Bachelor Project

Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Computers

Comparison of anomaly detection techniques

Lev Kolomazov

Supervisor: Dmytro Shykhmanter, Ing.
May 2019

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kolomazov** Jméno: **Lev** Osobní číslo: **452739**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Comparison of anomaly detection techniques

Název bakalářské práce anglicky:

Pokyny pro vypracování:

The motivation of the project is to study and compare anomaly detection techniques [1].
The practical implementation will be performed on an open source credit card fraud dataset[4]. It will consist of developing a tool for visualising correlations between anomalies and frauds[3].
1. Student will perform a comparative analysis on available anomaly detection techniques.
2. For the most popular techniques, the student will develop a method to visualise
1. techniques' effectiveness,
2. correlation between anomalies and frauds.
3. Compare and assess the trade-off between different techniques using the developed tool based on the correlation between anomalies and frauds.

Seznam doporučené literatury:

1. PATCHA, Animesh. An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks [online]. 2007, 2007(Volume 51), 23 [cit. 2019-01-22]. Dostupné z: https://www.sciencedirect.com/science/article/pii/S138912860700062X
2. SALIMA, Omar. Machine Learning Techniques for Anomaly Detection: An Overview. International Journal of Computer Applications [online]. 2013, 2013(Volume 79), 7 [cit. 2019-01-22]. Dostupné z: https://pdfs.semanticscholar.org/0278/bbaf1db5df036f02393679d485260b1daeb7.pdf
3. Davidson, Ian. (2019). Anomaly Detection, Explanation and Visualization Introduction to Anomaly Detection. Dostupné z: https://www.researchgate.net/publication/265495904_Anomaly_Detection_Explanation_and_Visualization_Introduction_to_Anomaly_Detection
4. Open Source Dataset at kaagle. https://www.kaggle.com/mlg-ulb/creditcardfraud

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Dmytro Shykhmanter, Blindspot Solutions

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **13.02.2019** Termín odevzdání bakalářské práce: **24.05.2019**

Platnost zadání bakalářské práce: **20.09.2020**

Ing. Dmytro Shykhmanter podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Ing. Pavel Ripka, CSc. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. _____ _____

Datum převzetí zadání Podpis studenta

Acknowledgements

This work would not have been possible without all the work people at Czech Technical University in Prague do and financial support from the government of Czech Republic.

First of all, my thanks to Doc. Ing. Jiří Vokřínek, Ph.D., the supervisor of SIT FEL for helping me with official assignments.

Secondly, I would like to thank Ing. Ondřej Vaněk, Ph.D. for willing to work with me, proposing me a wide variety of topics and meeting me with Dmitrij.

I like to thank the supervisor of this project - Ing. Dmitrij Sichmanter. He's been there to help me during this whole time I've been developing. Dmitrij has supplied me with studying materials, we've had very interesting and insightful meetings.

I would also like to thank opponent of this thesis Ing. Jiří Šebek for consulting me with the assignments at the initial stages.

Last but not least, I am grateful to my family for giving me the opportunity to study at CTU and supporting me all the way through my life.

Declaration

I declare that this work is all my own work and I have cited all sources I have used in the bibliography.

Prague, May 20, 2019

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 20. května 2019

Abstract

Rapid technology progress introduces real-life tasks to the Internet. In commerce, with the growing number of credit card transactions, it becomes more and more important to be able to successfully detect the suspicious points in the data in order to maintain privacy and security. This thesis sets the goal of comparative analysis of anomaly detection techniques. It consists of techniques' overview, with a detailed explanation of those, that are proved to be the most effective and comparison practical analysis. The practical analysis part is performed on an open-source credit card fraud dataset. As a part of the assignment, there has been developed a tool for visualizing the results of the analysis. Not only it visualizes each technique's effectiveness but also focuses on correlations between outliers and actual frauds.

Keywords: Anomaly detection, Data Analysis

Supervisor: Dmytro Shykhmanter, Ing. E-mail: dmitrij.sichmanter@blindspot.ai

Abstrakt

Rozvoj technologii přináší do Internetu stále více úkolů. V sféře komerci je moc důležité umět rozpoznávat podezřelé případy pro účel udržování bezpečnosti soukromí. Tato bakalářská práce staví cíl porovnat existující metody detekci anomálii. Skládá se ze dvou částí: přehledu obecných metod a jejích nejúspěšnějších prvků, a praktického porovnání. Praktická část byla vyplněna nad veřejně dostupnými daty podvodů v oboru kreditních karet. Jako součást zadání byl vytvořen nástroj na vizualizační analýzu dat a porovnání metod detekci anomálii.

Klíčová slova:

Překlad názvu: Porovnání metod detekci anomálie

Contents

Figures

Tables

Chapter 1

Introduction

With the rising integration of technologies into the everyday life of people, the need for secure and safe storage, usage and transport of the information rises as well.

Nevertheless, it is a very complex task to maintain the integrity of the information at all times, even for well-known big companies. Failures happen even to the best of the market's players. In the e-commerce field, the aspect of safety and integrity of the data becomes even more crucial. These businesses rely on their reputation heavily [11].

Consequently, it is important to be able not only to react to the security leaks that have already happened but also to anticipate probable security leaks in order to prevent safety. Anomalies, or outliers, are in many cases the first indicators of a possible fraud case. There are multiple ways of detecting anomalies. It is important to choose the right detection technique that would do best for a specific goal.

This thesis focuses on fraud detection in the credit card transactions domain. The main goal of the thesis is to compare available anomaly detection techniques, evaluate them on the existing dataset and analyze the results. Since the fraudsters tend to simulate the behavior of a normal user, another interesting point to discover is the correlation between the statistical anomalies in data and actual positive fraud results.

In other words, the question: "When an anomaly is a fraud ?" is discussed in the thesis as well.

The specifics of the domain are that the research data are not easily obtainable due to privacy reasons, thus dataset used in this thesis is an open source dataset, on which the PCA transformation has been performed. Thus, feature analysis is not possible to be done. However, it is a labeled dataset so the evaluation can be done.

Chapter 2
Domain

To introduce a reader into the domain, I find it reasonable to list out the basic concepts of the domain.

2.1 Anomaly

It is rather a broad question - what is an anomaly (outlier) - but the widely accepted definition is given by David Hawkins: "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". [1]

Consequently, we can define three context-specific kinds of outliers (i.e anomalies).

2.1.1 Point Anomalies

Individual data instance can be considered as anomalous if it deviates significantly from the rest of the dataset [15]. Most of the techniques aim to identify point anomalies (global outliers).

An example from fraud detection domain is a transaction with large amount. However, such anomalies are rarely of fraudulent nature, since fraudsters do not want to be discovered. Following this logic, transactions of a small amount are more likely frauds.

Despite that, Chan [11] state that verifying the fraudulence of suspicious transaction with small amount can be not worthwhile the costs and thus is often not done.

2.1.2 Contextual anomalies

If a data instance is anomalous in a specific context, i.e it deviates significantly with from its context, it is called a contextual anomaly [15].

The notion of a context is specified as a part of the problem formulation. It is vitally important to correctly define context if the goal is to detect contextual anomalies. Consequently, context is almost always defined by domain experts.

As a rule, each data instance is defined using following two sets of attributes:

- Contextual attributes.

  Contextual attributes are used to determine the context (or neighborhood) for that instance.

  In the credit card fraud domain, such attributes may be time or location of the transaction.

- Behavioral attributes.

  The behavioral attributes define the non-contextual characteristics of an instance.

  Attributes such as the amount of transaction or receiver represent behavioral attributes in the credit card fraud domain.

The anomalous behavior is determined using the values for the behavioral attributes within a specific context. A data instance might be a contextual anomaly in a given context, but an identical data instance (in terms of behavioral attributes) could be considered normal in a different context. This property is key in identifying contextual and behavioral attributes for a contextual anomaly detection technique. [3]

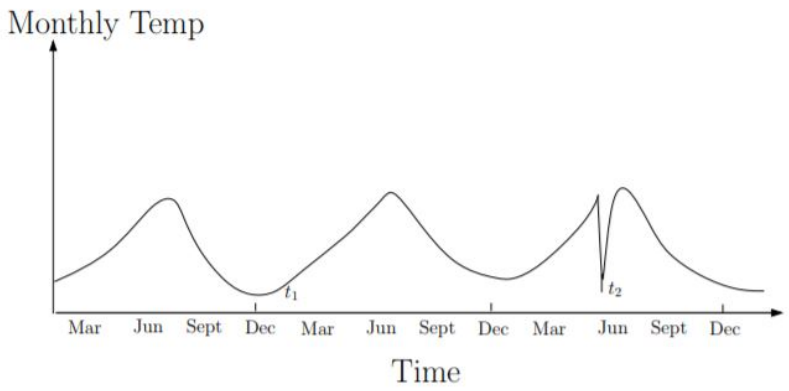

Figure 2.1: Contextual anomaly t_2 in a temperature time series. Note that the temperature at time t_1 is same as that at time t_2 but occurs in a different context and hence is not considered as an anomaly. Source: [3]

Han [15] provides an example when contextual outliers in credit card fraud detection lead to new business opportunities:

"Consider customers who use more than 90% pf their credit limit. If one such customer is viewed as belonging to a group of customers with low credit limits, then such behaviour may not be considered an outlier. However, similar behaviour of customers from a high-income group may be considered outliers if their balance often exceeds their credit limits. Such outliers may lead to business opportunities - raising credit limits for such customers can bring bew revenue "

Thus, anomaly detection may not only prevent harmful consequences, but also introduce new business decisions.

2.1.3 Collective Anomalies

If a collection of related data instances is anomalous with respect to the entire dataset, but the individual values are not anomalous, the collection is called Collective Anomalies.

In credit card fraud domain collective anomalies may be a group of delayed transactions. Normally, if a transaction gets delayed, it is not an outlier. But if there is a number of such transactions, it might raise suspicions.

Another example - consecutive numerous transactions of a higher amount.

It should be noted that while point anomalies can occur in any data set, collective anomalies can occur only in data sets in which data instances are related. In contrast, the occurrence of contextual anomalies depends on the availability of context attributes in the data. A point anomaly or a collective anomaly can also be a contextual anomaly if analyzed with respect to a context. Thus a point anomaly detection problem or collective anomaly detection problem can be transformed into a contextual anomaly detection problem by incorporating the context information [3].

Normally, the anomalies are expected to follow the Pareto principle [2]. Thus, most of the anomalies can be observed with just a few static detection rules. These rules are usually described by the domain expert. However, static rules tend to be too complex and too specific to be managed in real-world domains. Especially in ever-changing domains, such as the credit card fraud domain [3]. Therefore, this gap can be filled with automation - using machine learning approaches for detecting anomalies.

2.1.4 Novel patterns

Novel patterns are patterns in data, that are normal but have not been observed yet [15]. It is important for an anomaly technique to be able to distinguish between novel patterns and anomalous behavior.

However, novel patterns are normally found when some new data is added to the existing dataset, so in such cases like ours, when we have just one dataset, it is not possible to identify which patterns are indeed novel.

2.2 Data

Gogoi et al. [17] state that:

"The labels associated with a data instance denote if that instance is normal or anomalous. Labeling is often done manually by a human expert and hence requires substantial effort to obtain the labeled training data set.

Typically, getting a labeled set of anomalous data instances which cover all possible type of anomalous behavior is more difficult than getting labels for normal behavior. Moreover, the anomalous behavior is often dynamic in

nature, e.g., new types of anomalies might arise, for which there is no labeled training data."

In our case, the obtained data is labeled, so we can perform supervised as well as unsupervised learning, and compare the results.

The classes of the dataset are highly imbalanced. A whole series of issues may arise [12]. The possible workaround is undersampling the majority (non-anomalous) class in training data or generating synthetic anomalies. We will use the first workaround and compare it with training on raw data.

Generally, anomaly detection techniques can operate in one of the following modes.

2.3 Supervised anomaly detection

Supervised techniques assume the availability of a labeled training dataset. Usually, when people report frauds on their bank accounts, the appropriate data instances are labeled automatically. In some cases, the dataset is examined and labeled manually [15].

After the data is labeled, the problem is reduced to classification problem and classification methods can be applied.

There are multiple problems with this approach. The major one is that obtaining a labeled dataset is a cost. The higher quality data is desired, the more resource it takes [18].

The other problem is weak robustness. Supervised models are easy to overfit since they are learned to recognize some pattern, but novelties is usually a challenge for them.

The counterweight for those problems is a fact that supervised techniques are usually very effective in detecting anomalies of common patterns.

2.4 Unsupervised anomaly detection

Techniques that operate in unsupervised mode do not require labels for training data, and thus can be applied more robustly.

The techniques in this category make the implicit assumption that normal instances are far more frequent than anomalies in the test data. Consequently, the normal instances form some kind of a recognizable pattern [15]. If this assumption is not true then such techniques show a high false alarm rate.

The benefits of unsupervised techniques are: *(i)* no need of labeled data, and, consequently - lower costs for obtaining essential train data, *(ii)* by design, they are more efficient at working with the unseen before data, and *(iii)* they are less prone to overfitting.

However, the disadvantage of the unsupervised techniques is that they are usually not as efficient as supervised techniques are.

2.5 Output of Anomaly Detection

An important aspect of any anomaly detection technique is the manner in which the anomalies are reported.

Different techniques produce different outputs.

KNN and Decision Tree techniques produce labels for each data instance - normal or anomalous, and probability estimate of belonging to the anomalous class.

Isolation Forest technique produces a special score for each data instance. The decision if the instance is anomalous is then based on the chosen threshold.

Clustering techniques group data instances into clusters and their product are clustered data.

Thus, the goal of the analyst is not only to correctly select the parameters of the techniques, but also to choose the appropriate threshold and interpret the output of the techniques in the most optimal way.

We provide a more detailed explanation of the techniques in Chapter 3.

2.6 Credit Card Fraud Detection

In the domain of credit card fraud detection, the methods which are natural to utilize are classification and clustering [15].

Ensembles of anomaly detection techniques are applied to detect fraudulent credit card applications or fraudulent credit card usage (associated with credit card thefts).

The data typically consists of records defined over several dimensions such as the user ID, the amount spent, the time between consecutive card usage, etc [15]. In our case, the data have been anonymized with the PCA algorithm. The frauds are typically reflected in transactional records (point anomalies) and correspond to purchase of items never purchased by the user before, high rate of purchase, an unusual second side of transaction and more.

In real-world applications, banks and financial institutions adjust the techniques for groups of clients individually, creating profiles of users. These profiles are based on a user's credit card usage history. Any new transaction is compared to the user's profile and flagged as an anomaly if it does not match the profile. This approach is typically expensive since it requires querying a central data repository, every time a user makes a transaction. Another approach known as by-operation detects anomalies from among transactions taking place at a specific geographic location. Both by-user and by-operation techniques detect contextual anomalies. In the first case, the context is a user, while in the second case the context is the geographic location [3].

Chapter 3

Detection Techniques

This chapter will give a comprehensive representation of each learning method with an appropriate technique.

3.1 Supervised techniques

Classification is used to learn a model from a set of labeled data instances and then, classify a test instance into one of the classes using the learned model [13]. Classification based anomaly detection techniques operate in a similar two-phase fashion. The training phase trains a classifier using the available labeled training data. The testing phase classifies a test instance as normal or anomalous using the classifier.

Classification based anomaly detection techniques operate under the following general assumption:

A classifier that can distinguish between normal and anomalous classes can be learnt in the given feature space [3] .

There are many supervised learning methods. We provide two widely used techniques: K-th nearest neighbor and Decision Tree.

3.1.1 K-th nearest neighbor

K-th nearest neighbor (here and later also referred as KNN techniques are based on the following assumption:

Normal data instances occur in dense neighborhoods, while anomalies occur far from their closest neighbors.

The neighbor of a data instance A can be defined in multiple ways. Essentially, it is another data instance B, the features' values of which are close to the features' values of A.

Nearest neighbor based detection techniques introduce the notion of distance between two instances. For the numerical attribute space datasets (such

as our's) the distance can be computed differently, but most commonly Euclidean distance is used. Distance's goal is to measure the similarity between the data instances. [15]

The k in the technique states for a number of the closest instance the technique should look for. Poor choice of parameter k can lead to overfitting or underfitting the model. Thus, k is normally chosen by simply trying some values, comparing the scores and finally choosing the best option [15].

3.1.2 Decision tree

Decision tree techniques are tree-like structures, where the inner nodes perform the test on one feature, and labeled leaves perform the final classification. Such techniques are fast and easy to observe.

The essence of the method is that starting from the root node, the instances are divided binary using a certain test (in case of two labeling classes) so that they are separated best. This process is recursive for all the features until the separation is no longer valuable.

It results into dividing the initial data instances into mutually exclusive subgroups [14]. After the tree is grown, it can be overfit, so the pruning is often applied.

The new instances are then classified by walking the tree from the root to some of the leaves.

Studies show that Decision tree techniques can sometimes be superior to other supervised learning techniques in the credit card fraud domain. For instance, Sahin and Duman [5] show that Decision Tree based technique called C&RT is capable of observing 33% more frauds comparing to Support Vector Machines techniques.

3.2 Unsupervised techniques

As has been stated above, unsupervised techniques do not require the labels for the training phase. Even though the obtained dataset is labeled, we pretend that it is not, in order to compare the performances of the techniques. We briefly review three techniques: Isolation Forest, DBSCAN and K-means.

3.2.1 Isolation Forest

Isolation Forest (IF) is a fairly new technique that is fast in both memory and time spaces. The technique tries to isolate the instances by slicing the feature space, so it becomes isolated from other instances. It takes advantage of the fact, that normally, anomalies are *(i)* minority, thus there are fewer of them and *(ii)* anomalies are very different.

The isolation process is performed recursively until an instance is partitioned from all other instances. Typically, a random value between minimum and maximum across the dataset of a random feature is chosen as a test value.

[6]. So, anomalies are isolated close to the root, while normal instances are isolated deeper.

Liu, Ting, and others [6] define anomaly score for Isolation Forest as:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

where $h(x)$ is the path length of observation x, $c(n)$ is the average path length of unsuccessful search in a Binary Search Tree and n is the number of external nodes. This value can later be normalized.

Each data instance obtains an anomaly score and the following decision can be made depending on that score:

- Score close to 1 indicates anomalies
- Score close to 0 indicates normal instances
- If all scores are close to 0.5 there are no obvious outliers in the dataset.

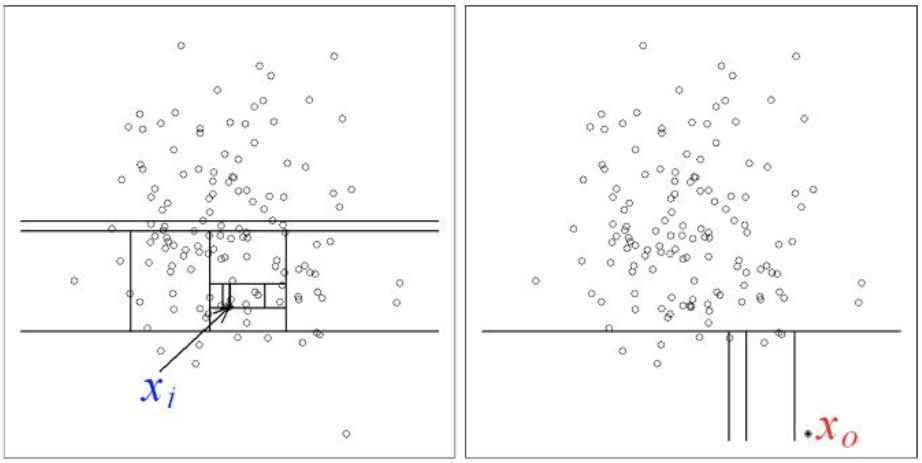

Figure 3.1: Isolation forest example. X_i is a normal instance, thus more slices is needed to categorize it, while X_o is an anomaly and less slices is needed to identify it. Source: [6]

Usually, because of vast class imbalance, the mean of the anomaly score across the dataset is much closer to zero than to one. The goal of the expert is to experimentally choose the most optimal threshold that would serve as a borderline between frauds and nonfrauds.

3.3 Clustering

The basic definition of cluster is following:

Cluster is a group of data objects.

Clustering detection techniques are based on partitioning the dataset into clusters so that objects inside one cluster are similar to one another and dissimilar to objects outside the cluster [15].

Clustering methods and algorithms are usually categorized into three categories.

First Category techniques base on the following assumption:

Normal data instances belong to a cluster in the data, while anomalies do not belong to any cluster.

These techniques just cluster the data and label instances, that do not belong to any cluster, as anomalous. An example of an algorithm from this category is DBSCAN.

DBSCAN (Density-based spatial clustering of applications with noise) is a robust clustering algorithm, introduced by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996 [20].

Generally, it's idea is to mark each instance that has at least n other instances in their ϵ neighborhood as *reachable*. Non-reachable instances are marked as noise (anomalies).

Second Category techniques base on the second assumption:

Normal data instances lie close to their closest cluster centroid, while anomalies are far away from their closest cluster centroid.

Techniques from this category first cluster the data. Then, for each data instance, it's anomaly score can be calculated as it's distance to the nearest cluster centroid. Techniques from this category have a disadvantage of not detecting anomalies, that do form their own cluster.

An example of an algorithm from this category is K-means clustering.

In K-means, initially k *means* data instances are chosen randomly. Then, k clusters are created, associating the data instances to their nearest k-mean. The centroid of each cluster becomes a new mean. The procedure repeats until the convergence is reached [15].

Finally, techniques from the third category base on another assumption:

Normal data instances belong to large and dense clusters, while anomalies belong to small or sparse clusters

Several clustering techniques require the notion of distance between the data instances. That makes them somewhat similar to nearest neighbor detection techniques. In the next chapter, we will test and compare the techniques.

Chapter 4

Data set analysis

In order to implement and compare detection techniques, there must be some data. The dataset used in this thesis is an open-source dataset obtained from widely known data science portal Kaggle [7]. The dataset has been anonymized and normalized with the PCA algorithm due to privacy reasons, so all the features have numerical values.

It contains 284 807 labeled credit card transactions, each transaction is represented by 28 features ($V1 - V28$), time, amount of transaction and a label class - 0 for non-frauds and 1 for frauds. The number of fraudulent transactions accounts only for 0.172% of all transactions (482 in total)

4.1 Basic Analysis

Transactions' time, class and amount features are presented in Table 4.1. We have' not included numerical features (V1 - V28) in the table, but their distributions can be seen in Figure 4.2

	Time	Class	Amount
mean	26.337	0.0017274	88.349
std	13.191	0.0415271	250.120
min	0.0	0.0	0.0
25%	15.055	0.0	5.6
50%	23.525	0.0	22.0
75%	38.700	0.0	77.164
max	47.997	1.0	25691.16

Table 4.1: Basic dataset description

If we look at the relative plot of time and amount of transaction in Figure 4.1, it is clear that the data is very dense in the region of the small amount with a handful of point anomalies in the area of large amount.

However, these point anomalies are just transactions with a high amount. The frauds are marked red, and it can be seen that fraudsters tend to imitate the behavior of normal users.

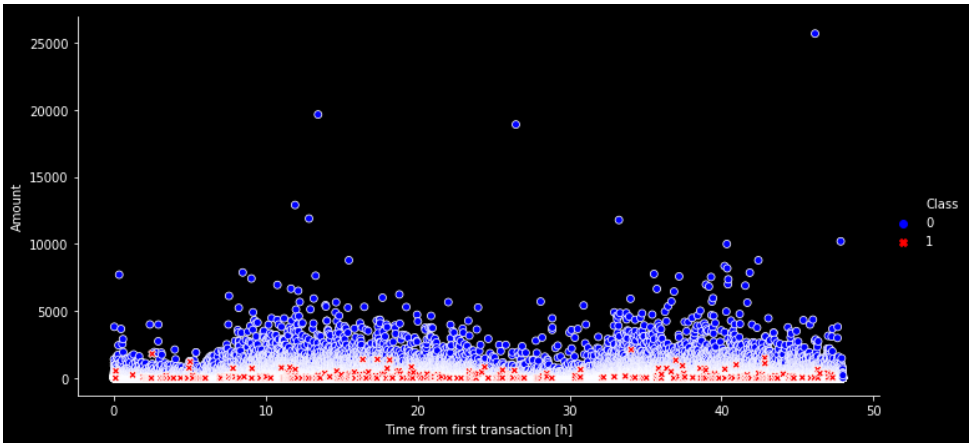

Figure 4.1: Relative plot of time and amount of transactions. X axis is time from the first recorder transaction in hours, Y axis is the ammount of transaction

In order to enhance techniques' performance quality, we normalize the feature values using a standard distribution scaler. The distributions of the dataset on which the techniques are going to operate are presented on Figure 4.2.

4.2 Subsampling

In our dataset, the number of nonfraud transactions is vastly larger than the number of fraud transactions. Assuming that most frauds are outliers, we can discuss, how supervised techniques can be improved in terms of detecting a handful of anomalies in a largely imbalanced dataset.

Subsampling is a method of alternating the train set. It is used to even the numbers of classes' instances to a certain degree. We will apply supervised techniques both with and without subsampling and compare the results.

Note: In the further chapters, when we say that some classifier is *undersampled*, it means that it has been trained on the undersampled train set.

Analogically, if we say that classifier is *original*, it means that it has been trained on the originally distributed train set.

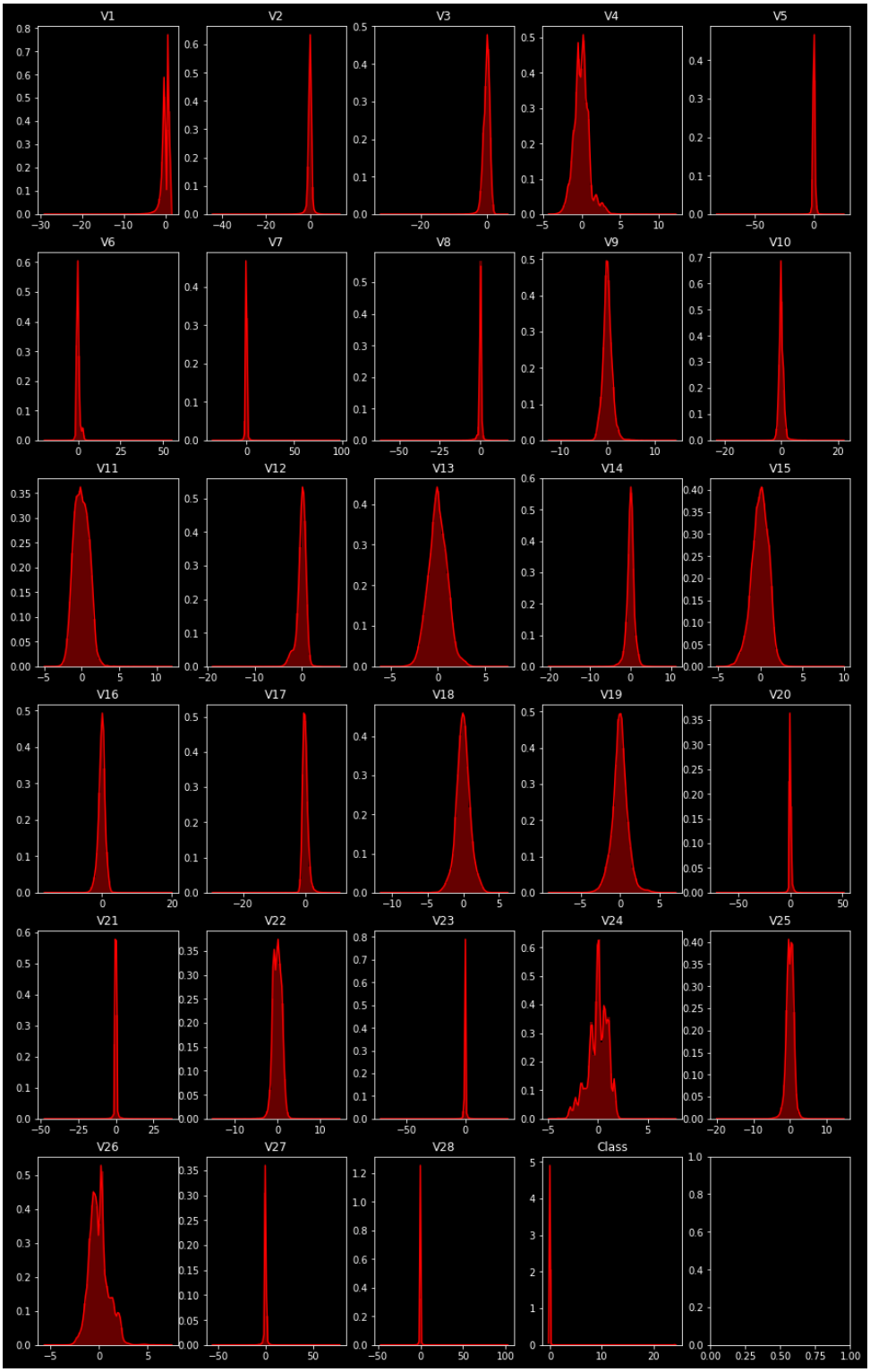

Figure 4.2: Distributions of features in dataset.

Chapter 5

Techniques' performance

In this chapter, we will apply the techniques described earlier, compare and analyze the results. Key questions that will be discussed in this chapter are:

1. Are supervised techniques superior to unsupervised or vice versa?
2. How does subsampling affect the performance of supervised techniques ?
3. Correlation of detected outliers and actual frauds.

5.1 Software stack

For purposes of applying the techniques we will use well-known data science technological solutions - Python programming language and following libraries:

- Pandas - for working with dataset
- Seaborn - for visualising purposes
- Scikit - for using the techniques
- Dash - for visual comparison tool

5.2 Applying techniques

In this section, we will apply the techniques, compare and analyze the results. Before we proceed to apply the techniques, we should define a metric that will be used for comparing the results. We'll use confusion matrices and F1 score as a comparable score of each technique.

5.2.1 Metrics

In order to compare the techniques, we need some metrics. For our domain the most important part is to be able to correctly recognize the frauds, thus the most relevant metrics for us would be:

- True positive rate (TPR), often also referred as *recall*

$$TPR = \frac{true\ positives}{true\ positives + false\ negatives}$$

- $F1$ score

$$F1 = \frac{true\ positives}{true\ positives + (false\ negative + false\ positives)\ /\ 2}$$

- false positive rate

$$FPR = \frac{false\ positives}{false\ positives + true\ negatives}$$

- Precision. This is a metric we'll utilize to analyze the correlation of anomalies and frauds

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

5.2.2 Supervised techniques

As has been stated above, supervised techniques work with labeled data and consist of two phases: training and testing. Our dataset is very unbalanced, so it makes sense to consider subsampling of the data for training the model. We will use two methods: undersampling and training on data with the same fraud ratio.
We will test the models on the subset of the initial dataset.

Since we would like to discuss the question, how does subsampling affect the model's effectiveness, we will do the following:

1. Construct test subset. It will be the same for testings.
2. Construct two train datasets. The first will be undersampled. The second will be constructed so that it preserves the fraud distribution of the original dataset. Each of these subsets must be mutually exclusive with the test subset.
3. For each technique, train both models separately.
4. Test both models using the test dataset.
5. Compare the results.

The constructed test subset contains 49 725 instances with 98 frauds and 49 627 nonfrauds. This distribution is very similar to the original dataset's one.

The original distribution train dataset contains 235 082 instances with 394 frauds and 234 688 nonfrauds. This distribution is very similar to the original dataset's one.

The undersampled train dataset contains 1477 instances with 394 frauds and 1 083 nonfrauds. The distribution of frauds is around 26 %.

Now we can proceed to train the models.

K-th nearest neighbor

In order for our models to be efficient, we've set estimator parameters using GridSearch[8] and trained them.

Model trained on undersampled dataset results are on figures 5.1 and 5.2.

	Predicted False	Predicted True
Actual False	49016	611
Actual True	10	88

Table 5.1: Confusion matrix for undersample KNN

	Value
TPR	0.898
F1	0.221
Precision	0.126

Table 5.2: Metrics for undersample KNN

Model trained on original distribution data results are on figures 5.3 and 5.4:

	Predicted False	Predicted True
Actual False	49623	4
Actual True	20	78

Table 5.3: Confusion matrix for original distribution KNN

	Value
TPR	0.796
F1	0.867
Precision	0.951

Table 5.4: Metrics for original distribution KNN

Analyzing the results, we can say, that the first model tends to label instances as frauds more often then the second one. Thus, it has recognized 10 more actual frauds, but at the same time marked much more nonfraud instances as frauds. So the first model gives a better recall and in our domain it is clear that it performs better, since frauds labeled as nonfrauds are worse than vice versa.

Decision Tree

Samely as for knn, best decision tree classifier has been selected with GridSearch[8] for each model.

Model trained on undersampled dataset results are on figures 5.5 and 5.6.

	Predicted False	Predicted True
Actual False	49317	310
Actual True	12	86

Table 5.5: Confusion matrix for undersample Decision Tree

	Value
TPR	0.878
F1	0.348
Precision	0.217

Table 5.6: Metrics for undersample Decision Tree

Model trained on original distribution data results are on figures 5.7 and 5.8:

	Predicted False	Predicted True
Actual False	49611	16
Actual True	22	76

Table 5.7: Confusion matrix for original distribution Decision Tree

	Value
TPR	0.776
F1	0.8
Precision	0.826

Table 5.8: Metrics for original distribution Decision Tree

Decision tree models trained on different datasets tend to make labeling similarly as knn models. But on contrary, decision tree model trained on original data does not show that much worse recall than the one trained on subsampled data. It is worth noting, that decision tree model turned out to be the fastest.

5.2.3 Summary

Analyzing results presented in Table 5.9, we can conclude, that although undersample KNN shows the best TPR, it also shows the highest FPR, thus, the classifier has gotten overfit. Undersample Decision Tree has gotten overfit as well.

Despite these facts, undersampled classifiers have detected more frauds than the original ones. So, if the nature of the problem requires classifiers to

	KNN Undersample	DT Undersample	KNN	DT
TPR	0.898	0.878	0.796	0.776
F1	0.221	0.348	0.866	0.8
FPR	0.0123	0.0062	0.00081	0.00032
Precision	0.126	0.217	0.951	0.826

Table 5.9: Summary metrics of supervised techniques

have higher TPR, undersampling can enhance that with the cost of higher FPR.

Taking a look at metric F1, we can conclude that the most *optimal* classifier is original KNN. It hasn't gotten overfit ant it detects almost 80 % of the frauds, with precision of 0.951.

5.2.4 Unsupervised techniques

Unsupervised techniques do not need labeled data to train, but some data is still required for the model fitting. For testing, we will use same dataset as in supervised techniques, and fit the model with the rest of the data.

Isolation forest

The important implementation note for Isolation Forest is that in scikit, standard IF classifier counts anomaly score like 0.5 - anomaly score. Thus, what is closer to 0, is considered to be more anomalous. [9]

Experimentally, we've found that 12 is the best number of estimators for the technique. As this parameter grow, technique shows better precision, but recall stays the same

	Predicted False	Predicted True
Actual False	48907	720
Actual True	28	70

Table 5.10: Confusion matrix for Isolation forest

	Value
TPR	0.714
F1	0.158
FPR	0.0145
Precision	0.0886

Table 5.11: Metrics for Isolation forest

Isolation forest classifier shows medium recall, but struggles of low precision. The reason for that might be that either there is a lot of outliers that are not

actually frauds or that the frauds are covered between normal instances. The situation is revealed in Section 6.3. Demonstration.

DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a robust technique for clustering the data and detecting the outliers at the same time.

It belongs to the first category of the techniques from [section 3.3 - Detection Techniques].

	Predicted False	Predicted True
Actual False	47617	2010
Actual True	24	74

Table 5.12: Confusion matrix for DBSCAN

	Value
TPR	0.7551
F1	0.0678
FPR	0.0405
Precision	0.0355

Table 5.13: Metrics for DBSCAN

DBSCAN shows relatively good recall, but at the same time lowest F1, precision and highest FPR score among the tested techniques.

The reason for that is that DBSCAN produces much more False Positives than all other techniques. It is caused, as we'll be able to discover via visualization tool, by the low relative density of some parts of the dataset. DBSCAN considers points that have relatively little number of neighbors around as noise and labels them as anomalies.

K-means

K-means clustering is another technique for clustering the data. It belongs to the second category from [section 3.3 - Detection Techniques].

The main difference here is that not only we choose the parameters for the technique (experimentally, 30 clusters turned to give best results), but we also select the top quantile of the instances sorted by distance from closest cluster centroid descending.

The precision-recall curve for k-means on different quantiles is on figure 5.1 . We've chosen the most optimal: $q = 0.98875$

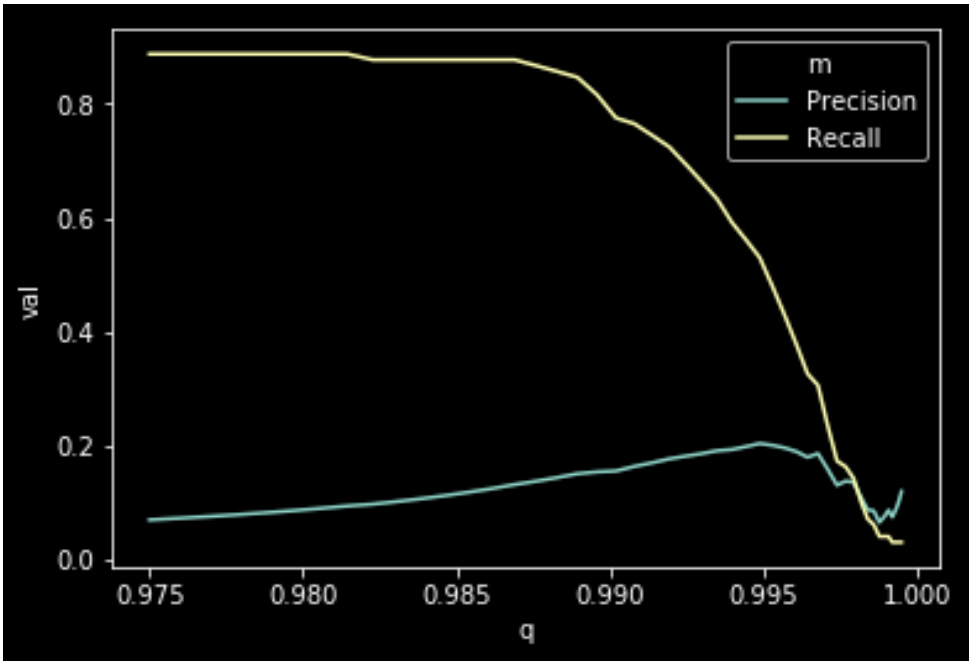

Figure 5.1: K-means precision-recall curve

	Predicted False	Predicted True
Actual False	49141	486
Actual True	24	74

Table 5.14: Confusion matrix for K-means

	Value
TPR	0.755
F1	0.225
FPR	0.0098
Precision	0.132

Table 5.15: Metrics for K-means

Summary

	Isolation Forest	DBSCAN	K-means
TPR	0.714	0.755	0.755
F1	0.157	0.068	0.225
FPR	0.014	0.0405	0.0098
Precision	0.088	0.0355	0.132

Table 5.16: Summary metrics of unsupervised techniques

Comparing unsupervised techniques, we can conclude that K-means and

DBSCAN turned out to have the same TPR equal to 0.755, with K-means having significantly lower FPR. Isolation Forest demonstrated lower TPR equal to 0.714 and FPR value between DBSCAN and K-means.

Thus, K-means is optimal technique to utitlize in real world if labeled dataset is not obtainable.

It must be noted, that in applications other than our's, another technique can show advantages as well.

5.3 Result comparison

	KNN original	K-means
TPR	0.796	0.755
F1	0.866	0.225
FPR	0.00081	0.0098
Precision	0.951	0.132

Table 5.17: Summary metrics of two most optimal classifiers

- **Supervised techniques perform better than unsupervised for this dataset.**

  Both knn and decision tree show better recall and precision than isolation forest or clustering methods. The explanation of this result is following: Models that are trained on labeled data are better at classifying the same pattern data.

  If the goal is to detect as much frauds as possible and labeled dataset is presented, *knn* undersample is a preferable option because of it's highest $Recall = 0.898$. Downside of that is high $FPR = 0.0123$.

  If mistakes ratio is crucial, undersampled classifiers are not suitable because of high FPR, so original KNN is the most optimal option with $Recall = 0.796$ and $FPR = 0.00081$.

  The metrics for the most optimal classifiers from each category can be found in Table 5.17.

- **Unsupervised techniques can indeed be applied in real world cases.**

  When labeled data set is not easily obtainable, unsupervised techniques can be relatively effectively utilized for anomaly detection.

  Comparing most optimal supervised technique *knn* and best unsupervised k-means clustering, it turns out that KNN detects 5% more frauds, but it has significantly lower FPR, thus makes less mistakes (Table 5.17).

So, while knn is more effective at detecting the frauds, k-means is still detecting a decent part of them.

- **Undersampling enlarges models' recall but lowers their precision.**

  Models that were trained on undersampled data tend to recognize more frauds. Downside of that is that they label more nonfraud instances as frauds.

  Undersampled KNN classifier detect 12.821 % more frauds than one trained on original distribution data.

  Same for decision tree classifier - increase is 13.16 %.

  On the other hand, undersampled classifiers mark more non frauds as frauds: 611 against 4 false positives for undersampled KNN vs original KNN and 310 against 16 for undersampled Decision Tree vs original Decision Tree.

  It is a question of application - which model is more successful - because costs of verifying transaction's fraudulence must be taken into account. However, very often technique with better recall counterweights.

  Thus, undersampling does help to detect more frauds, but causes models to overfit

- **Correlations of frauds and anomalies are dependent on the technique**

  The *Precision* metric, that we've been using among other metrics can be interpreted as a fraction of frauds to all anomalies.

  And for our dataset, precision metrics vary between techniques.

  DBSCAN technique has the lowest precision value, and thus, for it, frauds are just small subset of all the anomalies. There are only 74 frauds among 2084 anomalies.

  As precision grows, correlation becomes more and more notable. For KNN trained on dataset with original distribution (precision = 0.95122) there are 78 frauds among 82 detected anomalies.

  As a matter of fact, some frauds are not considered anomalous even among the most efficient techniques. For instance, knn trained on undersampled data has not considered 10 frauds anomalous, while having the greatest recall. The reason for that is that these frauds are mixed within normal instances.

  These results can be seen in Section 6.3 Demonstration

- **K-means is the most effective classifier among unsupervised.**

  K-means has showed the best F1 score detecting same number of frauds as DBSCAN while being more accurate. Surprisingly enough, Isolation Forest didn't show better results even after decent parameters adjustments.

  We'll take a closer look why in Section 6.3 Demonstration.

Chapter 6

Visual Comparison Tool

This chapter contains description and implementation details of the visual comparison tool developed as a part of the assignment.

6.1 Visualisation purpose

The purpose of visualizing data is to empower the interpretability of the models. It allows analyst to feel the data, understand it's nature. Consequence of data visualization can be enhancement of the techniques used upon the data. Visualised data also brings possibility of better mutual understanding between tech developers and management, opposed to nonvisual methods of data representation.

6.2 Developed tool

The tool, developed as a part of this thesis, allows analyst to have a visual representation of the dataset, used for testing the techniques discussed earlier, as well as to observe the performance of each technique. In particular, the tool is capable of showing the way the techniques are labeling instances. Consequently, analyst can adjust the parameters to reach a better tradeoff.

6.2.1 Features

The developed tool has following features:

- Visualizing dataset in 3D space.
- Selecting the technique.
- For selected technique, visualize it's performance using colourful markers in 3D space, as well as metrics and confusion matrix.

6.2.2 Implementation details

In order to represent the dataset in 3D space, we'd have to perform some dimension reduction routine on it. There are multiple methods, but we've

chosen PCA. Fodor, Imola [10] state that it's the best in the mean-square error sense.

It should be noted though, that we've used PCA on a dataset, which has already been used PCA upon, so the visualized result should be considered an exploration tool, rather than strictly analytical.

After the testing dataset has been reduced to three dimensions, the metrics' results labels were concatenated to it.

The tool layout is inflated with:

- 3D graph of the dataset.
- Side-block with marker additional information, available on marker click.
- Dropdown of available techniques.
- Information block with confusion matrix for chosen technique.
- Information block with metrics for chosen technique.

6.3 Demonstration

One of the main purposes of the developed tool is to reject or confirm general hypothesis about the dataset.

For example, in section Isolation forest, the technique detected a large number of false positives and thus obtained low precision score. Possible reasons include that either or both: *(i)* frauds are mixed within normal instances, and thus hardly recognizable *(ii)* great part of instances labeled as anomalous are not frauds

So, utilizing the tool and taking a look as on Figure 6.1,

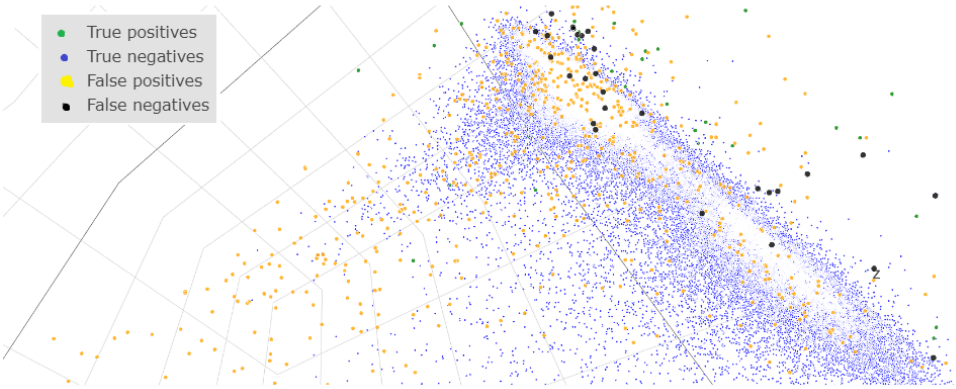

Figure 6.1: Tool demonstration on Isolation Forest.

we can conclude that both hypothesis have confirmation. It is clear, that Isolation Forest recognizes a lot of instances outside the dense cluster as

outliers (yellow dots), since less partitions is needed to get them isolated. However, a very small part of those instances are frauds. In fact, frauds (black and green dots) are mostly positioned inside this big cluster, so Isolation Forest needs more steps to isolate those instances, and thus marks them as normal. Probably, Isolation forest is not the best choice for this dataset.

Furthermore, preserving the point of view, we can observe, how did the other technique do, for example, KNN trained on originally distributed train set. (Figure 6.2).

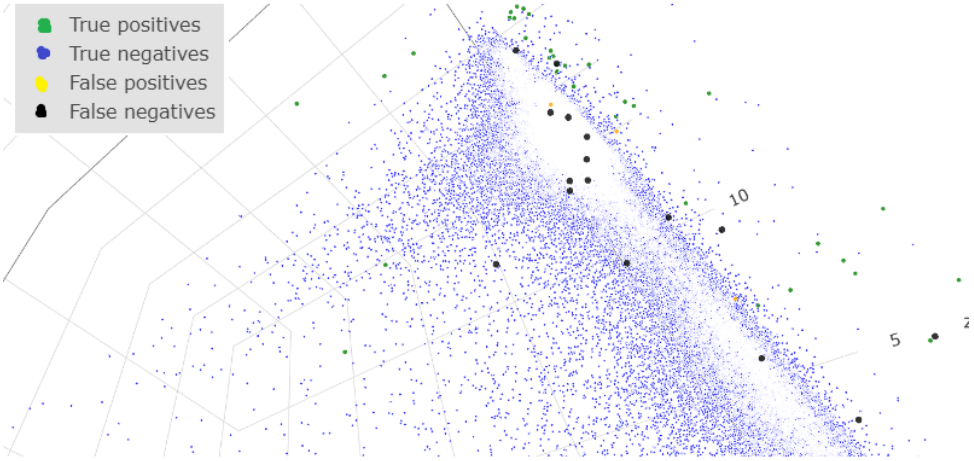

Figure 6.2: Tool demonstration on KNN original.

We can see, that KNN is way better at detecting the outliers that are not frauds (much less yellow markers), but still, some frauds remain undetected (black dots among blue), since they are mixed within normal instances.

More use cases of the tool include observing correlations of the frauds and anomalies for each technique.

Consider another example: sparse cluster at Figure 6.3. It's a point of view at one and the same data space combining 3 techniques.

While undersampled decision tree considers all data points anomalous and guesses half (yellow and green clusters), knn trained on original distribution train set is capable of separating the true negatives correctly (blue cluster), detecting almost all frauds(green dots), but makes two costly false negative mistakes (black dots).

DBSCAN performs labeling, as it is stated in it's description [19]. And fraudsters might have utilized this knowledge, in case they are aware that some bank is using DBSCAN based technique. In this case, black dots are positioned is such way, that they are chosen as core points by the algorithm and thus mistakenly labeled as non-frauds.

Another interesting look is to see how our two most optimal classifiers - knn original and k-means - perform on Figure 6.4. It can be visually seen why k-means struggles of higher FPR - there are lots of outliers that are not frauds.

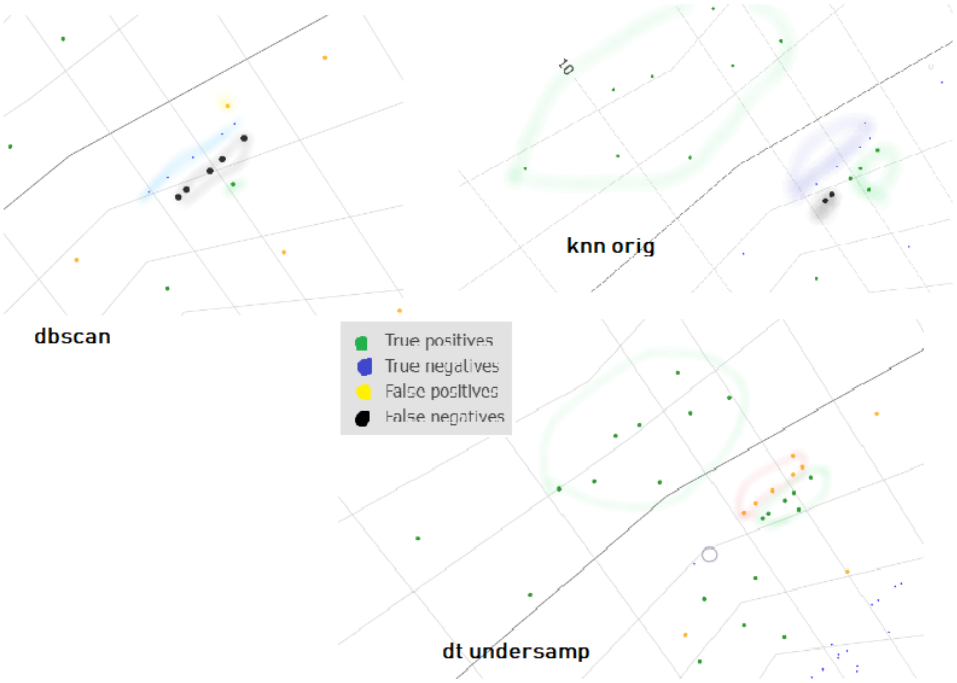

Figure 6.3: Sparse cluster of anomalies.

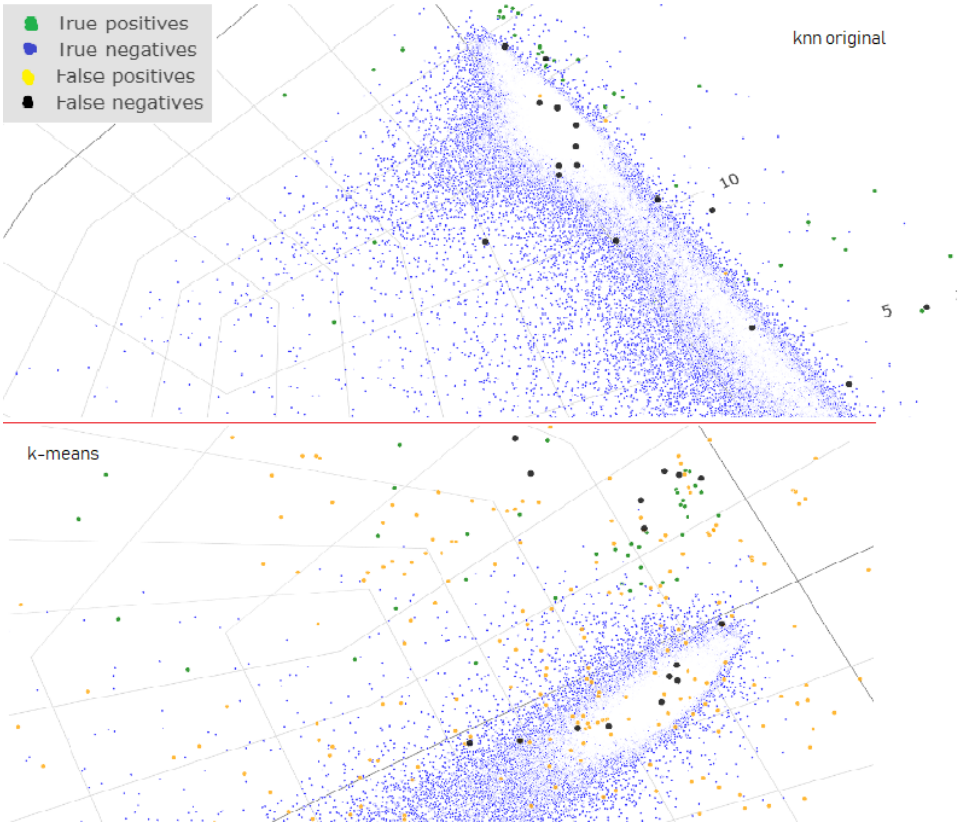

Figure 6.4: The most optimal classifiers. Top is knn undersample, bottom is kmeans

We can also observe the downside of using undersampled train set - overfiting the model. On Figure 6.5 there are presented two KNN classifiers - undersampled and original.

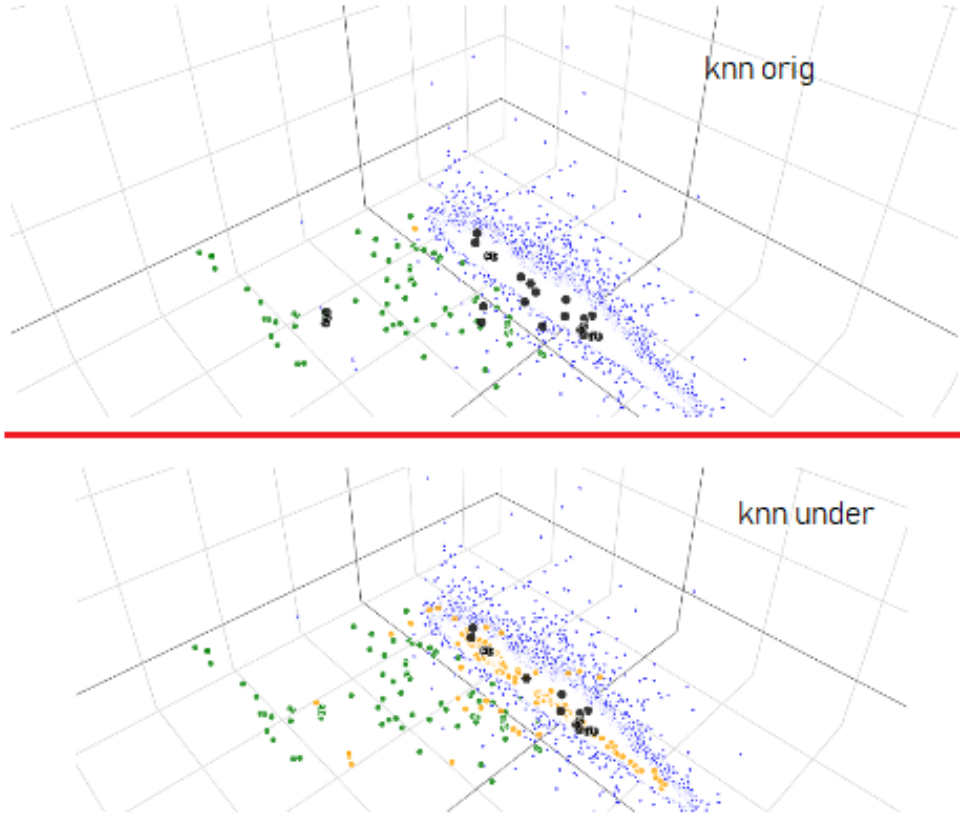

Figure 6.5: Result of undersampling overfitting the models. Top is original knn, bottom - undersample knn.

Chapter 7

Conclusion

All parts of the assignment were completed.

We have studied available anomaly detection techniques, both supervised and unsupervised, described their principles and proposed multiple examples for each category.

The comparative analysis showed, that supervised techniques perform better for the given dataset. Both KNN and decision tree classifiers turned out to outperform isolation forest and clustering techniques in terms of both recall and precision.

In terms of detecting the frauds, the best classifier between supervised techniques turned out to be Undersampled KNN ($Recall = 0.898$). However, undersample KNN showed relatively high FPR (0.012), so the optimal tradeoff is using original KNN with $Recall = 0.796$ and $FPR = 0.0008$.

Among unsupervised techniques, k-means clustering with $Recall = 0.755$ and $FPR = 0.0097$ is the most effective in terms of detecting frauds and optimal in terms of making mistakes classifier.

If there is no possibility of obtaining a labeled dataset, k-means clustering technique turned out to be effective. However, if the mistakes are costly, and it is important to keep the FPR low, unsupervised techniques can not do that (K-means FPR is 0.0098 comparing to original KNN's 0.00081) and using supervised techniques is mandatory.

Undersampling the training set turned out to enlarge supervised techniques' recalls, but lower their precisions.

Undersampled KNN classifier($Recall = 0.898$) detects 12.821 % more frauds than the one trained on originally distributed data. Same for decision tree classifier($Recall = 0.8775$) - increase is 13.16 %. Precision losses are corresponding: 7.56 times loss for knn undersample vs original and 3.804 times loss for decision tree undersample vs original.

Metrics and dataset analysis revealed that the correlation between anomalies and frauds is not defined by common rule, but it is rather a question of the specific technique. For techniques that have a higher precision metric value, the correlation is more obvious.

However, some fraud cases tend to imitate the behavior of the normal cases,

and thus they were not recognized by any technique.

Specifically, supervised techniques have shown a better correlation. KNN original has shown the highest correlation of frauds and anomalies: between discovered 82 anomalies there were 78 frauds (95%), for original decision tree there were 76 frauds of 92 anomalies (82.6%) in total.

Undersampling decreases the correlation, for undersampled knn the numbers are 88 frauds from 599 anomalies (14.7 %), and for undersampled decision tree it's 86 frauds from 396 anomalies (21.7%).

Unsupervised techniques have shown a worse correlation. The lowest correlation was observed for DBSCAN: 74 frauds of 2084 anomalies (3.42%), for Isolation forest it's 70 frauds from 790 anomalies (8.86%) and for k-means it's (13.21%).

The visual tool has been developed and it serves goals of gathering visual sense of the data and exploring the performance of the techniques. Correlation between anomalies and frauds can be explored in the tool as well.

Utilizing the visual tool, it is possible to verify the conclusions presented in this thesis.

7.1 Future work

There are lots of aspects, described briefly in the thesis. Further works may include a comparison of ways of subsampling the dataset, analyzing the behavior of the techniques online, i.e when new data arrives, generalization in case of multiclass datasets. All these topics can be covered in more detail.

There is also plenty of room to improve the techniques described above. Feature engineering would have helped to alter the techniques and domain experts could have succeeded to tune the parameters reaching the best outcomes.

Bibliography

[1] Hawkins, Douglas M. *Identification of outliers.* Vol. 11. London: Chapman and Hall, 1980.

[2] Hosking, Jonathan RM, and James R. Wallis. *Parameter and quantile estimation for the generalized Pareto distribution.* Vol. 11. London: Technometrics (1987).

[3] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. *Anomaly detection: A survey.* ACM computing surveys (CSUR) 41.3 (2009).

[4] Breunig, Markus M., et al. *LOF: identifying density-based local outliers.* ACM sigmod record. Vol. 29. No. 2. ACM, 2000.

[5] Şahin, Yusuf G., and Ekrem Duman. *Detecting credit card fraud by decision trees and support vector machines.* 2011

[6] Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou *Isolation forest.* IEEE, 2008.

[7] GROUP, M. L. *Scikit-learn: Machine Learning in Python* Kaggle, 2015. Avaliable from: https://www.kaggle.com/mlg-ulb/creditcardfraud

[8] Pedregosa, Varoquaux, Gramfort A., et al. *Scikit-learn: Machine Learning in Python* 2011. Avaliable from: https://scikit-learn.org/stable/index.html

[9] Pedregosa, Varoquaux, Gramfort A., et al. *Scikit-learn: Machine Learning in Python* 2011. Avaliable from: https://scikit-learn.org/stable/index.html

[10] Fodor, Imola K. *A survey of dimension reduction techniques.* No. UCRL-ID-148494. Lawrence Livermore National Lab., 2002.

[11] Chan, Philip K., et al. *Distributed data mining in credit card fraud detection.* IEEE Intelligent systems 6 (1999).

[12] Chawla, Nitesh V., Nathalie Japkowicz, and Aleksander Kotcz.*Special issue on learning from imbalanced data sets.* ACM Sigkdd Explorations Newsletter 6.1 (2004)

[13] Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. *Association analysis: basic concepts and algorithms.* Introduction to Data mining. Addison-Wesley, 2005.

[14] Aggarwal, Charu C. *Data classification: algorithms and applications.* CRC press, 2014.

[15] Han, Jiawei, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques.* Elsevier, 2011.

[16] Singh, Karanjit, and Shuchita Upadhyaya. *Outlier detection: applications and techniques.* International Journal of Computer Science Issues (IJCSI) 9.1 (2012).

[17] Gogoi, Prasanta, Bhogeswar Borah, and Dhruba K. Bhattacharyya. *Anomaly detection analysis of intrusion data using supervised unsupervised approach.* Journal of Convergence Information Technology 5.1 (2010).

[18] Weiss, Gary M., and Foster Provost. *Learning when training data are costly: The effect of class distribution on tree induction.* Journal of artificial intelligence research 19 (2003): 315-354.

[19] Schubert, Erich, et al. *DBSCAN revisited, revisited: why and how you should (still) use DBSCAN.* ACM Transactions on Database Systems (TODS) 42.3 (2017).

[20] Ester, Martin, et al. *A density-based algorithm for discovering clusters in large spatial databases with noise.* Kdd. Vol. 96. No. 34 (1996).

Appendix A

Contents of the CD

The attached CD contains electronic version of this thesis in PDF format, the source code of the techniques' evaluation routine, dataset visualization and the developed tool.

Installation can be done in a simple manner: using anacondas environment (preferable) or by installing Python and required libraries: jupyter, scikit, numpy, pandas and dash.

The additional details on installation and running can be found in readme.txt file.