From One-Class to Two-Class Classification by Incorporating Expert Knowledge: Novelty Detection in Human Behaviour

Dieter Oosterlinck[a], Dries F. Benoit[a], Philippe Baecke[b]

[a] *Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2, B-9000 Ghent, Belgium*
[b] *Area Marketing, Vlerick Business School, Reep 1, B-9000 Ghent, Belgium*

Abstract

One-class classification is the standard procedure for novelty detection. Novelty detection aims to identify observations that deviate from a determined normal behaviour. Only instances of one class are known, whereas so called novelties are unlabelled. Traditional novelty detection applies methods from the field of outlier detection. These standard one-class classification approaches have limited performance in many real business cases. The traditional techniques are mainly developed for industrial problems such as machine condition monitoring. When applying these to human behaviour, the performance drops significantly. This paper proposes a method that improves existing approaches by creating semi-synthetic novelties in order to have labelled data for the two classes. Expert knowledge is incorporated in the initial phase of this data generation process. The method was deployed on a real-life test case where the goal was to detect fraudulent subscriptions to a telecom family plan. This research demonstrates that the two-class expert model outperforms a one-class model on the semi-synthetic dataset. In a next step the model was validated on a real dataset. A fraud detection team of the company manually checked the top predicted novelties. The results show that incorporating expert knowledge to transform a one-class problem into a two-class problem is a valuable method.

Keywords: Analytics, One-class classification, Novelty detection, Expert knowledge, Decision support systems

Email addresses: `dieter.oosterlinck@ugent.be` (Dieter Oosterlinck), `dries.benoit@ugent.be` (Dries F. Benoit), `philippe.baecke@vlerick.com` (Philippe Baecke)

1. Introduction

Novelty detection is concerned with detecting data that is different from the known data that characterizes a normal or stable situation. The term novelty detection is frequently used interchangeably with the more narrow term one-class classification. Models in this domain are used when only one class is known, while the other class is absent, poorly sampled or not well defined (Khan and Madden, 2014). One-class models rely heavily on outlier assumptions. These methods are therefore suited for applications with clear outliers, where the novelties do not interfere with the normal data. In machine monitoring for example, where the normal data is stable and outliers are usually pronounced, these methods are applicable (Japkowicz et al., 1995). However, applications that classify human behaviour typically possess a much higher variability in the data, resulting into a less strict boundary between novelties and the normal data. Novelties are not always outliers and outliers are not always novelties (Das et al., 2016). To deal with this increased classification difficulty, we need a method that uses more than solely the data of the one class of normal behaviour. As one-class classification is a harder problem than two-class classification (Tax and Duin, 2001), there is value to be found in the transformation of the one-class problem into two-class. Previous research developed methods to generate artificial data for the unknown class. As will be shown in this study, this generated data is however not informative enough to effectively boost performance in applications with diverse human behaviour. This research therefore proposes a method that incorporates expert knowledge to generate data for the unknown class. Modelling human behaviour with the support of human experts proves to be a good match.

The methodology is evaluated through a case study with a large European telecommunications provider. The company released a new mobile offering where customers can bundle themselves in a so called *family plan* (Desai et al., 2018). Due to unique factors of this product, people could take advantage of the service by using it in a way that is not allowed by general terms and conditions. This *subscription fraud* leads to losses in revenue. As fraud detection is becoming more and more important in preventing these losses (Barse et al., 2003), the goal was to develop a state-of-the-art fraud detection system (FDS) that distinguishes normal users from fraudsters. Before the launch of the new product, all data contains only non-fraudulent customers by definition. Once the product is launched, fraudulent cases will occur in the dataset, however those are unidentified and unlabelled, which rules out traditional two-class classification. The *post*-launch dataset will be used to validate the proposed model.

In the remainder of the paper, the traditional novelty detection approaches and their extensions are reviewed. In the methodology section, our expert method for the transformation into a two-class problem is developed. This method is benchmarked against other methods in the subscription fraud case study and the results are empirically validated by means of manual inspection on the *post*-launch data.

2. Literature Review

2.1. Novelty Detection

Novelty detection is a major area of research. There are several closely related fields of which many are used as synonyms; one-class classification, anomaly detection, outlier detection, concept learning, data description, single-class classification (El-Yaniv and Nisenson, 2007). Less commonly used are the terms noise detection, deviation detection and exception mining (Hodge and Austin, 2004). The terms novelty and anomaly detection are broader in scope than the often as a synonym used one-class classification (OCC). OCC, as introduced by Moya et al. (1993), is merely one approach to tackle novelty detection. However, it is the most common approach in this respect.

Novelty detection is concerned with classifying data that differ in a certain way from the available data in the training phase (Pimentel et al., 2014). Figure 1 displays a simplified representation of the concept. Cases from only one class are known, this class is referred to as target or positive class, while the unknown class is referred to as unstable, negative or outlier class (Hempstalk et al., 2008; Khan and Madden, 2014; Clifton et al., 2014). Throughout this research, the terms positive and negative class will be used.

One-class classification builds a model that describes the positive class, also called a model of stability (Clifton et al., 2014). This model only uses data from the known positive class. At prediction time, the model classifies new examples as a novelty or as being part of the positive class. Different types for these models will be discussed in 2.2.

Ding et al. (2014) explain that novelty detection is mostly based solely on the positive data since there is enough data about normal events, but none or only scarce data about non-normal events. Furthermore, it is often costly to acquire data about abnormal events. In these situations, it is general practice to fall back

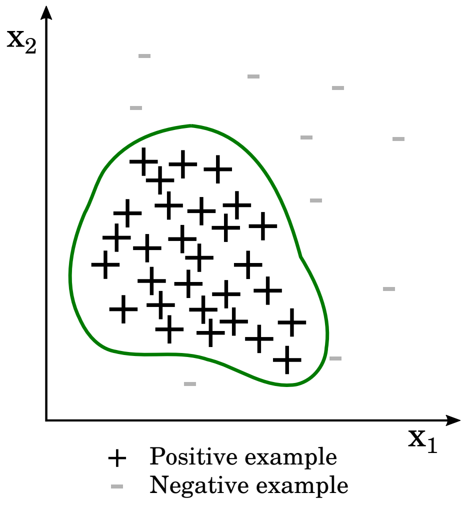


Figure 1: *Novelty detection.* Simplified representation with only two dimensions and perfect class separation. Only instances for the positive class are known. Data for the negative class is not (readily) available or unlabelled. Typically, a one-class model is employed (cf. the line surrounding the positive cases).

to these one-class, unsupervised approaches since developing explicit models for the novelty class is hard (Das et al., 2016).

Novelty detection techniques are traditionally developed and applied in more industrial applications, such as the monitoring of manufacturing processes (e.g. Al-Habaibeh and Parkin (2003)), machine condition monitoring (Carino et al., 2016; Clifton et al., 2014), mobile robotics (Sofman et al., 2011) and medical diagnoses (Tarassenko et al., 1995; Quinn and Williams, 2007; Clifton et al., 2011). These settings are usually determined by a stable positive class. Hence, the negative observations resemble more closely outliers in their most strict definition and the traditional outlier based detection methods are adequate. However, in cases where human behaviour plays an important role, both the positive and negative class are much less stable. Patcha and Park (2007) report that the traditional outlier or anomaly detection models result into high false alarm rates, when applied to network intrusion detection, a case where human hackers aim to intrude the network. The more volatile and varied nature of this data calls for new methods.

In the following, a concise overview of the prevalent approaches to novelty detection is presented. Figure 2 displays the main categories and the positioning of our proposed method within the novelty detection literature. A distinction is made between approaches that use only data from the positive class and those that create data for the negative class. The first category can be called one-class or unsupervised novelty detection, the second category two-class or (semi-)supervised novelty detection.

2.2. One-Class Novelty Detection

Within the one-class category, three major approaches are identified; probabilistic, distance-based and domain-based methods.

The probabilistic approaches estimate a probability density function of the positive data. The model will then classify points that lie outside of the high density region as a novelty. Both parametric and non-parametric approaches can be used. The multivariate Gaussian distribution is a frequently used parametric example.

$$P(x;\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{1}$$

The model will estimate the parameters of the multivariate Gaussian distribution. In the simplified bivariate example in Figure 1, a distribution will be constructed so that the positive examples are in the high density region and the (unobserved) negative examples would be classified in the tails of the distribution. In general, probabilistic methods for novelty detection are mathematically sound and effective if there is an accurately estimated probability density function (Pimentel et al., 2014). Higher dimensionality or small training sets are however dreadful for their performance. Pearson (2005) states that it is not convenient to find a suitable distribution. Non-parametric approaches can partly solve that issue, but they suffer from the curse of dimensionality and add computational complexity (Hempstalk et al., 2008). The curse of dimensionality implies that the number of required datapoints exponentially increases with the number of dimensions. Furthermore, a test point will be classified as a novelty if it does not follow the identified distribution, however the assumptions of many distributions might be too simplistic for real-life data. Ding et al. (2014) point at the importance of prior knowledge to circumvent this issue.

Distance-based methods include both clustering and nearest neighbour based methods. Nearest neighbour methods are among the most used approaches for novelty detection (Pimentel et al., 2014). The k-nearest-neighbours (k-NN) method is based on the assumption that normal points have close neighbours in the positive training set. A new data point x_{new} is classified as positive if the distance between x_{new} and its k nearest neighbours $NN_k(x_{new})$ is smaller than the distance between $NN_k(x_{new})$ and its respective k nearest neighbours $NN_k(NN_k(x_{new}))$ in the training set. This leads to the following formula for the kNN score.

$$\text{k-NN Score} = \frac{||x_{new} - NN_k(x_{new})||}{||NN_k(x_{new}) - NN_k(NN_k(x_{new}))||} \tag{2}$$

The observations in the new dataset that have the highest k-NN scores will be classified as negative. A common distance formula is Euclidean distance. The Euclidean distance formula between a and b for n dimensions is given by:

$$d(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{3}$$

Also Manhattan and Minkowski distance are used. Ding et al. (2014) selected k-NN as the best novelty detection method in an experimental evaluation, using ten benchmark datasets with different scale, dimensionality and problem complexity. Distance based methods have the advantage that they require no assumptions about the probability distribution of the normal data. The curse of dimensionality is again present; as the number of dimensions increases, the distance formula uses so many coordinates that the differences in distance will become relatively small. Clustering has the extra downside that the computational complexity increases quickly and the method is therefore not very scalable.

Domain-based methods construct a boundary using only the positive dataset. The density, which is crucial for probabilistic approaches, becomes irrelevant, because these methods are only concerned with the boundary. The reasoning behind this methodology is that one does not need to solve a more general problem than what is necessary (Schölkopf et al., 2000; Tax and Duin, 2004). The two leading approaches within the domain-based category are both based on support vector machines (SVM) and this category is therefore also referred to as one-class SVM. Schölkopf et al. (2000) developed a method that they describe as a natural extension of SVM to the case of unlabelled data. This algorithm returns a function f that is positive in a region that captures the majority of the datapoints and is negative elsewhere.

$$f(x) = sgn(\sum_i \alpha_i k(x_i, x) - \rho) \tag{4}$$

This method requires the user to fix in advance a percentage of the positive data that is allowed to fall outside the boundary that defines the positive class (the ν parameter). This means that outliers in the training data are tolerated more, which helps with the issue that not all outliers are examples of the negative class and not all examples of the negative class are outliers (Das et al., 2016). This flexibility is beneficial for the classification of diverse human behaviour, however the impact is expected to be still limited. Also, this parameter has a strong influence on the overall performance and therefore has to be chosen with great care (Manevitz and Yousef, 2001). The second domain-based approach was developed by Tax and Duin (2004). Their support vector data description (SVDD) method defines the novelty boundary as the hypersphere with minimum volume that includes all (or most) of the positive training data. A result of this definition is that the method is not well suited for high-dimensional spaces because of sparseness issues. In general, one-class SVMs are well known and repeatedly used for novelty detection (Clifton et al., 2014).

2.3. Two-Class Novelty Detection

The aforementioned unsupervised, purely one-class algorithms are often criticised for their high false negative rates (Das et al., 2016; Ding et al., 2014). Görnitz et al. (2013) also mention their frequently low predictive performance and point at the need for labelled data. Tax and Duin (2001) explain the inferior performance of one-class methods by the fact that the decision boundary is only supported from one side. On top of that, a vast amount of methods have been developed for (two-class, binary) classification and it would be beneficial if novelty detection could make use of these established methods. Two major approaches have been developed with the purpose of assigning a label to the unlabelled data points; manually labelling existing negative points (e.g. Görnitz et al. (2013)) and generating artificial data (e.g. Surace and Worden (2010)).

Manual or expert based labelling has been done through the inclusion of feedback loops (Abe et al., 2006; Görnitz et al., 2013). Abe et al. (2006) use active learning to interactively query the user. The user needs to manually label selected observations. The model learns from this information. The Active Anomaly Discovery (AAD) method, introduced by Das et al. (2016) is very similar and incorporates expert feedback through an interactive data exploration loop. It is clear that these approaches are very inefficient in most novelty detection applications since the overall presence of novelties is very low and it therefore takes a long time before novelties are discovered.

The artificial data generation approaches do not have this drawback and are therefore expected to be more efficient. Steinwart et al. (2005) mathematically prove that it is worthwhile to generate artificial data in order to apply a binary classification algorithm, given that the artificial samples are well chosen. In certain cases it is not possible to use authentic data for the negative class because for example the target service is under development (Barse et al., 2003). Artificial data provides an interesting solution.

A probabilistic approach to artificial data generation was introduced by Hempstalk et al. (2008). Their technique enhances the standard used one-class probabilistic approach by transforming the problem to two-class. Density estimation is used to form a reference distribution for the artificial class ($P(X|-)$). This distribution should be as close as possible to the positive class. $P(X|-)$ is used to generate data for the negative class. The positive data and the generated negative data are then labelled as such and mixed so that two-class classification can be used. The authors use Bayes' rule to combine the density function of the reference distribution ($P(X|-)$) with the class probability estimates ($P(+|X)$) in order to yield a description of the density function for the positive class ($P(X|+)$). This results in the following relation.

$$P(X|+) = \frac{(1 - P(+))P(+|X)}{P(+)(1 - P(+|X))} P(X|-) \tag{5}$$

$P(+)$ can be estimated by the proportion of positive examples in the mixed dataset. Using a balanced dataset ($P(+) = P(-) = 0.5$) reduces the formula to the following.

$$P(X|+) = \frac{P(+|X)}{1 - P(+|X)} P(X|-) \tag{6}$$

Applying a learning algorithm to this two-class training set results into a class probability estimator that will take the role of $P(+|X)$. $P(X|-)$ can also be calculated if an appropriate function was selected. Hempstalk et al. (2008) demonstrated with multiple datasets that this artificial data generation method improved performance.

Surace and Worden (2010) use a largely distance based approach, called negative selection, to generate the artificial data. A data point is pseudo-randomly generated using Gaussian distributions. If it is not similar enough to the existing positive data, it is labelled as part of the negative class. The similarity is calculated using the cosine similarity, which is an alternative for Euclidean distance.

$$sim(x, y) = \frac{\sum_{i=1}^{l} x_i y_i}{\sqrt{\sum_{x=1}^{l} x_i^2 \sum_{i=1}^{l} y_i^2}} \tag{7}$$

Another example of artificial data generation is given by Clifton et al. (2014). They develop a two-class counterpart for the one-class SVMs by generating data with the purpose of calibrating SVM output into probabilities. Their goal is different, but the methodology and their case study is relevant for this research. The monitoring of an industrial combustion engine was tackled by simulating data. The initial training phase of the engine was considered as the positive, stable data. Data for the negative class was generated by simulating unstable combustion through increasing fuel flow rates. This approach is thus based on data simulated by experiments, which would be infeasible when dealing with a human behaviour setting, such as fraud detection.

There is not one established, generally applicable method for novelty detection. This is largely due to the fact that specific settings require specific methods and a well-tailored method usually outperforms the more general method. The use of artificial data is a method that can be well-tailored and therefore supports this idea. There are however two important comments to be made on the use of artificial data for the negative class. Abe et al. (2006) and Hempstalk et al. (2008) notice that it is important that the artificial data is not too different from from the positive data, since the risk exists that the classifier would simply learn to distinguish real from artificial examples. A second remark is made by Görnitz et al. (2013), who warn that using artificially created data for the unknown class may in certain cases be inappropriate, since totally new and unseen (negative) classes are not easily picked up with a two-class method that was not trained on such data. One-class methods are expected to outperform the two-class approaches in that respect. However, the benefits of the two-class approach will in many situations outweigh this possible downside. The argument of Görnitz et al. (2013) also suggests that one-class classifiers do not suffer from the drawback that new negative examples are not picked up. It should however be remarked that traditional one-class approaches are based on the implicit assumption that all examples of the positive class are present in the dataset. This assumption is too strong in most cases and the result is that new positive examples will be misclassified, resulting in a higher number of false negatives (FN). A model that is able to learn from two classes will generally be able to make a better decision in these cases. Overcoming the risk that the two-class model, created on artificial data, only learns to distinguish between artificial and real data remains the most precarious issue. Our study proposes a method to overcome this remaining issue.

3. Method Development

3.1. Expert Knowledge

Models based on (partly) artificial data try to enhance the informativeness and therefore improve the predictive performance. The objective of this research is to further enhance the informativeness beyond what has been done in previous research, where artificial data was mainly used to create a decision boundary, without strong assumptions about the negative class. Ding et al. (2014) emphasize that the success of semi-supervised novelty detection is strongly dependent on the quality of the generated negative data. The method that we present takes into account the principle behind the guideline of Hempstalk et al. (2008); namely that it is important to prevent that the classifier only learns to distinguish between real and artificial examples. Our solution however differs because it does not require the artificial data to be close to the positive data. If it is known that the real negative data does differ enough from the real positive data, it would be suboptimal not to include this kind of information. From these findings, the most important condition for the artificial data arises; namely that the created instances should be as realistic as possible. They should be a good surrogate for actual negative data and prevent that the model only learns to distinguish between artificial and real data.

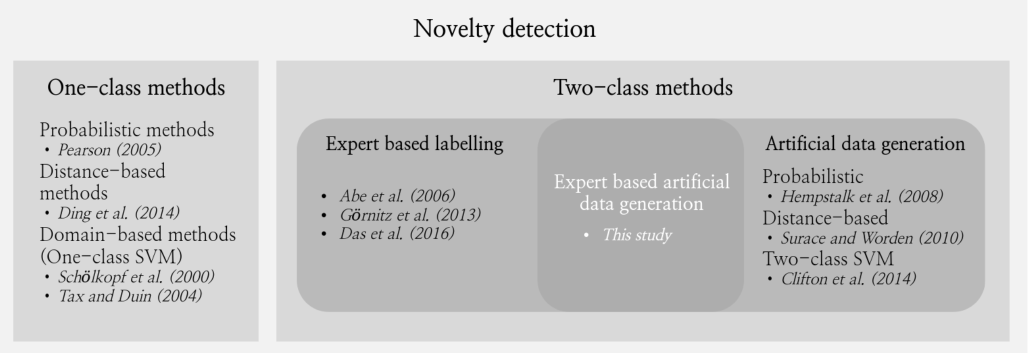


Figure 2: Positioning of our expert data generation method in the novelty detection literature.

As illustrated in Figure 2, this paper proposes the incorporation of expert knowledge in order to meet the condition of realism. Experts are qualified to come up with representative instances of negative data and to assess the realism of the synthetically created data. It has been suggested that prior (expert) knowledge about the case has the potential of tremendously increasing the performance of a classifier (Li et al., 2000; Larichev et al., 2002; Dayanik et al., 2006; Wang and Zhang, 2008; Lauer and Bloch, 2008; Utkin and Zhuk, 2014). Ashouri (1993) acknowledges that human reasoning enables identifying the structure of a problem and allows a qualitative analysis, but that handling quantitative, objective analysis is less obvious for human beings. The combination of expert knowledge with the predictive model incorporates the best of both worlds.

3.2. Expert Scenarios

The remaining question is how to implement the incorporation of expert knowledge in the creation of artificial data. Previous research has incorporated experts through a feedback loop (Abe et al., 2006; Görnitz et al., 2013), while others created synthetic data without incorporation of expert knowledge (e.g. Hempstalk et al. (2008)). Our method includes expert knowledge from the first stage, with the purpose of generating well informed data for the negative class. The expert knowledge defines one or more subspaces where the likelihood of observing negative data is higher. Semi-synthetic data will be constructed in these subspaces. An implementation that incorporates expert knowledge through the construction of scenarios is presented. Workshops with the relevant experts can be organised to come up with scenarios of abnormal or novel behaviour. These scenarios should be realistic and cover as much negative cases as possible.

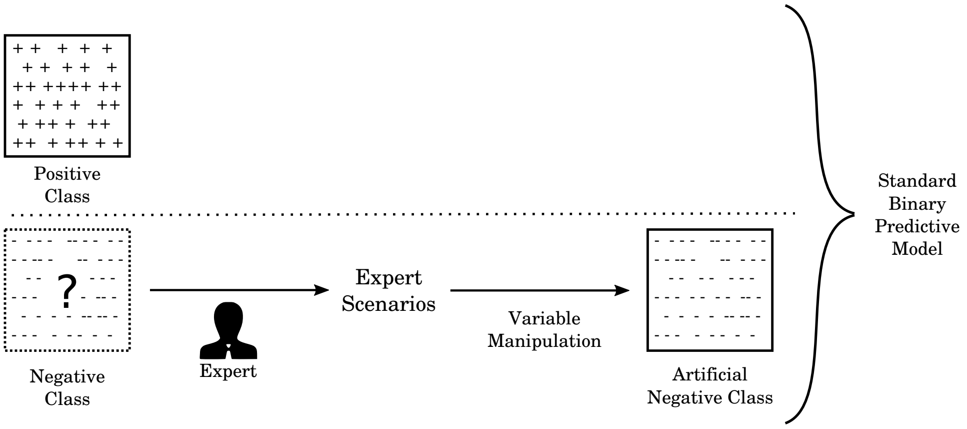


Figure 3: Expert Data Generation Method

Based on the selected scenarios, the next step involves generating actual data instances. By taking existing real datapoints as starting point, we guarantee that the new datapoints are realistic. The suggested approach can be called *variable manipulation*. Suppose that there are k variables in positive dataset P with p observations. The goal is to generate n instances for the negative class and store these in dataset N. We now proceed with the following steps:

1. Creation of instances of negative class - manipulated variables
   (a) Based on the expert scenarios, select m variables to manipulate (with $m < k$)
   (b) Set one or more expert rules for these m variables in order to generate artificial negative class examples. (e.g. variable x_k should have a value higher than 10 in scenario y).
2. Creation of instances of negative class - other variables
   (a) Calculate the values for all other non-manipulated variables
   (b) This results into dataset N
3. Stack dataset P (with the existing positive class examples) with set N
4. Build a predictive model, predicting $P(x_i \in P|x_{i,1}...x_{i,k})$.

This generic framework is meant to be applied to different cases. The structure and availability of the data will determine how to implement the calculations in step 2a. The calculations for this step will be network based in our case study.

There is a trade-off between the number of fixed, manipulated variables and the free variables. The higher the number of manipulated variables is, the higher the influence of expert knowledge will be. The goal is to restrict certain variables and see how the other variables react on that. The ratio p/n determines the balancedness of the final dataset. Our method enables a free choice of this ratio. This also eliminates the typical novelty detection problem of extremely unbalanced data; a lot of positive examples, but none or very few negative examples are known. The scenario method therefore offers an alternative to oversampling techniques which are normally used to remedy the unbalancedness. The synthetic minority oversampling technique (SMOTE), as developed by Chawla et al. (2002), is a widely used example. This purely data based technique shares similarities with our method, since it also generates new semi-synthetic instances.

The proposed method also provides freedom of selecting a specific binary classification algorithm, as it only interferes at the data generating process, the modelling part continues as if this was a standard two-class classification problem. In order to clarify the methodology, the next section applies the method to a fraud detection case in the telecom sector.

4. Case Study: Telecom subscription fraud

4.1. Business Problem

A case study is used to demonstrate that incorporating expert knowledge into the data generation phase enhances predictive performance in a real-life setting. The goal of this business case was to detect customers that commit telecom subscription fraud. Hilas and Mastorocostas (2008) define fraud detection as a field that uses techniques to monitor behaviour that deviates from the norm. This definition comes very close to the definition of novelty detection and it is therefore not surprising that novelty detection methods have been used for fraud detection (Patcha and Park, 2007; Jyothsna et al., 2011; Pimentel et al., 2014).

The problem of fraud is an important and worldwide issue in the telecommunications sector as it leads to an important loss in revenue (Farvaresh and Sepehri, 2011). Fawcett and Provost (1997) estimate that fraud costs the sector hundreds of millions of dollars per year. Hollmén and Tresp (1999) report that telecom companies lose between 2 and 5% of their total revenue to fraud. Fraud involves misuse, but it does not necessarily lead to direct legal consequences (Phua et al., 2010). Different types of telecom fraud have been identified by Gosset and Hyland (1999), Hollmén and Tresp (1999), Hilas and Mastorocostas (2008) and Farvaresh and Sepehri (2011); such as contractual fraud (subscription and premium fraud), hacking fraud, technical fraud and procedural fraud. This case deals with *subscription fraud* (Gosset and Hyland, 1999; Farvaresh and Sepehri, 2011), a type where advantage can be made of the service by using the mobile offering in a way that is not allowed by general terms and conditions of the subscription.

The company wanted to launch a *family plan* (Desai et al., 2018), which includes up to five SIM cards for a fixed total price. These SIM cards are of the flat-rate use type, which means that they include unlimited SMS and calls. The general terms of this *family plan* allow the SIM cards within one subscription to be used only by people within the same household. However, there is a financial incentive to distribute these anonymous cards to people outside of the household. Since an extra card - up to five - comes at no extra cost, the incentive for fraudsters is very high.

Telecom fraud detection systems are usually based on anomaly detection, where behaviour is compared with past behaviour of subscribers (Yesuf et al., 2017). Van Vlasselaer et al. (2013) state that because of the many domain-specific characteristics of different fraud types, it becomes important to use a domain-specific solution. Our method provides the necessary flexibility due to the incorporation of expert knowledge. Fraud detection also leads to a class imbalance problem, since in most cases there are very few fraud cases compared to the total dataset. As became clear from the previous section, the class imbalance problem is eliminated as the number of generated negative examples can be set as desired.

4.2. CDR Data

Identifying fraudulent customers in this case comes down to predicting whether the relationship between two customers that subscribed in the same household truly is a household relationship. To tackle this prediction problem, a vast amount of call detail record (CDR) data is used. Eagle et al. (2009) and Cho et al. (2011) confirm that CDR data has great potential to reveal relationships between people. Not only does the CDR data contain the calling behaviour between the individuals, it also contains their location. Geo-data has previously been used to infer social ties (Eagle et al., 2009; Crandall et al., 2010; Cranshaw et al., 2010; Cho et al., 2011).

This research uses the CDR data of two five-week periods. The *pre*-launch data is used for the model building and a first evaluation on a test set. The *post*-launch data enables to perform an additional real-life validation test of the modelling approach.

- Week 1 - 5: Pre-launch data
- Week 9: Product launch
- Week 24 - 28: Post-launch data

All of the analyses are performed on *dyad* level, where a *dyad* consists out of two customers. The main goal is to predict whether the relationship between two customers is a household (positive) or a fraudulent (negative) relationship.

Based on this network data, 66 variables are created in cooperation with the experts of the company. These variables can be categorized as pure network-based (e.g. number of calls within dyad, number of common contacts), spatial / spatio-temporal (e.g. distance between most used locations), network-spatial (e.g. distance between both when calling each other) and variables related to the home address (as approximated by the closest phone mast). An overview of these variables can be found in appendix.

4.3. Incorporating Expert Knowledge

Transforming the one-class problem into a two-class problem requires labelled data for both the positive and negative class. Examples of positive dyads are available in the *pre*-launch dataset. Before the launch of the new product, many customers already signed up as a household in order to receive only one bill. Apart from this practical convenience, there was no (financial) incentive to dishonestly sign up as a household. We can therefore use these dyads as examples of positive class data (see also Figure 4).

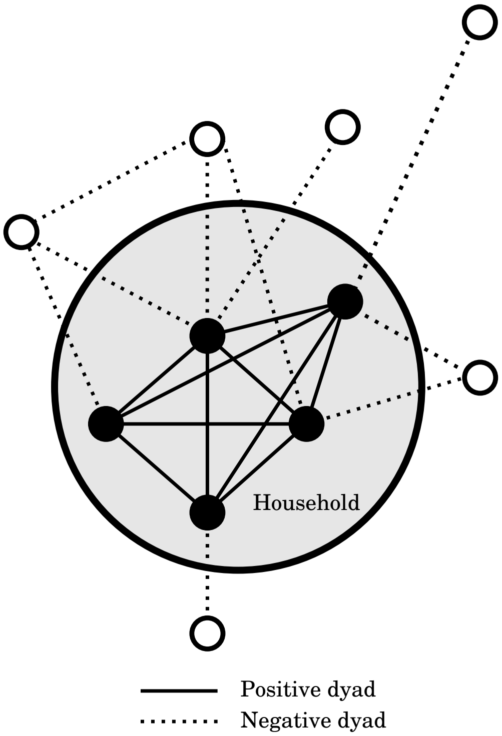


Figure 4: *Network dyad selection; positive and negative class.* Positive links connect individuals in the same household whereas negative links connect people that are not part of the same household.

Since there are no known fraud examples, instances have to be created for the negative class. One approach would be to use a random subset of all links between two individuals that are not within the same household. In other words, negative dyads are constructed by randomly combining two individuals. However, the major part of these instances would be dyads that are not at all related and therefore unrepresentative of the real negative class. This approach would also lead to negatives that are too different from the positives, conflicting with the guideline of Hempstalk et al. (2008). Moreover, complete random selection would also imply that there is no expert knowledge that can guide the selection process. Therefore restrictions are set in our method (see 3.2 Expert Scenarios: step 1b).

Our method overcomes this issue and meets with the requirements of realistic scenarios.

The experts identified two major fraud scenarios (with each three subscenarios): distributing the extra free SIM cards between friends and between neighbours. Translating these scenarios into usable data is done by using rules for variable manipulation. Based on the information in the existing CDR dataset, thresholds were set for three variables that define the scenarios. The assumption for the *friend* scenario is for example that there is at least one contact between them during the five-week period. For the neighbour scenarios, people that live within a radius of 200 metres are selected. Only taking this distance into account would lead to the previous described problem that most of those neighbours are unrelated. Therefore, the same calling behaviour rules as in the friend scenarios are taken into account.

Scenario	**Calls**		**SMS**	**Distance**
friend	≥ 1	or	≥ 1	
good friend	≥ 10	or	≥ 10	
best friend	≥ 40	or	≥ 100	
neighbour	≥ 1	or	≥ 1	$\leq 200m$
good neighbour	≥ 10	or	≥ 10	$\leq 200m$
best neighbour	≥ 40	or	≥ 100	$\leq 200m$

Table 1: *Expert Fraud Scenarios.* Or is non-exclusive.

This data is used for the subsequent analyses (see Table 2). An unbalanced dataset with 95% data from the positive class was used. The 5% data for the negative class was sampled from the expert generated negative dataset. This unbalanced set was used since this more closely resembles the true class distribution as expected by our experts. Instances from the different scenarios are merged and labelled as the negative class. Putting all scenarios together into one negative class enables to have a well-sampled representation of real-life data, where all scenarios will also occur together in the data. We use 5-fold cross-validation for all employed models in order to obtain a robust evaluation of the results. The division of the data over the five folds is reported in Table 2.

In the following, our expert model will be benchmarked against pure one-class models and models that create artificial data without expert knowledge (cf. Figure 2).

Scenario	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Total
household data (P)	$18,059$	$18,072$	$18,090$	$18,048$	$18,061$	$90,330$
fraud scenario data (N)	957	944	927	968	956	$4,752$
friend (N)	168	153	154	169	148	792
good friend (N)	158	154	161	155	164	792
best friend (N)	153	157	161	166	155	792
neighbour (N)	154	166	148	161	163	792
good neighbour (N)	173	151	158	156	154	792
best neighbour (N)	151	163	145	161	172	792

Table 2: *Pre-launch Data.* Number of instances (dyads) in the different folds for 5-fold cross validation.

4.4. Benchmark Model 1: One-Class Classification

We implement the three main categories of one-class classification, to benchmark our method. We follow the most important measures for novelty detection as identified by Ding et al. (2014) to evaluate our models; True Negative Rate (TNR) and AUC.

The main interest lies in the true negative rate (TNR), also called specificity, since the goal is to detect as many of the actual fraudsters as possible. TNR is equivalent to the fraud detection rate (FDR), it calculates the percentages of fraud cases that are detected. The TNR or FDR is calculated as $TN/(TN+FP)$, with TN = True Negative and FP = False Positive. As is common practice in novelty detection, we define the stable (=household) class consistently as positive throughout this research. As opposed to traditional two-class classification, therefore the class that we want to predict as accurately as possible is defined as the negative class, which turns the intuitive interpretation of specificity and sensitivity upside down. AUC measures the *Area Under the receiver operating characteristic Curve.* AUC provides an assessment of the overall performance of the classification model and is not dependent on a chosen cut-off value. The measure can be interpreted as the probability that a randomly chosen positive observation is ranked higher than a randomly chosen negative observation. The value should thus be as high as possible and will be between 0.5 and 1, where 1 indicates a perfect model and 0.5 indicates a random model from which we can not learn anything. All reported measures are the average performance over the folds of the 5-fold cross-validation (5-fold cv) approach. The values per fold can be found in Appendix (Table A2).

4.4.1. Benchmark Model 1a: One-Class Probabilistic

We selected a parametric model for the probabilistic one-class benchmark. The multivariate normal distribution was fitted to the positive data in the training set. The resulting density was then used to score the test data. The 5% observations with the lowest density score were classified as the negative class.

The results are displayed in the confusion matrix in Table 3, together with the other one-class benchmarks. All reported confusion matrices are summed over the different folds. The average FDR over the five folds is 7.64%. A random model would on average result into a FDR of 5% since, 5% of the test data is of the negative (fraud) class. This means that the one-class probabilistic model performs slightly better. The average AUC value over the five folds is 0.516. This also confirms that the model does slightly better than a random model.

4.4.2. Benchmark Model 1b: One-Class k-Nearest-Neighbours

Literature identified one-class k-NN as the best method in the distance based category. In two-class supervised learning, one can optimize the value for k. In one-class k-NN applications, this is not possible, because there is no ground truth. However, in our case, we can use the test data (see Table 2) to select a value for k. Based on the FDR, 1 was selected as k for four folds, $k = 3$ was selected for one fold (see Appendix). Furthermore, all variables were scaled before applying this distance based method.

The results of this benchmark model are again displayed in Table 3. The FDR amounts to 6.27%. The AUC for this model is 0.512.

4.4.3. Benchmark Model 1c: One-Class SVM

The one-class SVM (OCSVM) method of Schölkopf et al. (2000) is selected as benchmark for the third category of one-class models. Chang and Lin (2011) implemented the approach of Schölkopf et al. (2000) in the popular libSVM package. The interface to libSVM as provided in R was used (Meyer et al., 2017). The OCSVM was defined using the ν parameter. For two-class SVM, this parameter serves as an upper bound for the training error and a lower bound for the number of support vectors, whereas for OCSVM ν is an upper bound for the fraction of negative class data (Hornik et al., 2006). This way, ν can also be interpreted as the *novelty rate*. ν is set at 0.05 and the one-class model is trained on the positive household data using 5-fold cv as presented in Table 2.

FDR is 8.56% for the one-class SVM model. The AUC value can not be calculated because one-class SVM outputs only binary decisions without probabilities based on which the observations could be ranked

	Probabilistic			**k-NN**			**One-Class SVM**		
	(Benchmark 1a)			(Benchmark 1b)			(Benchmark 1c)		
	Predicted Positive	Predicted Negative	%Novelty	Predicted Positive	Predicted Negative	%Novelty	Predicted Positive	Predicted Negative	%Novelty
household (P)	85,938	4,392	4.86	85,873	4,457	4.93	85,687	4,643	5.14
fraud (N)	4,389	363	**7.64**	4,454	298	**6.27**	4,345	407	**8.56**
friend (N)	735	57	7.20	706	86	10.86	728	67	8.08
good friend (N)	748	44	5.56	738	54	6.82	740	52	6.57
best friend (N)	670	122	15.40	775	17	2.15	661	131	16.54
neighbour (N)	768	24	3.03	731	61	7.70	764	28	3.54
good neighbour (N)	770	22	2.78	739	53	6.69	770	22	2.78
best neighbour (N)	698	94	11.87	765	27	3.41	682	110	13.89

Table 3: *Confusion Matrix One-Class Benchmark Models.*(5-fold cv) %Novelty displays per class what percentage of the cases was predicted as novelty. For the fraud scenarios, %novelty equals the Fraud Detection Rate and can be interpreted as the percentage of actual fraud cases that are detected. For example, the FDR for the one-class SVM is thus 8.56%. For the positive, household class, %novelty equals 1 - True Positive Rate and can be interpreted as the percentage of cases incorrectly classified as fraud.

The performance of these three benchmarks is in the same range. We can observe that OCSVM (Benchmark 1c) scores best, followed by the probabilistic approach and one-class k-NN. Benchmark 1c has the best FDR and will therefore be selected as representative for the one-class approaches.

4.5. Benchmark Model 2: Two-Class Artificial Data Generation Models

As two-class artificial data generation can be seen as an intermediary step towards our expert data generation (see also Figure 2), we implement models from this category as a second class of benchmarks.

4.5.1. Benchmark Model 2a & 2b: Probabilistic Artificial Data Generation

We follow the approach of Hempstalk et al. (2008) and use equation (6) to implement two probabilistic models. Benchmark model 2a uses the multivariate normal distribution as reference distribution for the artificial class ($P(X|-)$), benchmark model 2b uses a uniform distribution (as suggested by Hempstalk et al. (2008)). Hempstalk et al. (2008) stress that $P(X|-)$ should be as close as possible to $P(X|+)$, we therefore use the parameters as estimated for $P(X|+)$ to generate data for the negative class. For the multivariate normal model (2a), this means that $P(X|-)$ is equal to the density estimated in benchmark model 1a. The boundaries of the uniform model are determined by the boundaries of the positive class.

Artificial data is generated from the respective distributions for the negative class. We then use SVM to train the classifier $P(+|X)$. The SVM implementation in R (Meyer et al., 2017) was set up to output class probabilities instead of only class labels, based on Platt et al. (1999). The used type of SVM is C-classification, with a radial basis function (RBF) kernel. All SVM models throughout this research (except for the one-class SVM) use these settings. As an intermediate result, we report the performance of $P(+|X)$ on the generated artificial dataset itself (Table 4).

$P(X\|-)$	AUC	FDR
Multivariate Normal (2a)	0.999	99.73
Uniform (2b)	1	100

Table 4: Average performance (5-fold cv) $P(+|X)$ on artificially generated data.

We observe perfect separation for model 2b and nearly perfect separation for 2a. This intermediate result suggests that the model only learns to distinguish between artificial and non-artificial data.

Combining $P(+|X)$ and $P(X|-)$ using equation (6) results in the final prediction (Table 5). In line with the novelty rate and the frequency of artificial fraud dyads in the dataset, the dyads with the 5% lowest densities are classified as fraud.

	Multivariate Normal			**Uniform**		
	(Benchmark 2a)			(Benchmark 2b)		
	Predicted Positive	Predicted Negative	%Novelty	Predicted Positive	Predicted Negative	%Novelty
household (P)	85,944	4,386	4.86	86,017	4,313	4.77
fraud (N)	4,383	369	**7.77**	4,310	442	**9.30**
friend (N)	736	56	7.07	686	106	13.38
good friend (N)	747	45	5.68	722	70	8.84
best friend (N)	666	126	15.91	733	59	7.45
neighbour (N)	769	23	2.90	688	104	13.13
good neighbour (N)	771	21	2.65	741	51	6.44
best neighbour (N)	694	98	12.37	740	52	6.57

Table 5: *Confusion Matrix Two-Class Probabilistic Artificial Benchmark Models* (5-fold cv).

We observe that the FDR for the multivariate normal model (2a) is slightly higher than for the corresponding one-class probabilistic model (Benchmark 1a). Selecting a uniform reference distribution (2b) leads to a further improvement in FDR.

4.5.2. Benchmark Model 2c: Distance-based Artificial Data Generation

The approach of Surace and Worden (2010) was mimicked to generate artificial data for the negative class. In a first step, data was generated from a multivariate Gaussian distribution (cf. Benchmark models 1a and 2a). In a second step, the average cosine similarity of the generated data points - with respect to the positive set - was calculated. Only cases that had a lower average cosine similarity than the 5% lower boundary of internal cosine similarity in the positive set were retained as generated negative cases.

Now that data has been created for the negative class, a traditional support vector machine (SVM) is used for classification. Therefore, we will refer to this method as two-class artificial SVM as well. Again, the dyads that are in the lowest 5% probability of being household are classified as fraud.

The results are presented in Table 6. The extremely high FDR, together with an average AUC of 0.999 over the folds indicates that the model offers almost perfect separation. Similar to the intermediate results of benchmark models 2a and 2b, this raises the alarm that the model might just be learning to distinguish between real data (of the positive class) and artificial data (of the negative class). This important aspect will be discussed further in the following sections.

	Predicted Positive	Predicted Negative	%Novelty
household (P)	$90,245$	85	0.09
fraud (N)	83	$4,672$	**98.25**

Table 6: Two-Class Artificial Model (Benchmark 2c): Confusion Matrix on the Artificial Test Set (5-fold cv). Fraud Detection Rate = 98.25%.

Using this model on the expert test set results into a huge drop in performance (see Table 7). Nearly all dyads are predicted as household, less than 0.001% of dyads have a lower than 0.975 probability of being household. Using the same absolute cut-off value as in Table 6 would lead to nearly all dyads in the test set being classified as positive. However, in accordance with other benchmarks, we classify the dyads with the 5% lowest probabilities as fraud in the confusion matrix (Table 7).

	Two-Class Artificial (Benchmark 2c)		
	Predicted Positive	Predicted Negative	%Novelty
household (P)	$86,068$	$4,262$	4.72
fraud (N)	$4,259$	493	**10.37**
friend (N)	728	64	8.08
good friend (N)	736	56	7.07
best friend (N)	646	146	18.43
neighbour (N)	724	68	8.59
good neighbour (N)	741	51	6.44
best neighbour (N)	684	108	13.64

Table 7: Two-Class Artificial Model (Benchmark 2c): Confusion Matrix on the Expert Test Set (5-fold cv).

4.6. Two-Class Expert Model

The implementation of our two-class expert method uses the same expert based dataset as the other benchmarks. The crucial difference is that the two-class expert model does use the generated negative expert data for the training of the model. To achieve maximal comparability with the benchmark models where a classifier was needed (1c, 2a, 2b and 2c), we again use SVM as binary classifier. This means that the difference in performance can be attributed solely to the two-class expert method and not to the difference in the background binary classifier. Furthermore, SVM has a strong theoretical foundation and excellent predictive performance (Lessmann and Voß, 2009). It also has a good generalisation performance when applied to noisy data. In this specific case there is a large heterogeneity in the behaviour of customers and a method that generalizes well is desirable.

The predictions on the test set are shown in Table 8. The FDR is now 48.72%. The model resulted in an average AUC of 0.824 over the folds. These values indicate decent performance, heavily improving upon the benchmark models. The two-class expert model also has the lowest number of households incorrectly classified as fraud, 2.70% compared to an average of 4.88% for the benchmark models. Keeping in mind that a random model would classify 5% of the households as fraud, this can be considered as a considerable improvement.

	Two-Class Expert		
	Predicted Positive	Predicted Negative	%Novelty
household (P)	87,890	2,440	2.70
fraud (N)	2,437	2,315	**48.72**
friend (N)	225	567	71.59
good friend (N)	339	453	57.20
best friend (N)	440	352	44.44
neighbour (N)	333	459	57.95
good neighbour (N)	454	338	42.68
best neighbour (N)	646	146	18.43

Table 8: Two-class Expert SVM Confusion Matrix per Scenario on the Test Set (5-fold cv). Fraud Detection Rate = 48.72%.

When applying the two-class expert method, the user is however not restricted to SVM. We therefore also demonstrate the robustness of the method by applying six different classifiers (Table 9 and Figure 5). The results are robust over the six different classifiers, with SVM scoring average.

Model	AUC	FDR
SVM	0.824	48.72%
Logistic Regression	0.835	45.56%
AdaBoost	0.884	51.02%
Decision Tree	0.770	41.50%
Random Forest	0.865	51.38%
Neural Network (1-layer)	0.793	40.46%

Table 9: *Robustness Check.* Average performance (5-fold cv) of the Two-Class Expert Method for different binary classifiers on expert test data.

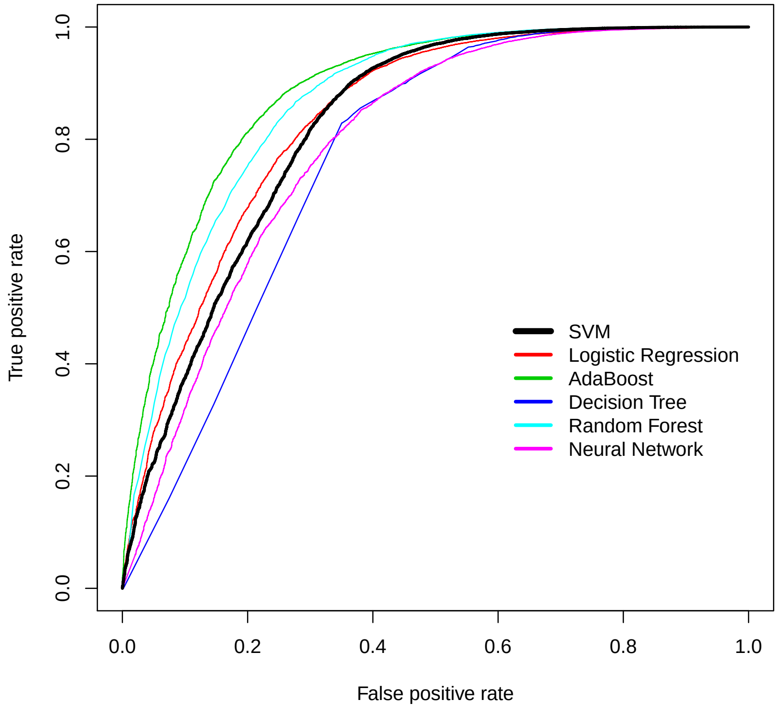


Figure 5: *Robustness Check.* ROC Curves of investigated binary classifiers for the Two-Class Expert Method on expert test data (5-fold cv).

A valuable novelty detection method should be generalisable and avoid overfitting. In other words, it should perform well on a new dataset that was not involved in the generation of the novelty detection model. Therefore, we also evaluate the performance of the most important models on the dataset generated by the other method. As the pure one-class models do not generate data, only the data generated by the two-class models can be used for this test. For the pure one-class approaches, model 1c was selected as the best performing model. Furthermore, our main objective is to compare a two-class expert method over a one-class method and a two-class artificial method. Hence, keeping the background algorithm (SVM) equal makes it ideal for comparison. For this reason, we also select benchmark 2c as example of the two-class artificial data generation models. The results of this cross comparison are reported in Table 10.

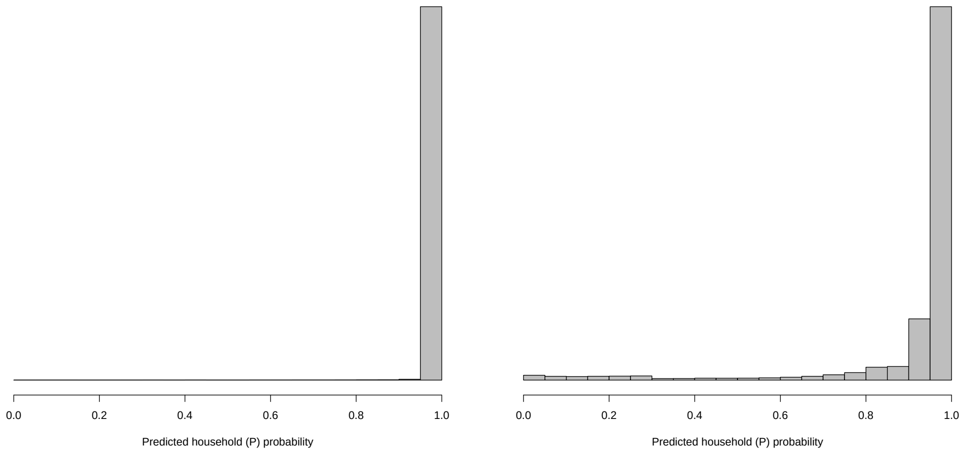
	Artificial Data	Expert Data
One-Class (Benchmark 1c)	5.89	8.56
Two-Class Artificial (Benchmark 2c)	98.25	10.37
Two-Class Expert	**15.01**	**48.72**

Table 10: Fraud Detection Rate (FDR) of three novelty detection methods evaluated on two datasets. The artificial dataset contains the negative data as generated by the purely artificial approach (cf. Section 4.5). The expert dataset contains the negative data as generated by our expert data generation method (cf. Section 4.3). The models were trained on their respective dataset.

The performance of the two-class artificial model drops tremendously when deployed on the other data set (87.88 percentage point drop). This indicates that the artificial negatives on which the model was trained are indeed too artificial and too different from the positives. It demonstrates the assumption that the artificial model only learns to distinguish between artificial and non-artificial cases. As expected, our two-class expert model also drops in performance when deployed on the other data set, however the drop is much smaller (33.71 p.p.). This illustrates that the expert model detects novelties in a more generalised way and thus not only performs well on the data on which it was trained. The expert model also performs better than the one-class SVM, even on the artificial data set on which it was not developed. These analyses show the value of expert based data generation for novelty detection. That is, the approach finds the required balance between generating novelties that are different enough from the positive data, while not being too distinct, so that classification algorithms do not overfit the artificial data.

4.7. Real-Life Post-Launch Implementation and Validation

In this and the next section we implement and validate the model on new real-life data. The two-class models are used to score all (100,000+) dyads of the *post*-launch data, i.e. data that contains potential fraudsters. Figure 6b displays the distribution of the predictions of the proposed expert model. As is typical for novelty detection and fraud problems, we observe a strong unbalancedness in the predicted probabilities. Nevertheless, the histogram is fairly dispersed when compared to the predictions of the artificial model (Figure 6a). The histogram of the artificial model shows that all cases are classified as positive, non-fraudulent cases. This again indicates that the artificial model actually classifies these cases merely as *real*, *non-artificial* cases, hence providing no information about fraudulent behaviour. Again, as in the previous section, the artificial model shows to only have learned to identify very specific artificial outliers rather than more general anomalies.



(a) Histogram of predictions of the Two-Class Artificial model (Benchmark 2c) on *post*-launch data. All cases are classified as positive. The model has no practical value.

(b) Histogram of predictions of the Two-Class Expert model on *post*-launch data. The predictions are as expected, the major part is classified as non-fraudulent. We clearly observe the characteristic unbalancedness of one-class and fraud problems. Cases in the left side of the histogram can be labelled as fraud suspects.

Figure 6: Comparison of predictions on *post*-launch dataset.

4.8. Manual Checks on Post-Launch Predictions

Now that the expert model predicts some dyads as fraud suspects, the company was involved to verify whether these cases are true fraudulent dyads. The sample of dyads with the lowest household probability scores was transferred to the company. A specialised fraud team that could make use of specific data sources, assessed whether a dyad was fraudulent. Due to the nature of this case and the fact that for many users there was limited additional information, it is impossible to assess all selected dyads. 478 dyads were checked in total in order to obtain 100 validly labelled dyads. Figure 7 shows the results for these 100 dyads. Dyads with a decision beyond reasonable doubt were labelled as *likely*. The pure fraud and household categories contain only cases where fraud could be identified by the specialised company fraud team. The results show that a very high proportion (about 90%) of these dyads indeed were considered to be fraudulent by the company fraud team. For the company this result was surprising as most of their fraud models (in different contexts though) suffer from much higher false positive rates.

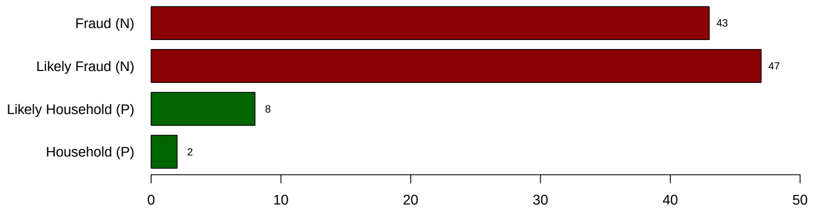


Figure 7: Results of manual labelling of predicted fraud suspects by fraud team of the company. The vast majority of these top predicted fraud dyads are indeed labelled as such in the real-life validation.

To illustrate an example, Figure 8 and 9 provide details on a fraud case that was identified by the expert model. The relation between the individuals depicted in orange and red was part of the top 100. In Figure 8 we see that the red individual is situated further in the network structure. In Figure 9 we observe the same in terms of location. The manual checks indeed identified this person as a fraudulent part of the household. The predictions for all dyads in this household, together with their actual label are presented in Table 11. The table also displays the predictions of representative benchmark models. For this example, it is clear that the latter provide virtually no information about fraud behaviour.

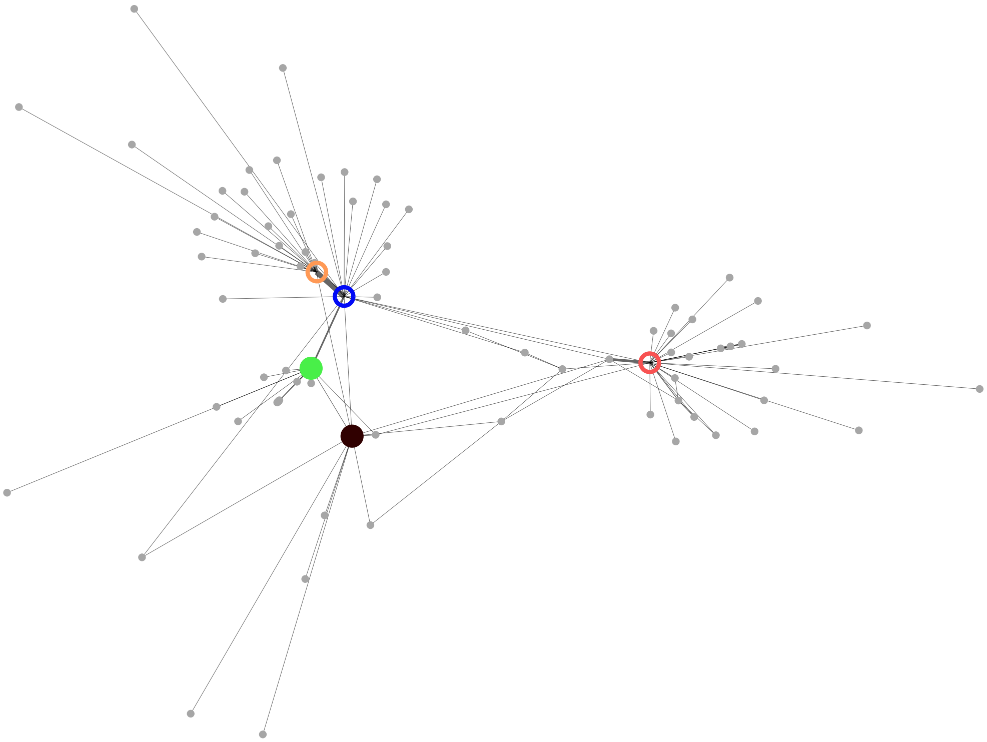


Figure 8: *Identified Fraud Case: CDR Network Visualisation.* Full circles represent individuals that are valid members of the household, hollow circles are fraudulent members. The linewidth of the links reflects the number of calls and SMS between two individuals within the dyad. People with stronger connections are also displayed closer to each other, as calculated by the ForceAtlas2 algorithm in the Gephi software (Bastian et al., 2009). Our model identified the relation between the red and the orange individual as fraudulent. We observe in this network that both individuals have no clear connection. There is very little overlap in their respective social networks as well. In Figure 9 we can draw the same conclusion based on location data.

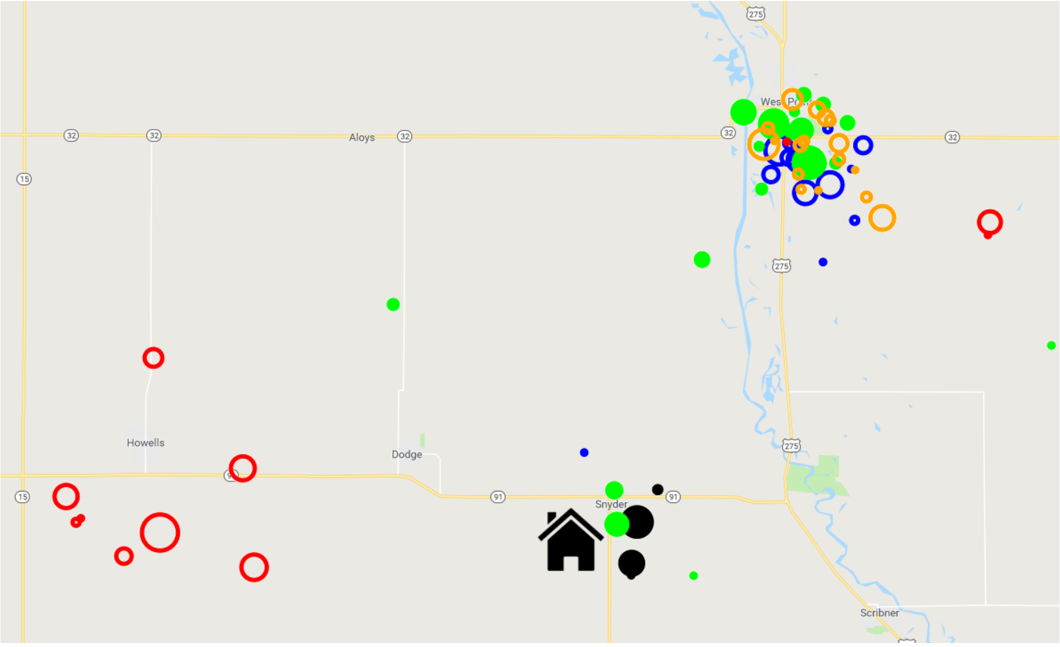


Figure 9: *Identified Fraud Case: Location Plot.* The size of the circles represents the number of calls/SMS on a certain location. The house indicates the home location of the household. The individuals are displayed with the same coding as Figure 8. The background map in this figure does not represent the true geographical location in order to preserve anonymity. We can observe that the red and orange individuals - who were identified as fraudulent by our model - have no location tags near the home location. The locations of both individuals also are very distinct from one another.

A	B	Manual Label	Two-Class Expert	Two-Class Artificial (Benchmark 2c)	One-Class (Benchmark 1c)
Orange	Red	Fraud (N)	0.207	1.000	P
Orange	Black	Fraud (N)	0.347	1.000	P
Green	Red	Fraud (N)	0.392	1.000	P
Blue	Red	Fraud (N)	0.407	0.999	P
Orange	Green	Fraud (N)	0.736	1.000	P
Blue	Green	Fraud (N)	0.759	1.000	P
Blue	Black	Fraud (N)	0.919	1.000	P
Black	Red	Fraud (N)	0.923	1.000	P
Green	Black	Likely Household (P)	0.960	0.993	P
Blue	Orange	Likely Household (P)	0.972	1.000	P

Table 11: Predictions of the different classes of models for a fraud example. For the *expert* and *artificial* model, these are the household (P) probability predictions. The *one-class* model outputs no predictions, but a binary decision, in this case positive for all dyads, hence all dyads are predicted as household.

5. Discussion

The two-class expert method outperformed both the one-class and two-class artificial benchmarks in this study. The latter techniques have nevertheless displayed acceptable results in previous research. An important distinction with the proposed method is that human behaviour is modelled, whereas traditionally applications in for example machine monitoring have been explored. This research demonstrated that the existing methods are not sufficient for the classification of human behaviour.

The one-class methods suffer from the obvious drawback that they can only learn from one class. The artificial two-class methods also failed to significantly boost performance. These artificial data generation approaches take two rather extreme forms, that are both not well suited for a human behaviour application. The first (cf. Benchmark models 2a and 2b) generates artificial data based on the distribution of the original positive class. Intuitively it is clear that we can not learn a lot about the actual negative class in such case. The results indicate that improvement upon the one-class models is indeed minimal. The other approach (cf. Benchmark model 2c) is extreme in the sense that it creates artificial data that is very distinct from the positive class. When dealing with human behaviour, the variability within the data becomes large, both for the positive and negative data, the overlap between both classes is larger than in non-human applications. The boundary between both classes becomes less strict and hence the boundary that will come out of such model will be too strict and too artificial. These artificial, non-expert, data generation methods thus use automated, unrealistic assumptions, whereas we proposed to incorporate well-informed expert based assumptions. This addition of extra information, in the form of expert knowledge, adds strongly to the classification power.

Taking a closer look at the confusion matrix of the expert model (Table 8),we observe that the more restricted the scenarios become, the lower the FDR becomes. This can be explained by the fact that in this case, the more strict scenarios for the negative class more closely resemble the positive (household) class, which makes it more difficult for a model to distinguish both classes. Nevertheless, it is important to include these scenarios, because according to the experts these scenarios more closely resemble realistic cases. What the model learns from these cases is likely the most important for the actual detection of fraud cases in the real-life validation.

Even though it remains to be explored how much the expert can add to the more traditional machine monitoring cases, the presented method promises to be well suited to tackle these and other novelty detection problems as well, due to its flexible nature. The expert scenario method can flexibly introduce scenarios that are not in the original dataset. Hence, creating semi-synthetic data has the benefit of providing data that is well tailored to specific requirements. Furthermore, creating expert data is much cheaper when compared to manually labelling data.

This research is an addition to and not an argument against the traditional one-class approaches for novelty detection. Pure one-class approaches are sometimes considered to be better at identifying complete novel cases. Therefore, the selection of the appropriate methodology will depend on the misclassification cost of these cases. However, when implementing the expert-based two-class methodology, it is important to invest a fair amount of time in the construction of the scenarios, so that all relevant scenarios are represented in the negative data. Furthermore, the two-class expert method is able to detect cases that are not explicitly modelled in one of the scenarios. The classification algorithm detects underlying, shared characteristics between the scenarios that are also shared with novel cases. In the case study, other types of subscription fraud that were not explicitly modelled, were detected. An example is the use of the extra SIM cards by older children of the household that already moved out. Furthermore, as discussed before, most one-class methods assume that the positive class is perfectly represented in the positive dataset. However, this is unlikely to be true in many cases, which leads to a higher number of false negatives.

The major limitation of the presented research is that the expert method is validated in a single case study. To enlarge the validity of this method, future research needs to explore how the method translates to other cases and contexts. Another remaining issue is the trade-off between the number of manipulated variables and the free variables. The higher the amount of expert knowledge, the higher the number of manipulated variables. In general, a higher level of expert knowledge will shrink the space in which data for the unknown class is generated. These better defined regions however come at the cost of possibly losing the generality that characterizes traditional one-class approaches. Further research is needed to examine what the impact of using different levels of expert knowledge could be.

6. Conclusion

The transformation of a one-class problem into a two-class problem was examined. This method was assessed in the context of fraud detection for a new telecom service. The absence of labelled fraud examples calls for the use of one-class novelty detection methods. However, traditional one-class methods perform poorly in a case dealing with human behaviour. Hence, a new method is developed to deal with this issue. Using semi-synthetic data for the negative class has great potential. Previous research used artificial data with the same purpose of better defining a boundary around the positive class, but without clear assumptions about the negative class. We introduced the incorporation of expert knowledge in order to use clear assumptions. This enhances the informativeness of the artificial data and further improves the classification performance. Experts build realistic, representative scenarios that describe the behaviour of the humans belonging to the negative class. Using these scenarios, instances were generated for the negative class with variable manipulation. The method was tested in a real-life telecom subscription fraud case. The two-class expert method clearly outperformed the conventional one-class benchmark models. The method also improved upon the artificial two-class non-expert benchmarks, that were characterized by the problem of creating models that merely learned to distinguish between artificial and non-artificial cases. The performance of the model was also examined in a manual validation phase for a new post product launch dataset. The model performed very well in this real-life setting and was able to detect real fraud cases with a model build on expert fraud scenarios. Including expert knowledge strongly helped to classify the diverse human behaviour data, where less flexible traditional methods failed. The manual checks are costly in terms of manpower and hence a predictive model that prioritises, can generate a lot of value.

Acknowledgements

The authors would like to express their gratitude towards the many involved employees of the anonymous telecommunications provider.

References

Abe, N., Zadrozny, B., Langford, J., 2006. Outlier detection by active learning. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 504–509.

Al-Habaibeh, A., Parkin, R., 2003. An autonomous low-cost infrared system for the on-line monitoring of manufacturing processes using novelty detection. The International Journal of Advanced Manufacturing Technology 22 (3-4), 249–258.

Ashouri, F., 1993. An expert system for predicting gas demand: A case study. Omega 21 (3), 307–317.

Barse, E. L., Kvarnstrom, H., Jonsson, E., 2003. Synthesizing test data for fraud detection systems. In: Computer Security Applications Conference, 2003. Proceedings. 19th Annual. IEEE, pp. 384–394.

Bastian, M., Heymann, S., Jacomy, M., 2009. Gephi: An open source software for exploring and manipulating networks.
URL `http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154`

Carino, J. A., Delgado-Prieto, M., Zurita, D., Millan, M., Redondo, J. A. O., Romero-Troncoso, R., 2016. Enhanced industrial machinery condition monitoring methodology based on novelty detection and multi-modal analysis. IEEE access 4, 7594–7604.

Chang, C.-C., Lin, C.-J., 2011. Libsvm: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST) 2 (3), 27.

Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P., 2002. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research 16, 321–357.

Cho, E., Myers, S. A., Leskovec, J., 2011. Friendship and mobility: user movement in location-based social networks. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 1082–1090.

Clifton, L., Clifton, D. A., Watkinson, P. J., Tarassenko, L., 2011. Identification of patient deterioration in vital-sign data using one-class support vector machines. In: 2011 federated conference on computer science and information systems (FedCSIS). IEEE, pp. 125–131.

Clifton, L., Clifton, D. A., Zhang, Y., Watkinson, P., Tarassenko, L., Yin, H., 2014. Probabilistic novelty detection with support vector machines. IEEE Transactions on Reliability 63 (2), 455–467.

Crandall, D. J., Backstrom, L., Cosley, D., Suri, S., Huttenlocher, D., Kleinberg, J., 2010. Inferring social ties from geographic coincidences. Proceedings of the National Academy of Sciences 107 (52), 22436–22441.

Cranshaw, J., Toch, E., Hong, J., Kittur, A., Sadeh, N., 2010. Bridging the gap between physical location and online social networks. In: Proceedings of the 12th ACM international conference on Ubiquitous computing. ACM, pp. 119–128.

Das, S., Wong, W.-K., Dietterich, T., Fern, A., Emmott, A., 2016. Incorporating expert feedback into active anomaly discovery. In: Data Mining (ICDM), 2016 IEEE 16th International Conference on. IEEE, pp. 853–858.

Dayanik, A., Lewis, D. D., Madigan, D., Menkov, V., Genkin, A., 2006. Constructing informative prior distributions from domain knowledge in text classification. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 493–500.

Desai, P., Purohit, D., Zhou, B., 2018. Allowing consumers to bundle themselves: The profitability of family plans. Marketing Science.

Ding, X., Li, Y., Belatreche, A., Maguire, L. P., 2014. An experimental evaluation of novelty detection methods. Neurocomputing 135, 313–327.

Eagle, N., Pentland, A. S., Lazer, D., 2009. Inferring friendship network structure by using mobile phone data. Proceedings of the national academy of sciences 106 (36), 15274–15278.

El-Yaniv, R., Nisenson, M., 2007. Optimal single-class classification strategies. In: Advances in Neural Information Processing Systems. pp. 377–384.

Farvaresh, H., Sepehri, M. M., 2011. A data mining framework for detecting subscription fraud in telecommunication. Engineering Applications of Artificial Intelligence 24 (1), 182–194.

Fawcett, T., Provost, F., 1997. Adaptive fraud detection. Data mining and knowledge discovery 1 (3), 291–316.

Görnitz, N., Kloft, M. M., Rieck, K., Brefeld, U., 2013. Toward supervised anomaly detection. Journal of Artificial Intelligence Research.

Gosset, P., Hyland, M., 1999. Classification, detection and prosecution of fraud in mobile networks. Proceedings of ACTS mobile summit, Sorrento, Italy.

Hempstalk, K., Frank, E., Witten, I. H., 2008. One-class classification by combining density and class probability estimation. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 505–519.

Hilas, C. S., Mastorocostas, P. A., 2008. An application of supervised and unsupervised learning approaches to telecommunications fraud detection. Knowledge-Based Systems 21 (7), 721–726.

Hodge, V., Austin, J., 2004. A survey of outlier detection methodologies. Artificial intelligence review 22 (2), 85–126.

Hollmén, J., Tresp, V., 1999. Call-based fraud detection in mobile communication networks using a hierarchical regime-switching model. Advances in Neural Information Processing Systems, 889–895.

Hornik, K., Meyer, D., Karatzoglou, A., 2006. Support vector machines in r. Journal of statistical software 15 (9), 1–28.

Japkowicz, N., Myers, C., Gluck, M., et al., 1995. A novelty detection approach to classification. In: IJCAI. Vol. 1. pp. 518–523.

Jyothsna, V., Prasad, V. R., Prasad, K. M., 2011. A review of anomaly based intrusion detection systems. International Journal of Computer Applications 28 (7), 26–35.

Khan, S. S., Madden, M. G., 2014. One-class classification: taxonomy of study and review of techniques. The Knowledge Engineering Review 29 (3), 345–374.

Larichev, O., Asanov, A., Naryzhny, Y., 2002. Effectiveness evaluation of expert classification methods. European Journal of Operational Research 138 (2), 260–273.

Lauer, F., Bloch, G., 2008. Incorporating prior knowledge in support vector machines for classification: A review. Neurocomputing 71 (7-9), 1578–1594.

Lessmann, S., Voß, S., 2009. A reference model for customer-centric data mining with support vector machines. European Journal of Operational Research 199 (2), 520–530.

Li, H., Li, Z., Li, L. X., Hu, B., 2000. A production rescheduling expert simulation system. European Journal of Operational Research 124 (2), 283–293.

Manevitz, L. M., Yousef, M., 2001. One-class svms for document classification. Journal of machine Learning research 2 (Dec), 139–154.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., 2017. e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.6-8.
URL `https://CRAN.R-project.org/package=e1071`

Moya, M. M., Koch, M. W., Hostetler, L. D., 1993. One-class classifier networks for target recognition applications. Tech. rep., Sandia National Labs., Albuquerque, NM (United States).

Patcha, A., Park, J.-M., 2007. An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer networks 51 (12), 3448–3470.

Pearson, R. K., 2005. Mining imperfect data: Dealing with contamination and incomplete records. Vol. 93. Siam.

Phua, C., Lee, V., Smith, K., Gayler, R., 2010. A comprehensive survey of data mining-based fraud detection research. arXiv preprint arXiv:1009.6119.

Pimentel, M. A., Clifton, D. A., Clifton, L., Tarassenko, L., 2014. A review of novelty detection. Signal Processing 99, 215–249.

Platt, J., et al., 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers 10 (3), 61–74.

Quinn, J. A., Williams, C. K., 2007. Known unknowns: Novelty detection in condition monitoring. In: Iberian Conference on Pattern Recognition and Image Analysis. Springer, pp. 1–6.

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., Platt, J. C., 2000. Support vector method for novelty detection. In: Advances in neural information processing systems. pp. 582–588.

Sofman, B., Neuman, B., Stentz, A., Bagnell, J. A., 2011. Anytime online novelty and change detection for mobile robots. Journal of Field Robotics 28 (4), 589–618.

Steinwart, I., Hush, D., Scovel, C., 2005. A classification framework for anomaly detection. Journal of Machine Learning Research 6 (Feb), 211–232.

Surace, C., Worden, K., 2010. Novelty detection in a changing environment: a negative selection approach. Mechanical Systems and Signal Processing 24 (4), 1114–1128.

Tarassenko, L., Hayton, P., Cerneaz, N., Brady, M., 1995. Novelty detection for the identification of masses in mammograms.

Tax, D. M., Duin, R. P., 2001. Uniform object generation for optimizing one-class classifiers. Journal of machine learning research 2 (Dec), 155–173.

Tax, D. M., Duin, R. P., 2004. Support vector data description. Machine learning 54 (1), 45–66.

Utkin, L. V., Zhuk, Y. A., 2014. Imprecise prior knowledge incorporating into one-class classification. Knowledge and information systems 41 (1), 53–76.

Van Vlasselaer, V., Meskens, J., Van Dromme, D., Baesens, B., 2013. Using social network knowledge for detecting spider constructions in social security fraud. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. ACM, pp. 813–820.

Wang, W., Zhang, W., 2008. An asset residual life prediction model based on expert judgments. European Journal of Operational Research 188 (2), 496–505.

Yesuf, A. S., Wolos, L., Rannenberg, K., 2017. Fraud risk modelling: requirements elicitation in the case of telecom services. In: International Conference on Exploring Services Science. Springer, pp. 323–336.