append it with the corresponding anomaly score a_i for all $i \in I$. In other words, reconstruct the data set such that

$$D' = \{(\mathbf{x}_i', y_i)\}_{i=1}^{n},$$

where

$$\mathbf{x}_i' = ((x_{ij})_{\forall j\in J'}, a_i)$$

with

$$J' = J \setminus S = \{j \in J \mid j \notin S\}$$

is a tuple with $|J'| + 1$ elements containing the values of attributes that were not selected in the first step and the conditional anomaly score of observation i. Note that, if $J' = \emptyset$, this results in an empty tuple, hence $\mathbf{x}_i' = ((), a_i) = (a_i)$ becomes a singleton containing only the anomaly score.

Finally, given a binary classification learning algorithm $\mathscr{A}$, train a classifier on the newly constructed data set: $g = \mathscr{A}(D')$.

3.1.2. Prediction phase

Next to model construction, it is also relevant to compute the predicted value for a given test instance $\mathbf{x}_{i'} = (x_{i'j})_{\forall j\in J}$. Note that the index i' is applied to indicate that the instance was *not* used for the construction of the models. To make a prediction with the *i*Forest$_{\text{CAD}}$ approach, the following steps are applied:

(i) identify the distinct combination $k \in K$ the test instance belongs to, $k = (x_{i's})_{\forall s\in S_{nom}}$, (ii) retrieve the corresponding trained *i*Forest $m_k \in M$ and compute the anomaly score $a_{i'} = m_k(t_{i'})$ with $t_{i'} = (x_{i's})_{\forall s\in S_{num}}$, (iii) remove the values of the nonselected attributes and append the anomaly score: $\mathbf{x}_{i'}' = ((x_{i'j})_{\forall j\in J'}, a_{i'})$, and finally (iv) compute the predicted value $\hat{y}_{i'}$ of the test instance using the trained classifier: $\hat{y}_{i'} = g(\mathbf{x}_{i'}')$.

Note that, depending on the classifier, the predicted value can either be an estimated class label or a score expressing the confidence of the classifier to assign the test instance to a class. Either way, the predicted values should be processed appropriately to obtain meaningful results.

[2] The general rule to check equality of two m-tuples is $(a_1, a_2, ..., a_m) = (b_1, b_2, ..., b_m)$ if and only if $a_1 = b_1, a_2 = b_2, ..., a_m = b_m$.

3.2. Considered designs for the incorporation of anomaly scores

There are two straightforward ways how the anomaly scores can be incorporated into the final classifier training step. The first way involves the removal of the selected attributes and the appending of the anomaly scores, as described in Section 3.1. The second way is to augment the data set with the anomaly scores without removing any attributes. Formally, the observation tuple i in the reconstructed data set is expressed as

$$\mathbf{x}_i^{'} = ((x_{ij})_{\forall j \in J}, a_i),$$

having $|J| + 1$ elements.

In the WC fraud detection use case, we tried both options and found no considerable difference in detection performance. We decided in favor of the first option to incorporate it into the final $i\text{Forest}_{\text{CAD}}$ approach, since it showed a higher appreciation among the PIs and seems to allow for an easier communication. This is mainly due to dimensionality reduction benefit, which requires to explain less numbers of attributes to the PIs. Additionally, what is important for data scientists, the dimensionality reduction speeds up the training time of all classification models.

3.3. Strengths and limitations of $i\text{Forest}_{CAD}$

The *i*Forest algorithm, as described in Section 2.3, is devised to *directly* model anomalies by means of using the novel concept of isolation. By design, *i*Forest is a *global* anomaly detector and is as such usually applied on the entire data set (*unconditional anomaly detection*).

The aim of the proposed $i\text{Forest}_{\text{CAD}}$ approach is to inject expert knowledge into the modeling process through the incorporation of information of nominal attributes meaningful to domain experts. This is achieved by splitting up the data set into meaningful partitions in order to perform anomaly detection conditionally on a set of instances that share the same nominal characteristics (*conditional anomaly detection*). Hence, anomalies are detected with respect to the given partition, where the respective instances form their own "group-specific baseline" or "reference group." This way allows detecting anomalies that otherwise would be concealed when performing anomaly detection on all instances. These "*hidden anomalies*" can particularly be interesting to domain experts since the reference groups are defined by the distinct combinations of values of nominal attributes that are meaningful to them. Thus, $i\text{Forest}_{\text{CAD}}$ integrates a mixture of nominal and numeric attributes in a meaningful way.

Performing the transformation according to $i\text{Forest}_{\text{CAD}}$ has the additional benefit to ease the interpretation. That is, when knowing an instance exhibits a high anomaly score, this instance is an anomaly within its respective reference group *without* having to know the reference group itself. In other words, for a given instance, information from a set of attributes is compressed into a single, meaningful score. These scores can be leveraged as an input attribute, as we demonstrate it for the detection of fraudulent WC claims. A byproduct of the information compression is the *dimensionality reduction*, meaning that instead of the original $|J|$ only $|J^{'}| + 1$ (with $|J^{'}| + 1 < |J|$) attributes are used for training the classifier.

A clear bottleneck of $i\text{Forest}_{\text{CAD}}$ is the computation of the Cartesian product in Eq.(5), as the number of resulting data partitions grows quickly with the distinct combinations of values of selected nominal attributes. Hence, the selection of nominal attributes for $i\text{Forest}_{\text{CAD}}$ may be limited. Blindly selecting all nominal attributes for $i\text{Forest}_{\text{CAD}}$—which is not recommended—most likely prohibits the algorithm from producing scores, since the Cartesian product becomes quickly very large and hence many data partitions likely have (near) zero instances. Because of this reason, it is recommended that the selection of attributes is driven by expert knowledge, and if necessary apply categorization methods as pointed out in the previous section. However, if it is meaningful to domain experts, $i\text{Forest}_{\text{CAD}}$ can be executed more than once on the same data set but based on different subsets of attributes.

Table 3
Representative sample from bivariate distributions for height and weight of men and women in the US.

	$j = 1$ Gender	$j = 2$ Weight [kg]	$j = 3$ Height [cm]
$i = 1$	Female	58.9	157.7
$i = 2$	Male	82.5	173.4
⋮	⋮	⋮	⋮
$i = 199$	Male	77.1	177.8
$i = 200$	Female	80.2	173.7

3.4. Proof of concept

In this subsection, the proposed $i\text{Forest}_{\text{CAD}}$ approach is showcased on an artificial yet realistic data set. In particular, we look at intuitive attributes that are typically found in a life insurance data set. Important factors for life insurance organizations to determine the rate class are height, weight, and gender of a person. Fortunately, research literature provides parameter estimates for data distributions, making the creation of artificial values of these three attributes straightforward. More specifically, a representative sample from accurate bivariate distributions for height and weight of men and women in the US is generated, where distribution parameters were inferred from a large population survey [41]. Note that [42] and[43], for example, also used the same generative model to produce such artificial data. Table 3 shows a glimpse of a realization of 200 randomly drawn instances each with three attribute measurements for gender (nominal), weight in kilograms (numeric), and height in centimeters (numeric).[3]

Clearly, in this demonstration, it is assumed that these three attributes are meaningful to users in the context of a life insurance business application.

Plotting the data reveals that the data distributions of male and female heavily overlap (Fig. 3a). For example, observation 156 (128) is the tallest woman (the shortest man), and may be considered as an interesting peculiarity to the user, yet when data are viewed in their entirety, this observation does not strike as a peculiarity since it is concealed by the data of the other gender. This is also evident in the anomaly scores produced by the *i*Forest trained in the conventional way, meaning that nominal attributes are discarded and no particular data partitioning is performed in the anomaly detection exercise (Fig. 3b).

However, in settings such as fraud, one desires to also include information of nominal attributes in a meaningful way with the aim to further improve the detection performance and present anomalies to users that might be more interesting to them. In particular, we demonstrate how $i\text{Forest}_{\text{CAD}}$ enables users to detect hidden anomalies leveraging information of the nominal gender attribute. To enable the detection of such instances, we carry out the first three steps of proposed $i\text{Forest}_{\text{CAD}}$ approach with $S_{nom} = \{1\}$ and $S_{num} = \{2,3\}$. Hence, there is only one relevant value set, namely $V_1 = \{\text{female,male}\}$. The Cartesian product then becomes a set of two 1-tuples: $K = \{(\text{female}), (\text{male})\}$, which results in two data partitions split according to female and male with $|D_{(\text{female})}| = 104$ and $|D_{(\text{male})}| = 96$. Next, conditional

[3] It should be noted that this demonstration focuses on the detection of hidden anomalies by performing step 1 to step 3 of the proposed $i\text{Forest}_{\text{CAD}}$ approach. Hence, it merely performs conditional anomaly detection. The final step of classifier training is omitted due to the fact that arbitrary assignment of fraud labels to instances is inevitable for this illustrative data set, ergo pointless to demonstrate it. Also, for a better comparison, the number of instances in every data partition will be smaller than *i*Forest's default value of $\psi = 256$. This merely means that *i*Forest does not apply sub-sampling and is fitted on the complete data (partition).

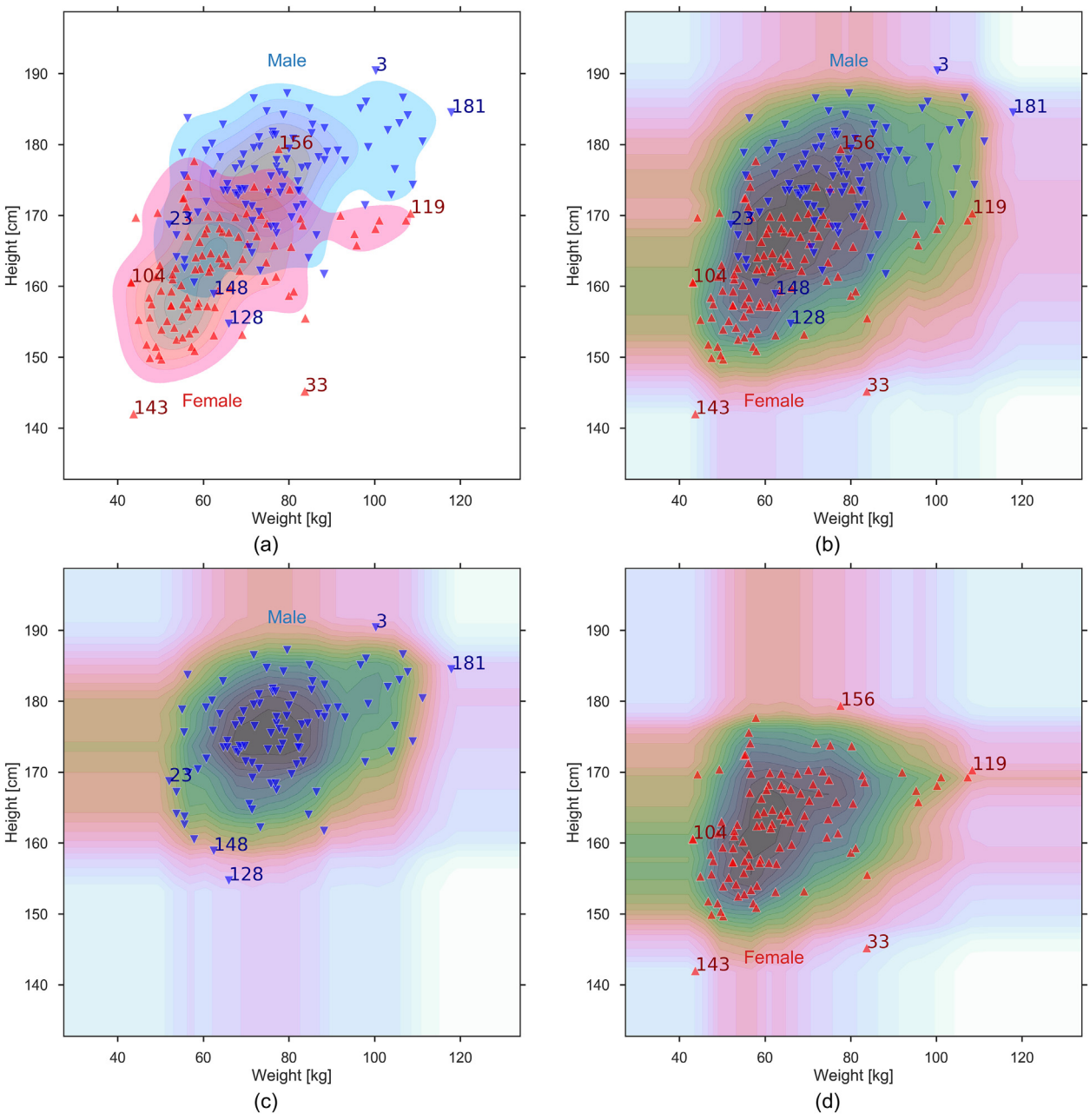

Fig. 3. Conditional anomaly detection. (a) Representative sample ($n = 200$) from bivariate distributions for height and weight of men (▼) and women (▲) in the US with superimposed contour lines of the density estimation for each gender. (b) Anomaly scores of *i*Forest trained according to the conventional approach. A lighter color implies a higher anomaly indication. (c) Anomaly scores of *i*Forest trained on data of men only ($|D_{(\text{male})}| = 96$). (d) Anomaly scores of *i*Forest trained on data of women only ($|D_{(\text{female})}| = 104$). The depicted numbers in the plots are the indices of the instances. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

anomaly detection is performed by training an *i*Forest on each data partition and examining the anomaly scores. Evidently, the emerged pattern deviates from the previous analysis, which now clearly identifies instance 156 (128) as an outlying observation (Fig. 3c and d).

The comparison of the anomaly scores, as well as the resulting ranks, further substantiates that the inclusion of nominal attributes, as in the $i\text{Forest}_{\text{CAD}}$ approach, is highly beneficial to detect anomalies which would otherwise remain undetected (Table 4). For example, observation 156 (128) has an anomaly score of − 0.0709 (− 0.0067) and receives a rank of 120 (41) with the regular *i*Forest. Thus, these observations would likely be regarded as normal rather than anomalous. On the other hand, the proposed $i\text{Forest}_{\text{CAD}}$ approach (anomaly scores in the gray area) shows strong indication that this observation is anomalous and moves its rank from 120 (41) upwards to 5 (3). This discrepancy in scores and ranks evidently demonstrates that such hidden anomalies would have likely remained undetected with the regular *i*Forest. Yet, it is also important to note that anomaly scores and ranks of the global anomalies stay more or less unchanged, see, for instance, observation 3 (the tallest man) and 143 (the shortest woman). Despite performing a conditional anomaly detection, the $i\text{Forest}_{\text{CAD}}$ anomaly scores continue to coherently reflect global anomalies, therefore one may argue that our proposed $i\text{Forest}_{\text{CAD}}$ approach extends the detection capabilities of the regular *i*Forest.

An alternative approach to incorporate nominal attributes into the anomaly detection is to establish an arbitrary ordering and map nominal values to numeric ones, as proposed by [44]. The authors called it

Table 4
Anomaly scores of selected instances.

		*i*Forest	*i*Forest$_{ext}$	Proposed *i*Forest$_{CAD}$ approach			
				*i*Forest$_{(male)}$	*i*Forest$_{(female)}$	Rank	
i	Gender	(n = 200)	(n = 200)	(n = 96)	(n = 104)	(n = 200)	Hidden anomaly
3	male	0.1310 (4)	0.0890 (5)	0.0887 (3)		(7)	
23	male	−0.0261 (58)	0.0292 (21)	0.0464 (7)		(13)	Yes
33	female	0.1478 (2)	0.1305 (2)		0.1408 (2)	(2)	
104	female	0.0366 (12)	0.0151 (28)		0.0068 (12)	(25)	
119	female	0.0414 (11)	0.0817 (7)		0.0893 (4)	(6)	
128	male	−0.0067 (41)	0.0930 (4)	0.1179 (1)		(3)	Yes
143	female	0.1786 (1)	0.1511 (1)		0.1564 (1)	(1)	
148	male	−0.0630 (101)	0.0383 (16)	0.0627 (4)		(8)	Yes
156	female	−0.0709 (120)	0.0833 (6)		0.0975 (3)	(5)	Yes
181	male	0.1318 (3)	0.1181 (3)	0.1075 (2)		(4)	

Note: Anomaly scores lie in [−0.5, 0.5]. An instance with a score close to the upper bound is regarded as an anomaly. The ranks of the instances are in parentheses.The *i*Forest$_{ext}$, an alternative approach proposed in the literature, is built on three numeric attributes, where "female" and "male" are mapped to 0 and 1, respectively. The gray area indicates the outcome of the proposed *i*Forest$_{CAD}$ approach. A large deviation in scores (ranks) evidently shows that the proposed approach identifies instances that would otherwise remain undetected (marked as hidden anomaly) with the regular *i*Forest.

the extended isolation forest, which we abbreviate as *i*Forest$_{ext}$. Thus, following this approach, the values "female" and "male" in our example may be mapped to the numeric values 0 and 1, respectively. The outcome of this analysis for selected instances is shown in the *i*Forest$_{ext}$ column of Table 4. It can be noted that this method assigns more or less similar ranks to most of the selected instances as *i*Forest$_{CAD}$, except for the two hidden anomalies $i = 23$ and $i = 148$ the ranks are almost twice as large.

Despite that an outcome is obtained similar to *i*Forest$_{CAD}$ on this particular data set, we argue that the *i*Forest$_{ext}$ method by [44] for incorporating nominal attributes is inadequate on the conceptional level. Recall that an attribute is chosen randomly at each node in the construction of an *i*Tree. Hence, when considering the current example, the information that data are generated from different distributions is not properly processed since the *i*Tree will make splits that are determined from both male and female data when a numeric attribute is selected. Thus, the extended isolation forest [44] can be viewed as a compromise between the *i*Forest (only numeric attributes) and *i*Forest$_{CAD}$ (strict distinction between nominal values). Due to the separate anomaly detection on nonoverlapping data partitions in *i*Forest$_{CAD}$, anomalies are detected strictly with respect to their own reference group; whereas the *i*Forest$_{ext}$ stochastically jumps across the mapped values of the nominal attribute and thus blurs the relationship to the reference group. Undoubtedly, this has an influence on the construction of *i*Trees, and hence on the resulting anomaly scores. An undesired consequence of such simple nominal-to-numeric mapping is that instances which are assigned the lowest or highest mapped value receive a higher anomaly score merely because of the value arrangement in space (Fig. 4). This is, of course, inadequate on the conceptual level since the mapping from nominal to numeric values is arbitrary.

Thus, we can conclude the proof of concept example with *i*Forest$_{CAD}$ enabling users to identify hidden anomalies by means of conditional anomaly detection, thereby processing nominal attributes in an adequate manner.

4. Case study: workers' compensation fraud detection

In this section, we empirically evaluate *i*Forest$_{CAD}$ on real-world WC claims received from a large European insurance organization. In particular, we describe the incorporation of nominal attributes according to *i*Forest$_{CAD}$ that is meaningful to the special investigation unit (SIU) in order to enhance the detection of fraudulent WC claims.

4.1. Workers' compensation insurance claim data

The data set consists of $n = 9572$ real-world WC insurance claims from 2011 to 2015 with $d = 23$ predictor attributes and a binary response variable indicating whether or not a claim is fraudulent (Table 5).

Due to confidentiality reasons, only three attributes are discussed that one would expect to find in a data set given the nature of the

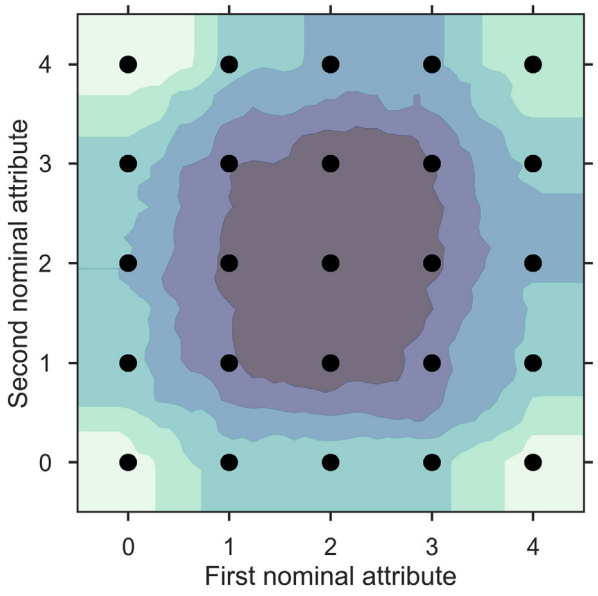

Fig. 4. Arbitrary mapping of values of two nominal attributes, each with five values, to numeric values {0,1,2,3,4}, where instances are uniformly distributed over the space. Instances at the corners receive a higher anomaly scores (lighter color) merely because of the spatial arrangement. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
WC insurance claim data.

Variable	Type	Description
X_1	Nominal	Type of injury
X_2	Nominal	Policyholder's industry sector
X_3	Numeric	Registered duration of incapacity
X_4–X_{23}	Numeric	Claim-related data, personal data, sociodemographic data
Y	Binary	Target variable

Note: Data set with $n = 9572$ real-world WC insurance claims (2011–2015) received from a large European insurance organization. Due to confidentiality reasons, detailed claim-related, personal, and sociodemographic data (i.e., X_4–X_{23}) are not disclosed.

insurance, i.e., type of injury the claimant sustained from the work accident (X_1), industry sector of the policyholder (X_2), and the duration of incapacity registered in the WC system (X_3).[4]

The former two are nominal attributes, where X_1 has values such as*fraction*and *concussion*, and X_2 exhibits values such as *construction*and *manufacturing*. The latter attribute, X_3, is a metric measure for the time period the claimant is declared to be incapable of resuming work (i.e., the estimated time required to recovery from the injury registered in the WC system). Needless to say, this is the time period in which the claimant receives WC benefits.

Discussions with the SIU revealed the challenging nature of proving a WC claim fraudulent. Definitive proof is required in order to prosecute fraudster in lengthy court proceedings. Hence, the SIU can only in a few number of cases be absolutely certain that a claim is fraudulent. Of course, the insurance company is highly interested in detecting and preventing fraud as early as possible. For this reason, claims with a high suspicion to be fraudulent have been assigned a fraud label by the SIU. Yet, the target variable Y still remains highly *unbalanced*. On top of this, the sheer amount of claims filed in a given time period makes it very challenging for the SIU to check each claim. One may be fairly certain that the assignment of fraud labels is nearly flawless, but this is not necessarily true for the assignment of non-fraud labels. In other words, it is possible that there is a number of claims in which fraudsters managed to stay undetected, and thus incorrect labels are assigned to those claims. We refer to this situation as *noise* in the fraud labels, which can similarly be observed in other lines of insurance (see, e.g., [45] and[46]).

4.2. Motivation for conditional anomaly detection

This subsection describes the first three steps of the proposed $i\text{Forest}_{\text{CAD}}$ approach in order to motivate the choices made under the guidance of expert knowledge. The accumulated knowledge of the private investigators (PIs) strongly suggests that the recovery time (i.e., X_3 in Table 5) is often a good indicator, in combination with other information, for suspicious behavior. In particular, it is believed that people working in some sectors are more inclined to perpetrate fraud than in others, as well as fraudsters unduly prolong the recovery period in which they receive WC benefits. This is in line with findings reported in the literature (see Section 2.1).

Hence, the task that poses upon the data scientist can be stated as follows: *Identify the WC claims that exhibit an abnormal recovery time given the injury type and sector in which the claimant performs his or her occupational duties.* This task can be broken down in several subtasks:

- Determine whether a given recovery time is anomalous in an *automatic*, *data-driven* manner.
- Take thereby into account the type of injury, since some injuries require a longer recovery time than others.
- Adjust for the injury type prevalence across the various sectors.

Our proposed $i\text{Forest}_{\text{CAD}}$ approach equips the data scientist with a methodology that allows him or her to successfully fulfill this task. More specifically, we carry out the first three steps of $i\text{Forest}_{\text{CAD}}$ with $S_{nom} = \{1,2\}$ and $S_{num} = \{3\}$ to perform conditional anomaly detection. This allows for a data-driven determination of whether a claim possesses an anomalous duration of incapacity given its reference groups (e.g., all claimants that reported a fraction and work in construction). Note that, under the guidance of expert knowledge, we regrouped the values of the nominal attributes, X_1 and X_2, such that the number of instances in each data partition is sufficiently high, i.e., $|D_k| \gtrsim \psi$ with $\psi = 256$ for all $k \in K$.

4.3. Preparation for WC fraud detection study

For this study, we consider the following common binary classification methods: logistic regression, decision tree (CART), random forest, SVM with linear kernel, and SVM with radial basis function (RBF) kernel. Data preparations are tailored to the specific classifier. That is, attributes are processed appropriately for machine learning methods such as SVM, which require standardized input (i.e., attributes with zero mean and unit variance). Weights inversely proportional to the class frequencies in the input data are assigned to address the class imbalance problem, as it can be activated for each classifier in the `scikit-learn` library.

Detection performance is measured by means of the area under the ROC curve (AUROC) and the area under the Precision-Recall curve (AUPR) through a stratified 10-fold cross-validation (CV) procedure. We made sure that the performances are evaluated on the exact same resamples for all classification models. For classifiers like SVM that require hyperparameter tuning, the stratified 10-fold CV procedure in combination with grid search is applied to find the optimal hyperparameter values in terms of the respective classification performance measure.

4.4. Results

Out of 20, 16 of the AUROC values are at the level of 80% or above (Table 6). There is a tendency that the black box models (i.e., random forest, linear SVM, and RBF SVM) possess a higher predictive power, yet the difference to the white box logistic model is marginal in each condition. For the given fraud data set, there is no clear indication that applying weighting helps to cope with the class imbalance. A clear pattern emerges when classifiers are trained with different attribute sets (marked as ① and ② in Table 6), where the difference between attribute set ① and ② is that the latter contains the conditional anomaly score attribute produced according to the proposed $i\text{Forest}_{\text{CAD}}$ approach. When trained with set ①, the AUROC performance is higher within the classifiers compared to when trained with attribute set ②. For the latter, the numeric attribute holding the $i\text{Forest}_{\text{CAD}}$ anomaly scores is identified to have the highest discriminative power according to all classifiers that inherently provide indication for variable importance.

A similar result is obtained when considering the AUPR performance (Table 7), except that RBF SVM's performance is practically constant no matter under what condition it is trained. It is also interesting that CART is consistently the worst model in terms of AUROC, yet, at the same time, it is the second best model in terms of AUPR in three out of four cases. However, as it can be expected from an unstable learner, CART has the largest standard error in all scenarios.

4.5. Discussion

The AUROC values presented in Table 6 are relatively high for most classifiers, indicating a good detection performance of fraudulent WC claims. However, mere AUROC performance should not be the only

[4] Note that, for the same reason, the presentation of results in subsequent subsections is limited.

Table 6
Cross-validated AUROC performances.

Classifier	① Without iForest$_{CAD}$ anomaly scores		② With iForest$_{CAD}$ anomaly scores	
	Without weights	With weights	Without weights	With weights
Logistic	*0.8766* (0.0225)	0.8612 (0.0233)	*0.8068* (0.0199)	0.8030 (0.0190)
CART	0.7569 (0.0409)	0.8019 (0.0358)	0.7237 (0.0233)	0.6305 (0.0428)
Random forest	0.8705 (0.0232)	**0.8725** (0.0188)	0.8027 (0.0213)	*0.8100* (0.0158)
Linear SVM	**0.8772** (0.0225)	0.8584 (0.0235)	**0.8075** (0.0192)	0.8038 (0.0190)
RBF SVM	0.8375 (0.0230)	*0.8721* (0.0204)	0.7798 (0.0174)	**0.8174** (0.0175)

Note: Average AUROC performances (standard errors in parentheses) computed based on the stratified 10-fold CV procedure. Two sets of attributes are used to train the classifiers: ① corresponds to the set in which no attribute transformation is performed according to the proposed iForest$_{CAD}$ approach; whereas ② corresponds to the set in which it is performed. Weighting is used to cope with the class imbalance problem, where weights are inversely proportional to the class frequencies in the input data. A bold (italic) number indicates the best (second best) performance within a condition.

Table 7
Cross-validated AUPR performances.

Classifier	① Without iForest$_{CAD}$ anomaly scores		② With iForest$_{CAD}$ anomaly scores	
	Without weights	With weights	Without weights	With weights
Logistic	0.1286 (0.0354)	0.1245 (0.0341)	0.0361 (0.0095)	0.0388 (0.0094)
CART	0.1338 (0.0394)	*0.2616* (0.0466)	*0.0671* (0.0265)	*0.0956* (0.0295)
Random forest	*0.1451* (0.0356)	0.1166 (0.0360)	0.0313 (0.0061)	0.0489 (0.0181)
Linear SVM	0.1290 (0.0360)	0.1236 (0.0341)	0.0365 (0.0097)	0.0382 (0.0089)
RBF SVM	**0.5032** (0.0001)	**0.5038** (0.0001)	**0.5031** (0.0001)	**0.5031** (0.0001)

Note: Average AUPR performances (standard errors in parentheses) computed based on the stratified 10-fold CV procedure. Two sets of attributes are used to train the classifiers: ① corresponds to the set in which no attribute transformation is performed according to the proposed iForest$_{CAD}$ approach; whereas ② corresponds to the set in which it is performed. Weighting is used to cope with the class imbalance problem, where weights are inversely proportional to the class frequencies in the input data. A bold (italic) number indicates the best (second best) performance within a condition.

evaluation criterion to assess the fraud detection approach. Other evaluation criteria are, for example, the ease of interpretation and the acceptance of the modeling approach by stakeholders. These criteria are less straightforward to quantify numerically.

Close collaboration with the insurer's SIU showed that the proposed iForest$_{CAD}$ approach finds a higher appreciation among the PIs. That is mainly because of the core idea of detecting anomalous behavior within reference groups that are meaningful and interesting to them. The iForest$_{CAD}$ approach was ultimately validated in a practical setting by using the elected classifier to predict fraudulent WC claims. The predictions were in turn evaluated by the PIs to assess the quality of the fraud leads. No detailed information can be revealed about the exact performance, but a large proportion of previously undetected, suspicious claims were identified. Additionally, the study outcome confirmed that the fraud labels are indeed noisy. That is, some WC claims managed to stay undetected and thus were assigned the incorrect label of non-fraud.

To relate back to the results in Table 6, an explanation of the lower AUROC performance of iForest$_{CAD}$ is likely due to the different ranking result. Note that the statistical interpretation of the AUROC is as follows [47]: "the [area under the ROC curve] of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance." In this study, a positive instance is a WC claim with a fraud label. The cornerstone of the iForest$_{CAD}$ approach is the creation of a new attribute that assigns more granular anomaly scores to instances which consequently affects the inner construction mechanisms of the classifiers. As demonstrated in the proof of concept example, observations marked as hidden anomalies have very different anomaly scores. When taken the iForest$_{CAD}$ anomaly scores as input and keeping the noise in the fraud labels in mind, it is likely that more claims with a non-fraud label, but are intrinsically suspicious or anomalous, are ranked higher by the classifier. As a result, the classification performance receives a lower AUROC value, because for those claims the incorrect label is assigned.

As for the AUPR, it represents the average precision of a classifier across different classification thresholds. The advantage of using AUPR over common measures such as precision, recall, and the *F*-measure is that it is a threshold independent performance measure (like AUROC). With the AUPR performance in Table 7, we observe a similar performance decrease as with the AUROC due to the same reason of claim ranking and noisy fraud labels. However, as confirmed by the SIU, the practical application of iForest$_{CAD}$ exhibits a high detection performance of suspicious claims that previously remained undetected, which contributes to the novelty criterion defined in Section 1. Recall that iForest is an unsupervised anomaly detection algorithm, meaning that it does not require label information for model construction. Thus, our proposed iForest$_{CAD}$ approach has a built-in unsupervised component combined with powerful supervised classification techniques. The combination of both supervised and unsupervised learning concepts provides an explanation for the high fraud detection rate when the proposed approach was put into practice.

5. Conclusions and future work

In this paper, we presented a case study on WC insurance fraud in which we analyzed real-world claims received from a large European insurance organization. Accumulated expert knowledge strongly suggests that disproportional duration of incapacity is often a strong indicator for fraudulent behavior. It is evident that the recovery time depends on the type of injury, but also that the prevalence of certain injuries which varies across the industry sectors.

We presented an isolation-based conditional anomaly detection approach, iForest$_{CAD}$, that integrates expert knowledge by processing information stored in nominal attributes in a meaningful way. In particular, the proposed approach allows zooming in to specific subgroups (e.g., people working in construction and sustained a fraction) and apply iForest to detect abnormally long durations conditional on a group of people that share the same (nominal) characteristics. The returned iForest$_{CAD}$ anomaly scores of our proposed approach permit a straightforward interpretation that is easy to communicate to stakeholders. By using the iForest$_{CAD}$ anomaly scores as input for training a binary classifier, we consistently found that the score attribute was selected to be the most important predictor variable for the detection of fraudulent WC claims. The usage of powerful supervised classification models resulted in high AUROC values of at least 80% for most classifiers. Additionally, in Section 3.4, we demonstrated that our proposed iForest$_{CAD}$ approach is capable of detecting hidden anomalies.

In the collaboration with the insurer's SIU, it has been confirmed that the fraud labels are noisy, meaning that fraudulent claims were assigned a non-fraud label. However, with the aid of the iForest$_{CAD}$ anomaly scores, these claims were ranked higher on the suspicion list to be fraudulent. As it turned out in the case study, an acceptable large

proportion of claims proved to be new and interesting to the PIs.

Other data science projects might not have the luxury of an easy access to supporting domain experts. For future work, we therefore intend to develop strategies for the automatic selection of attributes, create appropriate categorization strategies, as well as investigate suitable data clustering/partitioning methods for $i\text{Forest}_{\text{CAD}}$. Once such automatic mechanisms are in place, we plan to perform a complexity analysis and study the scalability aspects of the algorithm.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Coalition Against Insurance Fraud, Insurance Fraud Backgrounder: Why is Fraud so Big? (2017) Retrieved from http://www.insurancefraud.org/fraud-backgrounder.htm [Online; accessed May 2017].
[2] R.A. Derrig, Insurance fraud, Journal of Risk and Insurance 69 (3) (2002) 271–287.
[3] Insurance Information Institute, Workers Compensation, (2016) Retrieved from http://www.iii.org/issue-update/workers-compensation [Online; accessed March 2017].
[4] S. Viaene, G. Dedene, Insurance fraud: issues and challenges, The Geneva Papers on Risk and Insurance 29 (2) (2004) 313–333.
[5] R.A. Derrig, L.K. Krauss, First steps to fight workers' compensation fraud, Journal of Insurance Regulation 12 (3) (1994) 390–415.
[6] Insurance Information Institute, Insurance Fraud, (2016) Retrieved from http://www.iii.org/issue-update/insurance-fraud [Online; accessed September 2016].
[7] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Computing Surveys (CSUR) 41 (3) (2009) 15:1–15:58, http://dx.doi.org/10.1145/1541880.1541882.
[8] A.P. Sinha, H. Zhao, Incorporating domain knowledge into data mining classifiers: an application in indirect lending, Decision Support Systems 46 (1) (2008) 287–299, http://dx.doi.org/10.1016/j.dss.2008.06.013.
[9] P. Domingos, A few useful things to know about machine learning, Communications of the ACM 55 (10) (2012) 78–87, http://dx.doi.org/10.1145/2347736.2347755.
[10] K. Coussement, D.F. Benoit, M. Antioco, A Bayesian approach for incorporating expert opinions into decision support systems: a case study of online consumer-satisfaction detection, Decision Support Systems 79 (2015) 24–32, http://dx.doi.org/10.1016/j.dss.2015.07.006.
[11] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation forest, Proceedings of the Eighth IEEE International Conference on Data Mining (ICDM'08), IEEE, 2008, pp. 413–422.
[12] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation-based anomaly detection, ACM Transactions on Knowledge Discovery from Data (TKDD) 6 (1) (2012) 3:1–3:39, http://dx.doi.org/10.1145/2133360.2133363.
[13] J. Biddle, K. Roberts, Claiming behavior in workers' compensation, Journal of Risk and Insurance 70 (4) (2003) 759–780.
[14] R.J. Butler, H.H. Gardner, N.L. Kleinman, Workers' compensation: occupational injury insurance's influence on the workplace, Handbook of Insurance, Springer, 2013, pp. 449–469.
[15] A.B. Krueger, Incentive effects of workers' compensation insurance, Journal of Public Economics 41 (1) (1990) 73–99.
[16] B.D. Meyer, W.K. Viscusi, D.L. Durbin, Workers' compensation and injury duration: evidence from a natural experiment, The American Economic Review (1995) 322–340.
[17] A. Cheadle, G. Franklin, C. Wolfhagen, J. Savarino, P.Y. Liu, C. Salley, M. Weaver, Factors influencing the duration of work-related disability: a population-based study of Washington State workers' compensation, American Journal of Public Health 84 (2) (1994) 190–196.
[18] R.J. Butler, M.L. Baldwin, W.G. Johnson, The effects of worker heterogeneity on duration dependence: low-back claims in workers compensation, Review of Economics and Statistics 83 (4) (2001) 708–716.
[19] D. Bolduc, B. Fortin, F. Labrecque, P. Lanoie, Workers' compensation, moral hazard and the composition of workplace injuries, Journal of Human Resources 37 (3) (2002) 623–652.
[20] L. Lin, P.J.-H. Hu, O.R.L. Sheng, A decision support system for lower back pain diagnosis: uncertainty management and clinical evaluations, Decision Support System 42 (2) (2006) 1152–1169, http://dx.doi.org/10.1016/j.dss.2005.10.007.
[21] D.M. Hawkins, Identification of Outliers, 11 Springer, 1980.
[22] M. Markou, S. Singh, Novelty detection: a review–part 1: statistical approaches, Signal Processing 83 (12) (2003) 2481–2497, http://dx.doi.org/10.1016/j.sigpro.2003.07.018.
[23] M. Markou, S. Singh, Novelty detection: a review–part 2: neural network based approaches, Signal Processing 83 (12) (2003) 2499–2521, http://dx.doi.org/10.1016/j.sigpro.2003.07.019.
[24] V.J. Hodge, J. Austin, A survey of outlier detection methodologies, Artificial Intelligence Review 22 (2) (2004) 85–126, http://dx.doi.org/10.1007/s10462-004-4304-y.
[25] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection for discrete sequences: a survey, IEEE Transactions on Knowledge and Data Engineering 24 (5) (2012) 823–839, http://dx.doi.org/10.1109/TKDE.2010.235.
[26] A. Zimek, E. Schubert, H.-P. Kriegel, A survey on unsupervised outlier detection in high-dimensional numerical data, Statistical Analysis and Data Mining 5 (5) (2012) 363–387, http://dx.doi.org/10.1002/sam.11161.
[27] M. Gupta, J. Gao, C.C. Aggarwal, J. Han, Outlier detection for temporal data: a survey, IEEE Transactions on Knowledge and Data Engineering 26 (9) (2014) 2250–2267, http://dx.doi.org/10.1109/TKDE.2013.184.
[28] E. Schubert, A. Zimek, H.-P. Kriegel, Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection, Data Mining and Knowledge Discovery 28 (1) (2014) 190–237, http://dx.doi.org/10.1007/s10618-012-0300-z.
[29] M.A. Pimentel, D.A. Clifton, L. Clifton, L. Tarassenko, A review of novelty detection, Signal Processing 99 (2014) 215–249, http://dx.doi.org/10.1016/j.sigpro.2013.12.026.
[30] S. Agrawal, J. Agrawal, Survey on anomaly detection using data mining techniques, 19th International Conference in Knowledge Based and Intelligent Information and Engineering Systems, Elsevier, 2015, pp. 708–713, , http://dx.doi.org/10.1016/j.procs.2015.08.220.
[31] L. Akoglu, H. Tong, D. Koutra, Graph based anomaly detection and description: a survey, Data Mining and Knowledge Discovery 29 (3) (2015) 626–688, http://dx.doi.org/10.1007/s10618-014-0365-y.
[32] M. Ahmed, A.N. Mahmood, M.R. Islam, A survey of anomaly detection techniques in financial domain, Future Generation Computer Systems 55 (2016) 278–288, http://dx.doi.org/10.1016/j.future.2015.01.001.
[33] C.C. Aggarwal, Outlier Analysis, Springer, 2017, http://dx.doi.org/10.1007/978-3-319-47578-3.
[34] R.N. Calheiros, K. Ramamohanarao, R. Buyya, C. Leckie, S. Versteeg, On the effectiveness of isolation-based anomaly detection in cloud data centers, Concurrency and Computation: Practice and Experience 29 (18) (2017), http://dx.doi.org/10.1002/cpe.4169.
[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. VanderPlas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, Journal of Machine Learning Research 12 (Oct) (2011) 2825–2830.
[36] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A.C. Müller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, G. Varoquaux, API design for machine learning software: experiences from the Scikit-learn project, CoRR (2013) abs/1309.0238.
[37] B.R. Preiss, Data Structures and Algorithms With Object-oriented Design Patterns in Java, Wiley, 1999.
[38] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, 1st Ed., CRC press, 2012.
[39] B. Baesens, V. Van Vlasselaer, W. Verbeke, Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection, First, John Wiley & Sons, 2015.
[40] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed., Springer Series in Statistics, Springer New York, 2009.
[41] J. Brainard, D.E. Burmaster, Bivariate distributions for height and weight of men and women in the United States, Risk Analysis 12 (2) (1992) 267–275.
[42] J.K. Kruschke, Doing Bayesian Data Analysis: A Tutorial With R and BUGS, 1st ed., Academic Press, 2011.
[43] J.K. Kruschke, Doing Bayesian Data Analysis: A Tutorial With R, JAGS, and Stan, 2nd ed., Academic Press, 2014.
[44] L. Sun, S. Versteeg, S. Boztas, A. Rao, Detecting anomalous user behavior using an extended isolation Forest algorithm: an enterprise case study, CoRR (2016) abs/1609.06676.
[45] M. Artís, M. Ayuso, M. Guillén, Detection of automobile insurance fraud with discrete choice models and misclassified claims, Journal of Risk and Insurance 69 (3) (2002) 325–340.
[46] S.B. Caudill, M. Ayuso, M. Guillén, Fraud detection using a multinomial logit model with missing information, Journal of Risk and Insurance 72 (4) (2005) 539–550.
[47] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27 (8) (2006) 861–874, http://dx.doi.org/10.1016/j.patrec.2005.10.010.
[48] E.M. Knorr, R.T. Ng, Algorithms for mining distance-based outliers in large datasets, Proceedings of the 24th International Conference on Very Large Data Bases, Citeseer, 1998, pp. 392–403.
[49] S. Ramaswamy, R. Rastogi, K. Shim, Efficient algorithms for mining outliers from large data sets, Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 29 ACM, New York, NY, USA, 2000, pp. 427–438, , http://dx.doi.org/10.1145/342009.335437.
[50] S.D. Bay, M. Schwabacher, Mining distance-based outliers in near linear time with randomization and a simple pruning rule, Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 2003, pp. 29–38, , http://dx.doi.org/10.1145/956750.956758.
[51] M.E. Otey, A. Ghoting, S. Parthasarathy, Fast distributed outlier detection in mixed-attribute data sets, Data Mining and Knowledge Discovery 12 (2–3) (2006) 203–228.
[52] M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 29 ACM, New York, NY, USA, 2000, pp. 93–104, , http://dx.doi.org/10.1145/342009.335388.
[53] J. Tang, Z. Chen, A.W.-C. Fu, D.W. Cheung, Enhancing effectiveness of outlier detections for low density patterns, Advances in Knowledge Discovery and Data Mining, Springer, 2002, pp. 535–548.

[54] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996, pp. 226–231.
[55] E. Schubert, J. Sander, M. Ester, H.-P. Kriegel, X. Xu, DBSCAN revisited, revisited: why and how you should (still) use DBSCAN, ACM Transactions on Database Systems 42 (3) (2017) 19:1–19:21, http://dx.doi.org/10.1145/3068335.
[56] R.J. Campello, D. Moulavi, J. Sander, Density-based clustering based on hierarchical density estimates, Advances in Knowledge Discovery and Data Mining, Springer, 2013, pp. 160–172.
[57] R.J. Campello, D. Moulavi, A. Zimek, J. Sander, Hierarchical density estimates for data clustering, visualization, and outlier detection, ACM Transactions on Knowledge Discovery from Data 10 (1) (2015) 5:1–5:51, http://dx.doi.org/10.1145/2733381.
[58] B. Schölkopf, J.C. Platt, J.C. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional distribution, Neural Computation 13 (7) (2001) 1443–1471, http://dx.doi.org/10.1162/089976601750264965.
[59] D.M.J. Tax, R.P.W. Duin, Support vector data description, Machine Learning 54 (1) (2004) 45–66.
[60] T. Shi, S. Horvath, Unsupervised learning with random Forest predictors, Journal of Computational and Graphical Statistics 15 (1) (2006) 118–138, http://dx.doi.org/10.1198/106186006X94072.
[61] M. Schneider, W. Ertel, G. Palm, Expected similarity estimation for large scale anomaly detection, 2015 International Joint Conference on Neural Networks, IEEE, 2015, pp. 1–8, , http://dx.doi.org/10.1109/IJCNN.2015.7280331.
[62] M. Schneider, Expected Similarity Estimation for Large-scale Anomaly Detection, Ph.D. Thesis University of Ulm, Germany, 2017http://nbn-resolving.de/urn:nbn:de:bsz:289-oparu-4261-2.
[63] X. Song, M. Wu, C. Jermaine, S. Ranka, Conditional anomaly detection, IEEE Transactions on Knowledge and Data Engineering 19 (5) (2007) 631–645, http://dx.doi.org/10.1109/TKDE.2007.1009.
[64] F.T. Liu, K.M. Ting, Z.-H. Zhou, On detecting clustered anomalies using SCiForest, Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD'10), Springer, 2010, pp. 274–290.
[65] T.R. Bandaragoda, K.M. Ting, D. Albrecht, F.T. Liu, J.R. Wells, Efficient anomaly detection by isolation using nearest neighbour ensemble, Proceedings of the 2014 IEEE International Conference on Data Mining Workshop (ICDMW), IEEE, 2014, pp. 698–705.

Eugen Stripling is a PhD researcher at the Department of Decision Sciences and Information Management, KU Leuven (Belgium). His PhD research focuses on advanced methods for business-oriented data analytics. He obtained an MSc degree in Information Management at the Faculty of Economics and Business of KU Leuven, and an MSc degree in Business Statistics at the Leuven Statistics Research Centre.

Bart Baesens is a professor at KU Leuven (Belgium), and a lecturer at the University of Southampton (United Kingdom). He has done extensive research on big data & analytics, customer relationship management, web analytics, fraud detection, and credit risk management. See www.dataminingapps.com for an overview of his research.

Barak Chizi graduated in 1996 as an industrial and management engineer in the Technion - Israel Institute of Technology. He further specialized in machine learning and data mining (MSc and PhD in Tel-Aviv University), and since then teaches this subject at Tel-Aviv University and In Ben-Gurion University. In 2003 he became a data specialist for the Israeli government. In parallel, he started his own consulting firm. In 2011 he was appointed as Senior R&D director and Senior Researcher at Deutsche Telekom. In May 2015, he joined KBC and from August that year he started his role as General Manager for Big Data, Analytics and AI @KBC Group N.V.

Seppe vanden Broucke is an assistant professor at the Faculty of Economics and Business, KU Leuven, Belgium. His research interests include business data mining and analytics, machine learning, process management, and process mining. His work has been published in well-known international journals and presented at top conferences. Seppe's teaching includes Advanced Analytics, Big Data and Information Management courses. He also frequently teaches for industry and business audiences.