FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

ASSIGNMENT OF BACHELOR'S THESIS

Title:	Design of anomaly detection for stock market trading
Student:	Aleksandr Karpenko
Supervisor:	Ing. Martin Kopp
Study Programme:	Informatics
Study Branch:	Computer Science
Department:	Department of Theoretical Computer Science
Validity:	Until the end of summer semester 2018/19

Instructions

The goal of the thesis is to adapt standard anomaly detection algorithm for the stock trading domain.

1) Convert stock market logs into a format suitable for anomaly detection.
2) Analyse available features and select the most promising for anomaly detection.
3) Implement simple anomaly detectors using selected features.
4) Investigate a method for the false positive rate reduction.
5) Investigate a method for speeding up the detection

References

Will be provided by the supervisor.

doc. Ing. Jan Janoušek, Ph.D.
Head of Department

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Dean

Prague February 6, 2018

Czech Technical University in Prague
Faculty of Information Technology
Department of Theoretical Computer Science

Design of anomaly detection for stock market trading

by

Aleksandr Karpenko

Bachelor degree study programme: Informatics

Prague, 15th May 2018

Supervisor:
Ing. Martin Kopp
Department of Theoretical Computer Science
Faculty of Information Technology
Czech Technical University in Prague
Thákurova 9
160 00 Prague 6
Czech Republic

Acknowledgements

I would like to thank my parents, friends and colleagues who has always been supporting me along the way.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46(6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the "Work"), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity.

In Prague on 15th May 2018 .....

Abstrakt

Detekce anomálií v datech z akciových trhů v reálném čase je skutečně náročná úloha. Tato práce představuje první veřejný výzkum v této oblasti, který se nezaměřuje na cenu akcií. Nástroj pro konverzi záznamů obchodních dat ve FIX formátu do CSV formátu, vhodného pro další analýzu, byl vytvořen a použit v průběhu výzkumu. Příznaky jsou extrahovány z dat, analyzovány a klasifikovány pomocí různých technik detekce, a porovnána kvalita jednotlivých modelů. Na základě tohoto měření je vybrán model, který je nejoptimalnější vzhledem k četnostem false positive a true positive. Dále byly otestovány metody na zkrácení odezvy, které se ukázaly nepraktické, pokud jsou použity samostatně, a vyžadují ještě další zpracování. Navržený model na detekci anomálií je dostatečne přesný, že může být použit v praxi.

Klíčová slova:
FIX, detekce anomalií, tradování, HBOS.

Abstract

Detection of anomalies in trading data flow in real-time is a challenging task. This thesis presents the first public research in that area, which does not focus on stock price. A tool converting FIX traffic logs into CSV format, which is suitable for analysis, is developed and used during the research. Features are extracted from the data, analyzed and classified

with various anomaly detection techniques applied, and performance of different models is measured. Model which performs best in terms of false positive rate and true positive rate is proposed. Response time reduction methods are investigated and proven to be impractical when used alone and require additional post-processing. As a result, brokers can use proposed model to detect anomalies with high confidence.

Keywords:
FIX, anomaly detection, trading, HBOS.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Stock trading is a complex topic involving a lot of risks and challenging problems. One of the open issues for the brokers is real-time anomaly detection. They need to have an opportunity to react to an issue before it becomes overwhelming. It's not feasible to have a person monitor the trading and rely on his/her judgment, because there is too much data to follow. Human error is highly probable, and cost of the mistake can be substantial. Therefore, being able to process and analyze anomalies automatically provides significant value for broker companies.

This thesis deals with trading data and application of anomaly detection techniques on the trading data in real time.

Goals of the Bachelor thesis

1. Develop a tool to convert logs in the form of data in FIX format into a format suitable for analysis.
2. Identify features suitable for anomaly detection.
3. Investigate different anomaly detectors and build a model which provides the best performance
4. Investigate techniques for lowering the reaction time.

Thesis organization

1. *Introduction*: Introduces the problem, goals and thesis organization
2. *Theoretical Background*: Introduces the reader to the necessary theoretical background, surveys the current state-of-the-art and describes the tools used.
3. *Data*: Explains the data, its format, features extracted and their preprocessing
4. *Experiments*: Describes conducted experiments

5. *Conclusions*: Summarizes the thesis and suggests possible topics for further research.

Chapter 2

Theoretical Background

In this chapter, the problem is explained in details, and related works are discussed.

2.1 Trading Theoretical Background

2.1.1 Key definitions

Stock (security, instrument) is an item describing the ownership of a company. **Share** is a specific certificate of a particular company stock ownership.

Clients (called buy and sell sides) are entities which are performing operations on stocks.

Broker is an entity representing its clients (buy or sell sides) on stock exchanges (or trading there themselves).

Stock exchange market is a company which is entitled to keep a record of events happening with company's shares, manage activities of buying and selling them. Stock exchanges do not allow arbitrary entities to trade on their venue. Thus, in order to buy or sell shares on some particular exchange, an entity has to be either registered directly on the stock exchange or use services of brokers which are registered on the exchange.

2.1.2 Investing and Trading

Clients of brokers can trade or invest in stocks via brokers, using them as a proxy to trade on some stock exchange. Trading or investing is a process of buying and selling shares from stock exchanges, so in the end, all the financial intermediaries in the trading are representatives of an end user. It is generally not convenient for companies to be registered on the stock market by themselves, as it requires licensing and extra responsibilities. Therefore brokers provide services to multiple clients at the same time, generally also providing access to several markets.

Fig. 2.1 shows the example of a possible trading structure with multiple brokers serving different clients for trading on multiple stock exchange venues.

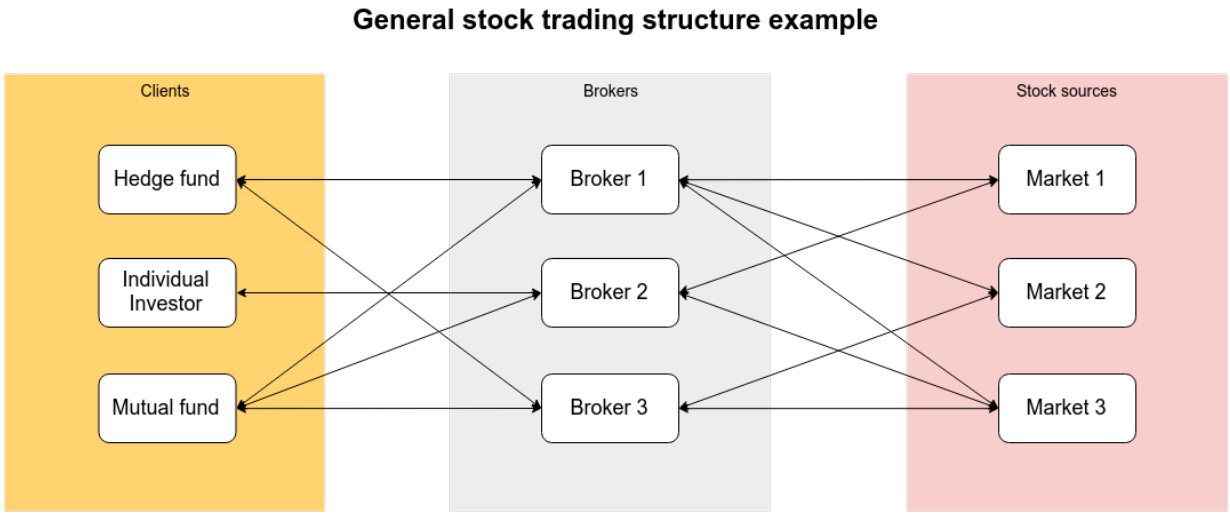

Figure 2.1: Example trading ecosystem.

2.1.3 Regulations

Trading is highly regulated domain. Depending on the state, the regulations are different, but generally, they aim to improve transparency of the market and protect investors.

In Europe, Markets In Financial Instruments Directive (MiFID) is an important regulation law. After the financial crisis of 2008, a new directive (MiFID 2 [Eur14]) was proposed to cover the reasons the crisis happened, to protect market participants from market abuse and make the process of investigation in case an issue arises easier. It became active on 3d January of 2018 [SA17].

2.1.4 Trading process

Trading happens by two parties exchanging messages. There are two kinds of messages: order handling and utility messages. Among utility messages, there are messages, such as Logon, Heartbeat, Logout, Resend Request, etc. More relevant for us are order handling messages.

New Order Single is an order message which indicates client willing to submit securities to a broker for execution electronically. In case the client wants to change the parameters of the existing order, for example, change the quantity it wants to buy or sell, the message **Order Cancel/Replace Request** is sent, so it is a *modifying* order. The message which requests the cancellation of the order is called **Order Cancel Request**. The way the other party replies to the client is mainly done via **Execution Report** message. Using this message a party indicates the follow-up action on order, such as confirmation of receiving, buy or sell actions, rejections and conveys the information about the current state of the order. When the order to buy or sell some stock matches with the

	Condition true	Condition false
Prediction true	TP	FP
Prediction false	FN	TN

Table 2.1: Relation between test condition and outcome

counter offer, *trade* happens (and corresponding Execution Report is sent). Trades can be *partial fill* or *fill*, where the former means that not all the quantity of the order was taken and latter indicates that someone took the entire order. All fills trades carry the price for the stock, and the amount of it. Trades can be canceled by the counterparty, which is called *trade bust* [Ltd17].

2.2 Statistics

There are measures which can be calculated in order to estimate the performance of the classifier, in our case, anomaly detector. **True Positive (TP)** is when sample is an anomaly and detector correctly identifies it. **False Positive (FP)** is when sample is not an anomaly, but we decide that it is. **True Negative (TN)** is when sample is not an anomaly and detector does not label is as such. **False Negative (FN)** is when sample is an anomaly, but detector decides that it is not. In this work we will be using the following measures to estimate the performance of models. **False Positive Rate (FPR)**, which is the proportion of incorrectly categorized samples as anomalies to all non-anomalous samples and is defined as $FPR = \frac{FP}{FP+TN}$; **True Positive Rate (TPR)** which is defined as $TPR = \frac{TP}{FN+TP}$ and shows the proportion of correctly identified anomalies to all actual anomalies; and **False Discovery Rate (FDR)**, or: $FDR = \frac{FP}{FP+TP}$, which indicated a proportion of incorrectly identified anomalies to all samples detected as anomalous. Tab. 2.1 explains the relation between test conditions and test results [Jam+14], [HTF01].

2.3 Problem statement

Broker entities are monitoring trading activity during the day for the following reasons:

1. Comply with the rules of the market they provide access to
2. Comply with the regulations imposed by the law
3. Protect clients from software malfunction on their side or the market side

With the new European MiFID 2 regulations, the rules and requirements for brokers in Europe became more strict than before, and the need to comply with MiFID 2 rules makes the active monitoring of the trading activity even more important requirement for the business.

The monitoring is done in real time throughout the day. In case a broker has multiple clients trading simultaneously on different venues, it becomes hard to spot an anomaly happening manually. Since there are so many variables in place, person monitoring gets tired, makes mistakes and the process becomes error-prone. Hence arises the need for the software system which can help to automate identifying anomalies, and send a signal to the responsible person to manually check if it is valid alert and take action in case it is.

Identifying and reacting as quickly as possible on possible issues is a critical ability that brokers seek, so the used models are expected to be suitable for detection in real-time complexity and memory wise. Statistical models provide the best performance for real-time streamed data [CKO17], so their application will be researched.

2.4 Anomaly Detection

2.4.1 Theoretical background

Anomaly detection is a problem of finding samples in data which fall out of the expected behavior. Those outliers or patterns are called anomalies. Anomaly detection finds its use in many different, yet critical fields, such as medicine, fraud detection, cyber security, intrusion detection, networks and so on [CBK09]. In this work, we focus on the application of anomaly detection on trading data.

Anomaly detection is important due to the nature of the subject of the research, as an anomaly in the critical system may imply significant issue, for instance, an intruder in a computer system, or disease seen after some medical procedure [CBK09].

Detection of anomalies is a hard topic for a number of reasons. To picture a few, firstly, the boundary between an anomaly and normal behavior is often not obvious, especially for a human eye. Secondly, choice of a model for a particular problem highly depends on the domain. Therefore it is not trivial to apply a model from one domain to the other. For example, in stock trading, the calibration of a price is considered normal, while in medicine, even small deviations in body temperature are alarming. Thirdly, the behavior of a system often changes over time, combined with noise, it requires a model to be flexible enough to be able to adapt over time [CBK09].

There are three general techniques of anomaly detection. Supervised anomaly detection, where a model is learned from labeled data. Samples which falls out of the prediction are then considered anomalous. The main disadvantage of this approach is that the amount of anomalies is far less than the number of other samples. Therefore we lack data to train robust model; also it is very challenging to label the data, as it requires a lot of working hours of expert work. The second approach is semisupervised anomaly detection, which assumes the data has labeled points of normal behavior, so it is possible to build a model based on typical behavior and consider anomalies something that does not fit in it. It has the same disadvantages as the supervised technique. Finally, unsupervised anomaly detection which does not need any labels and the models expect the normal behavior to occur far more frequent than anomalies [GU16].

In the trading domain, acquiring the data is problematic in itself, because the data are very sensitive asset for each broker.

2.4.2 Threshold-based

A simple method, which checks if values are above (or below) some preset threshold. Point x is considered an anomaly if $x > \tau$, where $\tau \in R$, where R is a set of real numbers (or the other version $x < \tau$). The method can be used in case the only requirement for the data is to be in some known boundaries. Also, it can be applied as last resort option to detect exceptional anomaly samples.

2.4.3 HBOS

Histogram-based Outlier Score (HBOS) is a statistics-based algorithm for unsupervised anomaly detection. The algorithm is efficient complexity-wise and therefore is suitable for real-time detection [GU16]. The algorithm calculates a histogram of the feature distribution. Then it identifies in which bin the new point lies. After, the algorithm calculates the height of the bin, which should determine the likeliness of the event and therefore decide if it is an anomaly [GU16].

HBOS suggests two strategies for creating bins of the histogram. First is to use k bins of equal size. The second strategy is to have k bins with $\frac{n}{k}$ values which are grouped into a single bin where n is the number of total instances. The grouping is done based on the value. If the sorted array is used to store samples, first n following items would go to the first bin and so on. In case there are more than $\frac{n}{k}$ points with the same value, the algorithm allows for grouping into bins larger than $\frac{n}{k}$. In this work, we will be using the second approach as the density estimation is more robust in case of having large outlying values [GU16].

In the original paper [GD12], the score for an event p with d features is calculated as follows:

$$HBOS(p) = \sum_{i=1}^{d} log(\frac{1}{hist_i(p)})$$

The score is then compared to a threshold (as in the threshold-based method) and it is decided if the output is an anomaly or not. In this work, score for each feature is calculated individually and compared to a threshold τ. In other words:

$$HBOS(p) = \frac{1}{hist_i(p)}$$

Fig. 2.2 shows an example histogram and bins which would contain values which might be considered anomalous. It takes bins with the smallest hight, as they refer to the values with the smallest probability of happening.

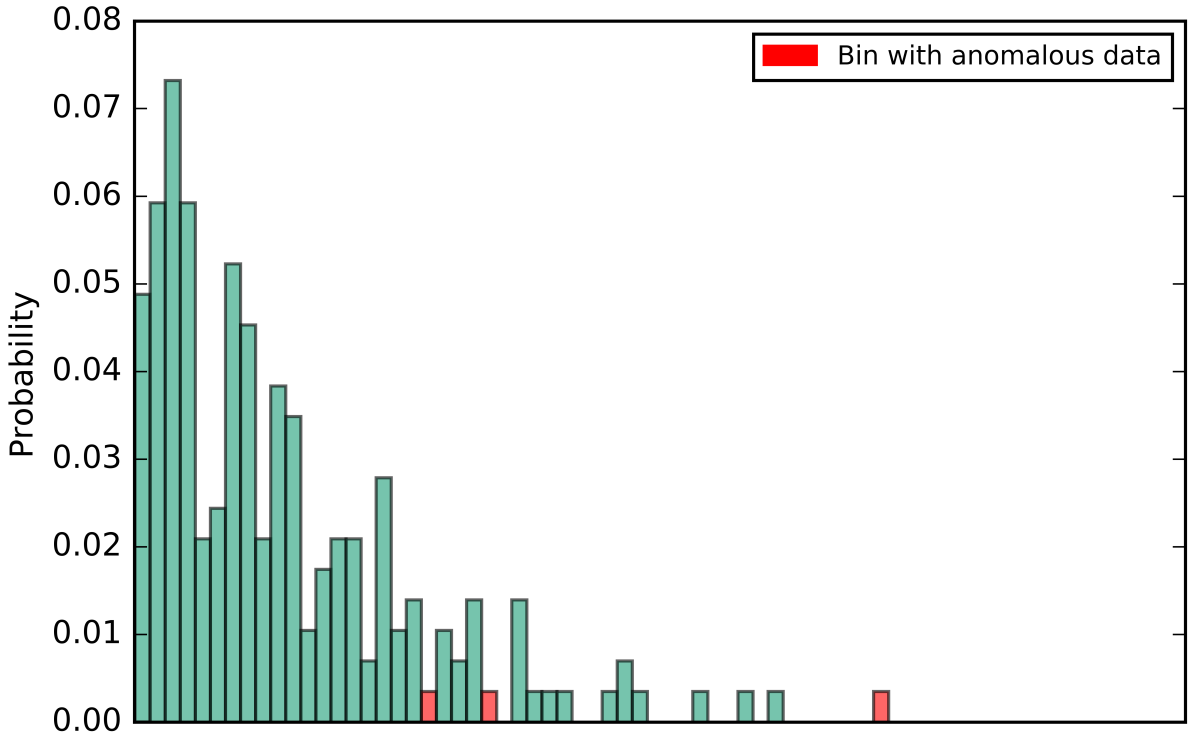

Figure 2.2: Example of histogram used for HBOS.

2.4.4 Mean-based

The mean-based algorithm is simple, fast and provides excellent heuristics for finding out volumetric anomalies [CKO17]. The idea is to check how much a particular point deviates from the mean value of previous observations. The interval is defined as a multiple ν of the standard deviation σ. In mathematical terms, value x is considered an anomaly if $\frac{x-\mu}{\sigma} < \nu$, where μ is mean of the feature's distribution [HTF01]. In our experiments, we take $\nu \in N$, where N is a set of natural numbers.

Fig. 2.3 illustrates the rationale behind the method. We can see that in case our data are normally distributed, the mean with some multiple of standard deviation covers a substantial amount of cases. In other words, every event happening with low enough probability is considered an anomaly.

Tab. 2.2 shows how different multiples of standard deviation affects the normally distributed data. It shows the amount of data it covers and percentage of outliers considered an anomaly. We can see that with the increased multiple, we significantly decrease the number of events to consider.

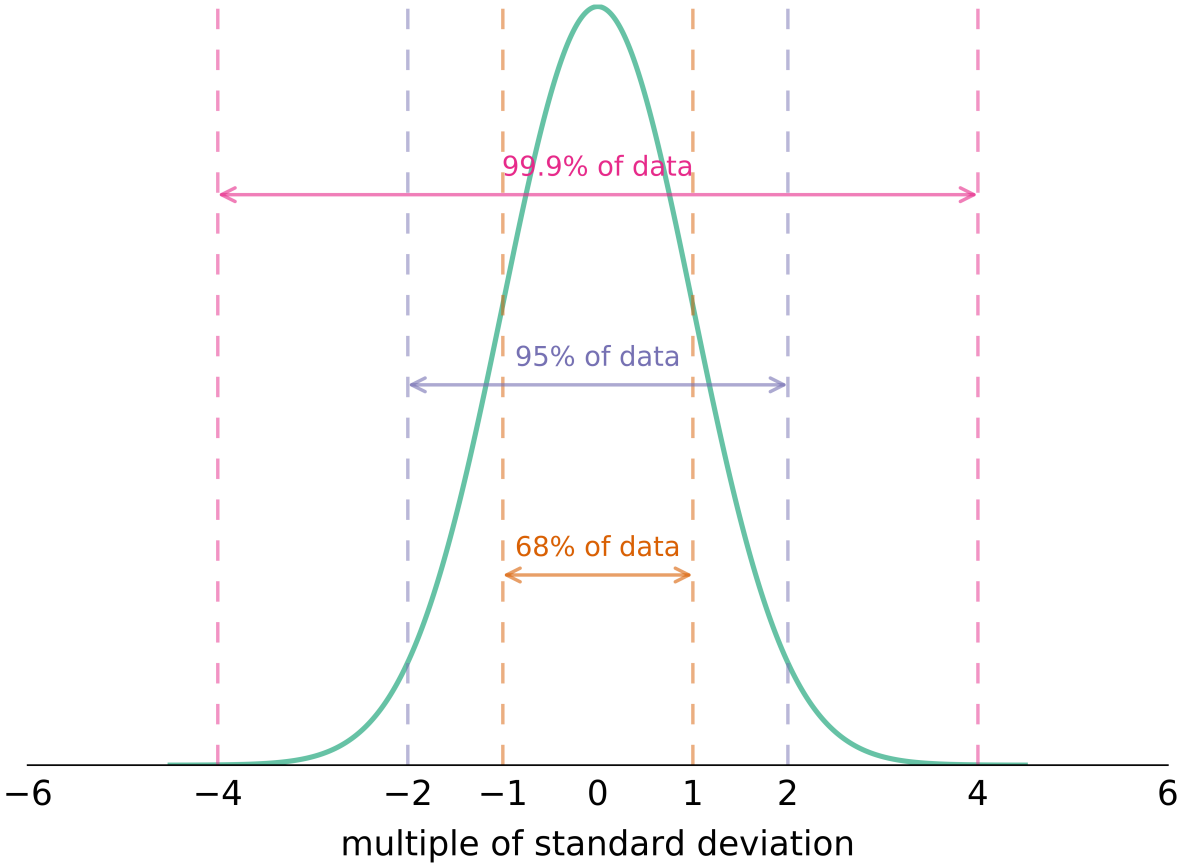

Figure 2.3: Normally distributed data and standard deviation.

	% of data covered	% considered anomaly
1σ	68	32
2σ	95.4	1.6
4σ	99.993	0.007
6σ	99.9999998	0.0000002
8σ	99.9999999997	0.0000000003

Table 2.2: Effects of different multiples of σ.

2.4.5 Moving Average

Models built over history which rely on previous behavior statistic are not aware of the current behavior trend. Moving average methods introduce an adaptive way for setting a threshold for an anomaly based on the behavior of the system just before thus taking current behavior into account. The idea behind those methods is similar to the mean-

based approach discussed above. We need to compute statistics over fixed, relatively small time window retrospectively and update the statistics with every incoming data. Based on statistics, we can predict the next value (average) and compare an actual new value against it.

In the trading domain moving average methods are often used for trading strategies [Zak15]. This method is used to predict the value of the stock to react somehow (buy, sell or hold).

2.4.5.1 Simple Moving Average

A simple moving average is calculated as an average value over the time window. Its disadvantage is the fact that everything contributes to the overall average equally, so the older data interferes with the newest ones, and therefore it is not able to catch a trend. Then, in case of an anomaly, it will also contribute to the next average computed. In mathematical terms, If M_t is a simple average for a moment t, the window contains n samples, and p is a sample, then:

$$M_t = \frac{1}{n} \sum_{i=t-n+1}^{t} p_i$$

2.4.5.2 Weighted Moving Average

The idea behind weighted moving average is the same as for simple moving average, but all the historical values are given a weight such that the older ones contribute less to the overall average. The weights are assigned linearly according to arithmetical progression:

$$M_t = \frac{np_t + (n-1)p_{t-1} + ... + p_{t-n+1}}{n + (n-1) + ... + 1}$$

2.4.5.3 Exponential Moving Average

Another modification of simple moving average is exponential moving average. This technique assigns weights in a way that weights decrease exponentially, so older values contribution decreases quickly and significantly. There are extensions of this method: double exponential smoothing, triple exponential smoothing and so on, which introduce additional factors for decreasing the weights over time. The difference from the weighted moving average is in the multiplier the value is assigned. It is calculated recursively as:

$$M_t = \alpha x_t + (1-\alpha)M_{t-1}$$

Where M_t is a moving average at the point t, smoothing factor α with restriction $0 < \alpha < 1$. Smoothing factor determines how quickly we forget the previous observations.

2.5 Tools

2.5.1 Python

The main language used in this work for model generations, computations, graph generations, and preprocessing is Python programming language. Python is a multi-paradigm general-purpose language which is popular in a scientific domain when the data has to be processed and then analyzed in some way.

2.5.2 Jupyter Notebook

The analysis was done in Jupyter Notebook which is a web-based editor, which enables to write readable and executable text (in case it is one of the supported languages, like Python). Notebooks integrate executable code and result in the same place, which makes it a great tool for interactive data analysis and development.

2.5.3 NumPy

NumPy is a Python scientific computing package which was used for any math-related computations.

2.5.4 Pandas

Pandas is a Python library which provides data structures and data analysis tools. It was used as the main container for the data, which allows for easy data manipulations.

2.6 Previous Results and Related Work

Researchers in anomaly detection in trading data has been focused on detecting an anomaly of the behavior of market on a particular stock, for example, stock price manipulation and detecting an insider activity. [ACU17] contains a survey of the performance of standard algorithms applied to stock trading data. [GZ15] takes a look at the detection of stock manipulation, discusses different approaches to the problem and suggests a novel approach – Contextual Anomaly Detection (CAD). [LTT16] focuses on the application of neural networks on two kinds of anomalies in stock trading (pump-and-dump and spoof trading) in their paper. The importance of the efficiency of algorithms for real-time detection and their actual efficiency is discussed in [CKO17].

Great survey of existing anomaly detection techniques and their performance is provided by [CBK09] and [GU16].

Chapter 3

Data

In this chapter, the data and its preprocessing are described

3.1 Data Format

3.1.1 FIX

In 1992 a group of institutions and brokers had started the Financial Information eXchange (FIX) as an effort to organize and standardize electronic trading in the trading community. Currently, the information exchange between clients and their counterparties in a majority of cases is done electronically [Gre16]. As a result, FIX protocol was born. It is an open protocol maintained by member firms of the community, which defines the way parties exchange messages and the messages themselves. At this moment, the protocol is supported by all major exchanges and is de-facto standard language parties speak in electronic trading [Lee06].

FIX is an ASCII message-based protocol. Each message consists of multiple tag-value pairs. Each tag-value pair has the following structure: tag, followed by "=" sign, followed by the value. A tag is a string representing a tag identifier (integer) of some field from FIX protocol. The value is a string representation of an actual value in a format specified by the protocol. Each pair is separated from each other with a SOH character (0x01 in ASCII) [Ltd17].

So in the end, the communication between two parties throughout the day can be described as an array of string, FIX messages.

3.1.2 CSV

Comma-separated values (CSV) format is a text format encoding the data in a simple form understandable by every data analysis tool. It encodes data as a plain text file consisting of records, each having the same sequence of fields, each field separated from another by a delimiter (for example comma).

3.2 Preprocessing

Usually data from trading are produced in FIX format, which is not suitable for further processing via data analysis we had to develop a tool for converting the data available into a format suitable for data analysis. The format chosen is CSV file.

3.2.1 Preprocessing tool

The tool developed for being able to analyze the trading traffic is a Python application, which takes a file of logs of FIX messages, and produces a CSV file. The output is a time-series data consisting of values of features over some requested period.

The application developed allows for straightforward adaptation for arbitrary FIX logs format by implementing a specific interface. Modularity of the code makes it easy to add new features on demand.

Source code with the guide how to use and examples can be found on enclosed CD.

3.3 Features

The following features were generated and tested in further experiments.

1. Quantity Buy – Sum of the quantity of bought stocks
2. AvgPrice Buy – Average of $OrderQuantity * PriceForStock$ of buy orders
3. Quantity Sell – Sum of the quantity of sold stocks
4. AvgPrice Sell – Average of $OrderQuantity * PriceForStock$ of sell orders
5. Total – Volume of traded stocks
6. OpenQty Buy – Sum of stocks left unbought
7. OpenVal Buy – Value of stocks which are left to be bought
8. OpenQty Sell – Sum of stocks left unsold
9. OpenVal Sell – Value of stocks which are left to be sold
10. Messages – Amount of messages sent
11. Orders – Amount of orders sent
12. Filled – Amount of fills sent
13. PartFilled – Amount of partial fills sent
14. Opened – Amount of opened orders

15. Cancelled – Amount of canceled orders
16. Modified – Amount of modification orders
17. Rejected – Amount of rejected orders
18. DFD – Amount of Done For Day orders
19. Trades – Amount of trades
20. Execs – Amount of execution reports
21. Replaces – Amount of replaces
22. Cancel Rjs – Amount of rejects of cancels
23. Replace Rjs – Amount of rejects of replaces
24. TradeCncls – Amount of busts of trades
25. Avg Ack – Average of time periods between sending an order and receiving acknowledgment of receiving it
26. Avg Replace – Average of time periods between sending a request for replacement of order and receiving acknowledgment of receiving it
27. Avg Cancel – Average of time periods between sending a request for canceling the order and receiving acknowledgment of receiving it

Chapter 4

Experiments

In this chapter, the experiments and their results are described.

4.1 Experimental Setup

The data used in the experiment are real broker data for the whole month of trading (23 days) of 2017. They were collected from multiple clients of that broker in the form of binary files containing FIX logs of trading activity for the specific client. Log for each day, depending on the client, contained 150000-300000 FIX messages on average.

In the area of anomaly detection, it is often hard to say what is an anomaly without expert opinion. Industry experts and authors experience were used to help identify and label anomalies to provide feedback on performance in this research. There were 52 anomalies labeled over different features out of nearly 10000 samples over 5 days. Anomalies were spread among all of the features and the evaluation measures (FPR and TPR) were calculated over all the anomalies altogether.

The data provided were converted into CSV file of 5 minutes snapshots with features described in 4.4. This newly generated data were used for experiments and analysis.

The experiments were split in two groups. First group deals with the reduction of false-positive rate using threshold-based, mean-based and HBOS-based models. Mean-based and HBOS-based methods rely on the historically learned data over a substantial period. The second part of the experiments aims to reduce the response time. In the second part moving average methods are used. They take into account the current trend of the data and thus provide an excellent framework to detect an anomaly as quickly as possible.

Different algorithms for different aims are used for a reason. Retrospectively learned models are good for estimating if data behaves differently historically, but they do not take into account current trend of the data behavior. Methods which take the trend in the account can detect local anomalies quickly. They can detect a sharp spike quicker, as they rely on previously short-window behavior and can tell if the data suddenly start raising or falling. Historically learned algorithm would detect that spike only when it reaches some

threshold missing the beginning of an anomaly. On the other hand, learned detectors are able to remember the general behavior patters of the data and react if newly streamed data behave differently relative to how it usually does. Trend-learning methods can not handle these kinds of anomalies.

4.2 False-Positive Rate

Normal and low-variance features were used for the analysis. For each client models were built individually, since clients had differences in their features classification (see 4.4). Models were tested against days with labeled anomalies. False positive rates and true positive rates were calculated for different models which were learned on 1, 5, 7, 10, 15 and 19 days chosen as directly preceding test days. Different time-windows (granularity of split into submodels) were tested where applicable. Most notable results of conducted experiments can be found in Tab. 4.4.

In all the results provided, threshold for HBOS algorithms was set to be 100. Experiments were conducted on HBOS-based 1 hour window model trained on 19 days with different thresholds. As a result, setting threshold to 100 combined both good results for FPR while keeping TPR high enough. Tab. 4.1 illustrates the results of the experiments.

The multiple of standard deviations was set to 4. Experiments on mean-based 1 hour model learned on 19 days showed that 4 standard deviations is optimal in terms of FPR and TNR. Tab. 4.2 illustrates the results of the experiments.

For threshold-based models the threshold was set individually for each feature.

Threshold	FPR	TPR
2	0.191646	0.758065
5	0.108635	0.758065
10	0.079502	0.758065
100	0.051948	0.758065
500	0.043521	0.618065
1000	0.041239	0.609677

Table 4.1: Different threshold for HBOS-based 1 hour window model trained on 19 days

Threshold	FPR	TPR
2	0.041418	0.822581
4	0.036680	0.822581
6	0.034398	0.690323

Table 4.2: Different threshold for mean-based 1 hour window model trained on 19 days

Model	FPR	TPR
Whole day (24h)	0.38053	0.799054
Whole trading day	0.011053	0.774194

Table 4.3: Comparison of FPR and TPR of 24 hours and trading day learning periods for mean-based model.

4.2.1 Time window

Testing the new value for anomaly could be done on the time-window basis. It means that the model is built in a way that it consists of multiple sub-models for time windows that split the day. Therefore new value is tested against some particular sub-model.

The experiments consisted of applying different time windows on HBOS-based and mean-based algorithms, which were trained on 19 days. The following time window splits were examined: per-hour, per-2-hours, per-4-hours, whole trading day.

Mean-based detector was trained on the whole (24 hours) day and on the trading times only. Learning on the data from the whole day altogether using single day-long (24 hours) time window does not perform well. Tab. 4.3 shows the comparison of these time windows based on mean-based 1 hour window models trained on 19 days. Taking an entire day as a time window introduces an error due to periodic nature of trading. Trading happens during stock exchange open hours, therefore during more the half of the day there is no traffic. Detector trained on the whole day takes those samples into account and artificially lowers the expectations for the mean, therefore allowing for some extra false positives. Since using time window of the whole 24 hours day performed poorly as it can be seen in Tab. 4.3, it was not used in further investigations and "whole day" further on refers to the trading day (roughly from 8 to 19 hours).

Per-hour models work well and are logical for the data as it should be able to fit subtle changes in activity (for example if there is a decrease in volume due to the lunchtime). Fig. 4.2 and Fig. 4.3 illustrate their performance in terms of FPR and TPR. However, there are a couple of places, which produce apparent false-positives. Those places are start and end of the trading. Those periods take approximately 1 hour for the tested clients. They do not start at some specific hour – there is around 30 minutes deviation between the points of start of the trading among different clients (the same with the end). Those rises and falls of volumes produce false positives because the time trading starts and ends usually cross the border of two time periods in per-hour model. Fig. 4.1 illustrates an example of the issue, it's visible that a number of subsequent anomalies detected are there and it is due to them belonging to the previous time window (2-hours window, HBOS-based model).

With the increase of time-window the false positive rate and true positive rate slightly decrease for both HBOS-based and mean-based models. Fig. 4.2 and Fig. 4.3 demonstrate how FPR and TPR change relative to the window size.

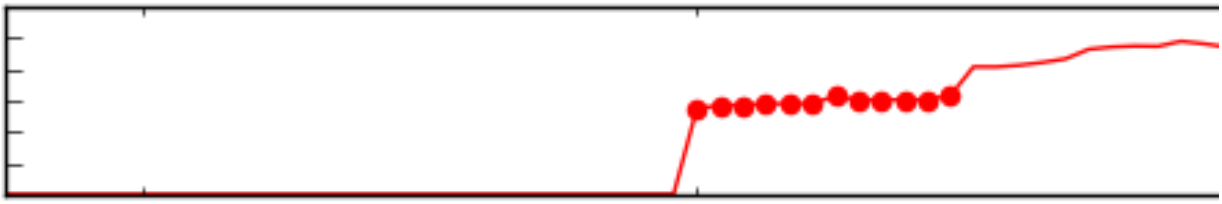

Figure 4.1: Start of the trading day. Incorrectly detected anomalies due to time-window overlap (2-hours window, HBOS-based model).

4.2.2 Learning period

Training mean-based and HBOS-based models is collecting mean for mean-based technique and histogram for HBOS. Different learning periods determine how many days were used to estimate those statistics.

Experiments consisted of testing multiple HBOS-based and mean-based models with different time windows. They were trained on 1, 5, 7, 10, 15 and 19 days.

It was discovered that learning period of around 15-19 days is optimal. It is long enough to avoid over-fitting and also not too long to carry the legacy of the old trading activities. Too many days to learn on can affect the analysis of the samples in case the behavior has changed. For instance, in case of the trading companies, it can be a change in the trading strategy, increase of wealth and therefore increase in the volume of the portfolio. Small learning period of less then 5 days suffer from high error since the model does not have enough data accumulated to classify a new sample.

Fig. 4.4 shows how false positive rate declines with the increase of learning period for all of the models. Fig. 4.5 illustrates how TPR rises at the same time. With the increased number of days TPR increases and FPR rate decreases. TPR for models trained on 19 days grows high comparing to models trained on smaller amount of days. FPR changes are generally not that drastic.

Overall, increase in learning period correlated with lower FPR and higher TPR.

4.2.3 Overall results

Throughout the experiments, it was evident that all time-window based models (such as HBOS-based or mean-based) share one common problem. They fail to fit the specifics of periodicity of trading – rise and the fall in volumes. It is not possible to adjust time windows so they don't overfit on those periods of the day. To solve the issue we propose a model, which splits the day into multiple parts and applies different algorithms to each. Proposed model:

1. Pre-trade and post-trade – a period before the start of the trading day and after the end of the trading day. There expected to be no activity during these periods,

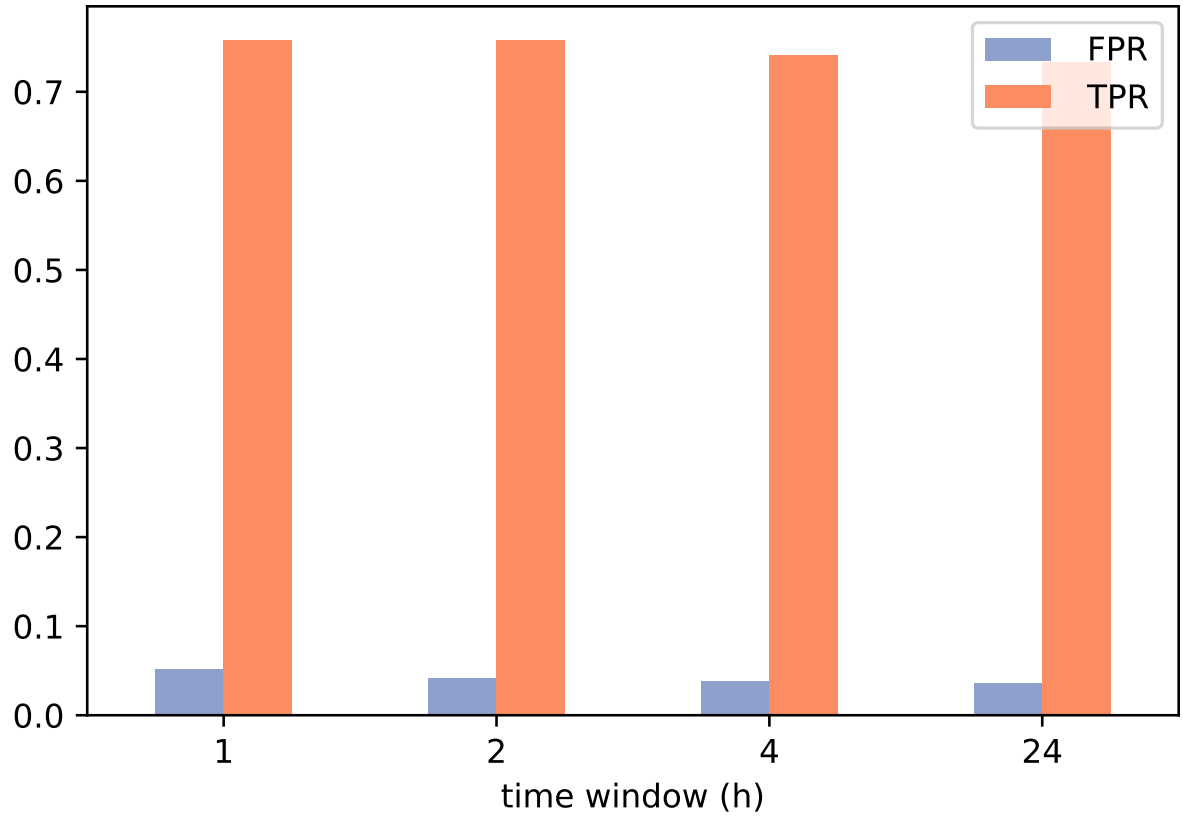

Figure 4.2: False positive rate and true positive rate relative to time window for HBOS-based models learned on 19 days.

therefore threshold-based detectors can be used to identify an inappropriate level of activity. The threshold has to be set per feature.

2. Start of the trade – a period when the trading starts, therefore normal features experience a rise in the volume. Due to not knowing when exactly the trading starts or ends, and in order not to mess with the other data, the periods have to be large enough (one hour). And from the nature of the data, during this hour, there will be either decline or rise of the volume. It is not possible to process robustly with the methods chosen, so for these periods, so we suggest either use a threshold-based method to detect reaching some critical value.

3. Day – time of the general trading activity. Per-hour HBOS model or mean-based model can be used for this type of activity.

4. End of the trading day – the same as the start of the trading day, but when the trading day ends, and the values decline. The only difference is that during this

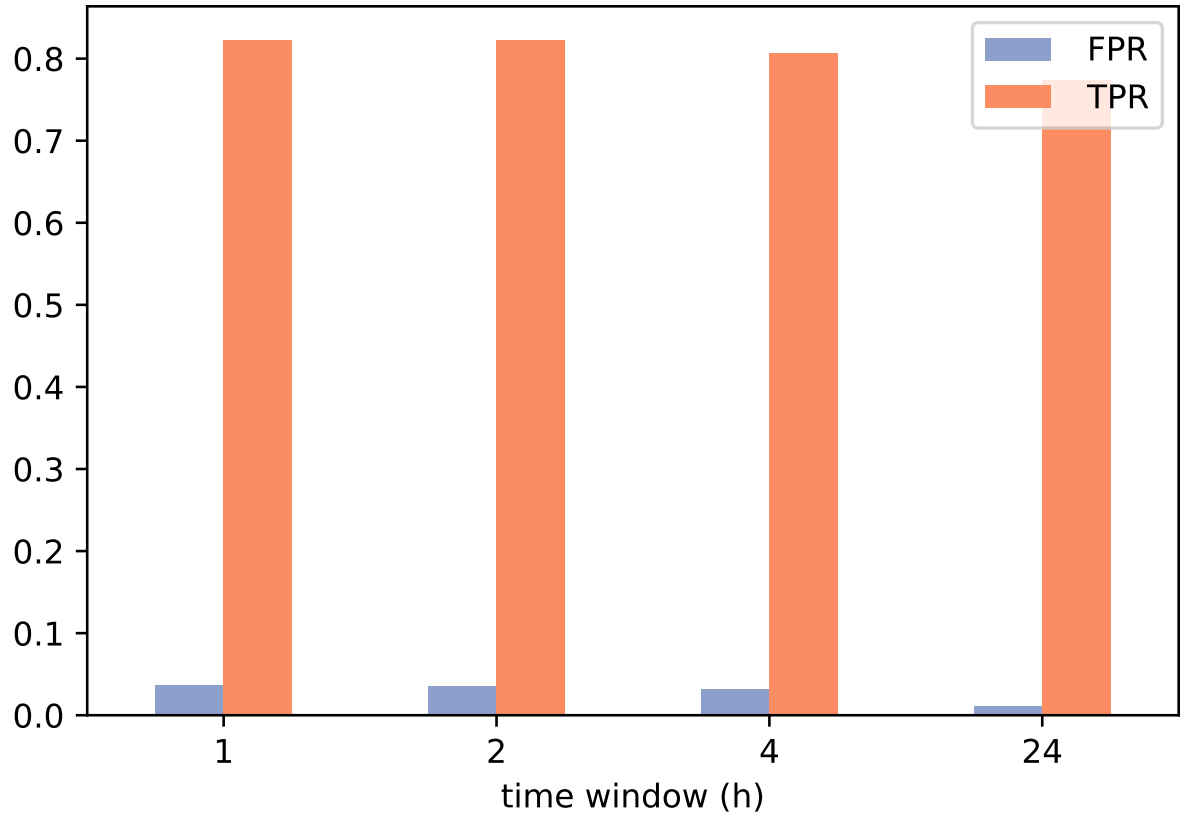

Figure 4.3: False positive rate and true positive rate relative to time window for mean-based models learned on 19 days.

period, some features experience a spike, which indicates closing the positions. For this period we use threshold-based method.

Tab. 4.4 illustrates the FPR and TPR of different models. Table represents most notable experiments and results, other variations (different learning period/time windows) showed similar results as the ones presented.

Two variations of proposed model learned on 19 days were tested. Model which uses HBOS-based 1 hour window model and mean-based 1 hour model for the trading day period of the split defined above. Proposed model, which uses mean-based predictor performed the best among others. Fig. 4.4 and Fig. 4.5 feature this proposed models and illustrates that mean-based proposed model is the best performing in terms of FPR and TPR. The other proposed model, which uses HBOS-based did not perform that well.

With learning period of less then 19 days, mean-based models performed better in terms of FPR, while HBOS-based ones showed better results in TPR. On 19 days learning period, mean-based models perform better then HBOS-based models in terms of TPR,

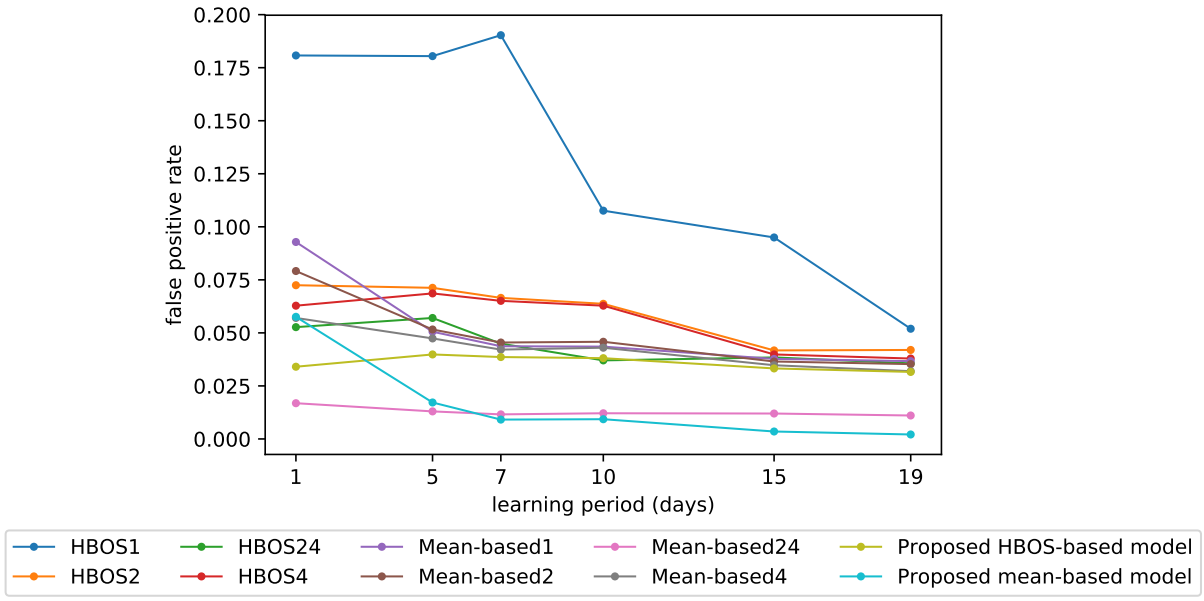

Figure 4.4: False positive rate relative to learning period.

FPR is similar.

4.2.4 Advantages of the Methods

Mean-based and HBOS-based methods are memory efficient as they require storing a relatively small set of numbers, which is, negligible in terms of memory. The complexity of methods is low, and therefore they are suitable for real-time detection, which is one of the requirements for the anomaly detection for the trading platforms. At the same time, using these methods, it's possible to build a model to achieve high true positive and low false positive rates.

Models created for the research show good performance in catching apparent and exceptional anomalies and can be either used as-is for simple anomaly detection or as a baseline for more advanced algorithms.

4.2.5 Disadvantages of the Methods

Mean-based and HBOS-based models on their own suffer from not being able to accustom rise and fall of volumes and therefore producing too many false positives during these periods.

Since the methods are statistics based, models are not aware of any relations between features (some features are interdepend). Tested models were able to detect only simple

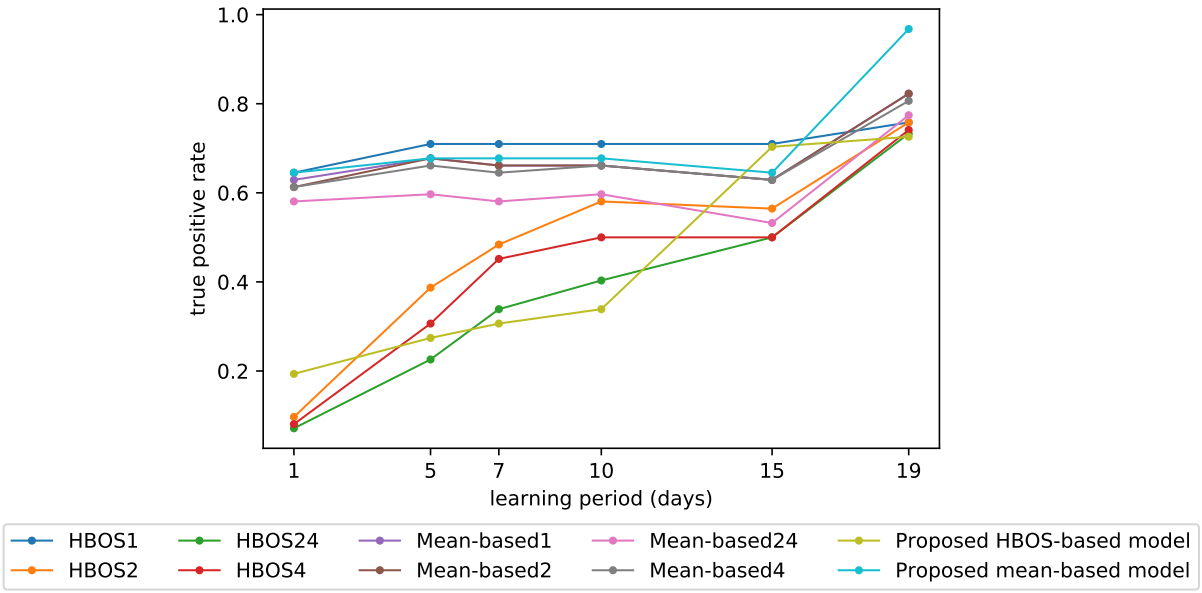

Figure 4.5: True positive rate relative to learning period.

anomalies indicating abnormal volumes for that particular feature. Hence, there are some anomalies which tested models can not be trained to recognize.

Regarding the proposed model, first, the need for hand-tailored time windows for each client due to different time zones and different behavior. Different clients have to have a different split of time windows. Even though some particular separation did well for one client, it has significant chances to perform poorly for some other with, for instance, different time zone. It introduces the scaling problem and human factor into the model, to correctly split the day into time windows one needs to know well the particular client behavior, and that can be error-prone and time-consuming.

Secondly, all the clients tested had distinct start and end of the day. In real life, it is possible for some clients to trade worldwide on different stock exchanges with varying times of opening. For them the picture would be different, they will not have distinct start and end of the trading days, there will continuously be some activity, with possible spikes on the opening of big stock exchanges worldwide. For this kinds of clients, proposed model wouldn't fit.

4.3 Response Time Reduction

For experiments in response time reduction specific client was chosen and features employed were *Messages* and *PartFilled* (see 4.4).

The period for creating a snapshot is what determines the response time. 1 minute and 30 seconds snapshots were generated via the developed tool (see 3.2.1) to test if we can

Models/Learning period	Measure	1	5	7	10	15	19
Proposed mean-based model	FPR	0.057564	0.01719	**0.009126**	**0.009302**	**0.003510**	**0.002106**
	TPR	**0.645161**	0.677419	0.677419	0.677419	0.645161	**0.967742**
Proposed HBOS-based model	FPR	0.03404	0.039832	0.038607	0.038080	0.033239	0.03159
	TPR	0.193548	0.274194	0.306452	0.338710	0.703226	0.725806
HBOS, 1 hour window	FPR	0.180765	0.180451	0.190323	0.107606	0.094977	0.051948
	TPR	**0.645161**	**0.709677**	**0.709677**	**0.709677**	**0.709677**	0.758065
HBOS, 2 hours window	FPR	0.072464	0.071236	0.066501	0.063696	0.041733	0.041948
	TPR	0.096774	0.387097	0.483871	0.580645	0.564516	0.758065
HBOS, 4 hours window	FPR	0.062812	0.068603	0.065097	0.062819	0.039835	0.037908
	TPR	0.080645	0.306452	0.451613	0.500000	0.500000	0.741087
HBOS, whole trading day	FPR	0.052706	0.057024	0.044921	0.037027	0.038428	0.035627
	TPR	0.071293	0.225806	0.338710	0.403226	0.500000	0.733021
Mean-based, 1 hour window	FPR	0.09284	0.050544	0.043700	0.043524	0.037733	0.036680
	TPR	0.629032	0.677419	0.661290	0.661290	0.629032	0.822581
Mean-based, 2 hours window	FPR	0.079151	0.051597	0.045455	0.045806	0.036504	0.035276
	TPR	0.612903	0.677419	0.661290	0.661290	0.629032	0.822581
Mean-based, 4 hours window	FPR	0.057038	0.047385	0.042120	0.042998	0.034749	0.031941
	TPR	0.612903	0.661290	0.645161	0.661290	0.629032	0.806452
Mean-based, whole trading day	FPR	**0.016848**	**0.012987**	0.011583	0.012110	0.011984	0.011053
	TPR	0.580645	0.596774	0.580645	0.596774	0.532258	0.774194

Table 4.4: False positive rate and true positive rate for different models. Bold highlights the best score in column.

reduce response time down.

Different moving average methods were applied for detecting anomalies. The number of samples per window was set to 10 as minimal required sample size according to [CKO17] and [HV07]. So in case of 1 minute snapshot, the learning period was 10 minutes, and for 30 seconds snapshots, it was 5 minutes prior which were taken into account to build the statistics.

Experiments consisted of running different moving average algorithms on 2 days with label anomalies and on 2 days without any anomalous data. 30 seconds or 1 minute snapshot data were generated for these days. For the client chosen for testing the feature "Messages" was a normal feature and "Part Filled" was low-variance feature. Analogous to false positive rate reduction, 4 multiples of standard deviation was chosen as a threshold.

All the algorithms for moving average detected all of the anomalies. At the same time they seem detect all the spikes as well. As a result, TPR is high (1), FPR is low, but FDR is very high. Tab. 4.5 presents the result of false positive and true positive and false discovery rates experiments. All of the moving average algorithms performed almost identically, so only the exponential moving average algorithm is presented with $\alpha = 0.4$. Different α values for exponential moving average were tested, but performed identical. Fig. 4.6 and Fig. 4.7 demonstrate how moving average algorithms performed on the day without anomalies, 30 seconds window the former and 1 minute window the latter.

Overall, these algorithms detect all the spikes and anomalies, consequently, are detected as well. So the absence of a signal about an anomaly is an indicator that there is no anomaly

	TPR	FPR	FDR
1 minute snapshot	1	0.022619	0.904761
30 seconds snapshot	1	0.020830	0.945945

Table 4.5: True positive rate and false positive rate for exponential moving average with $\alpha = 0.4$ for 30 seconds and 1 minute snapshots.

whatsoever. At the same time the presence of a signal does not mean that there has been an anomaly, since false discovery rates is very high. Since the results for both tested snapshots are close to each other, preferred snapshot is 30 seconds.

4.3.1 Advantages of the Methods

Methods used are efficient for near-real-time detection, since updating the current window statistics is fast and it has a small memory footprint because it is only needed to store n items at the given moment for n-large window. n tested equals to 10, so resulting memory is negligible.

It is feasible to use moving average methods for quick detection of the anomalies, the spikes are correctly identified, so the methods can be used for reducing response times to 30 seconds or 1 minute. Using this methods we keep high confidence of not missing an anomaly.

4.3.2 Disadvantages of the Methods

Moving average methods are detecting anomalies relative to the current behavior of the system and spikes which are identified, even though correct, have a high chance of not being an actual anomaly. Tests showed high false discovery rate, which indicates that most of the signals were not actual anomalies. On features with high oscillation, there are a lot of spikes detected, while in reality this feature is expected to oscillate and have frequent ups and downs.

4.4 Features Selection

Throughout the experiments, it was found out that the behavior of features does not follow the same pattern, and, what is more important for anomaly detection using a statistical approach, different techniques seem to be applicable for different features. Features can be split into the following categories (feature can be in multiple categories due to different clients trading strategies):

1. Low-variance
2. Normal

3. Not suitable for statistical anomaly detection

4.4.1 Low-variance

Low-variance features are features which statistically have low variance and are always around 0. For this kind of features using mean-based or HBOS approach is not feasible, a simple threshold-based filter would suffice, as usually any big spike in the value would be considered an anomaly and something for the monitoring person analyze. Hence, it was used in proposed models for a couple of periods.

At the same time, methods used for response time reduction can be used for these low-variance features, as they can immediately detect a sudden change in features behavior.

Fig. 4.8 illustrates an example of a low variance feature and how it behaves.

The following features have been classified to be low-variance features for some of the clients: Filled, PartFilled, Cancelled, Modified, Rejected, DFD, Trades, Execs, Replaces, Cancel Rjs, Replace Rjs, TradeCncls, Avg Ack, Avg Replace, Avg Cancel.

4.4.2 Normal

Normal features behave in the following manner. There is a rise in the start of the trading day, activity during the day, and decline to small values after the end of the trading day. For these features, statistical anomaly detection methods are applicable if we assume there is a pattern not only in its behavior metrics but also in volumes from day to day, which is the case for these features. Mean-based and HBOS-based models were successfully used on these features.

Fig. 4.9 and Fig. 4.10 show how normal features generally behave throughout the day.

The following features has been classified to be normal for some clients: OpenQty Buy, OpenVal Buy, OpenQty Sell, OpenVal Sell, Messages, Orders, Opened, Filled, PartFilled, Cancelled, Modified, Trades, Execs, Replaces, Avg Ack, Avg Replace, Avg Cancel.

4.4.3 Not suitable

Some features are not suitable for the analysis. The nature of their values is the result of multiple signals traders are receiving. Statistical approaches fail on those features because their behavior is different from day to day and the spikes observed are not predictable from the data alone. However, a simple threshold-based filter can work as a risk check in case the values are exceptional. The same can be said about using response rate reduction techniques; those can detect huge spikes successfully while leaving small spikes undetected.

Fig. 4.11 and Fig. 4.12 depict the example of the features not suitable for the anomaly detection analysis. The figures are presented with an example of anomalies detected via HBOS-based 1 hour model learned on 19 days to show how generally algorithms perform poorly. While some spikes do look like an anomaly and broker might want to be notified about it, those spikes vary from day to day and there is no obvious pattern, therefore

threshold-based method can be used for them. Those features, however, were not part of the experiments.

From the generated features not suitable ones are: Quantity Buy, AvgPrice Buy, Quantity Sell, AvgPrice Sell, Total.

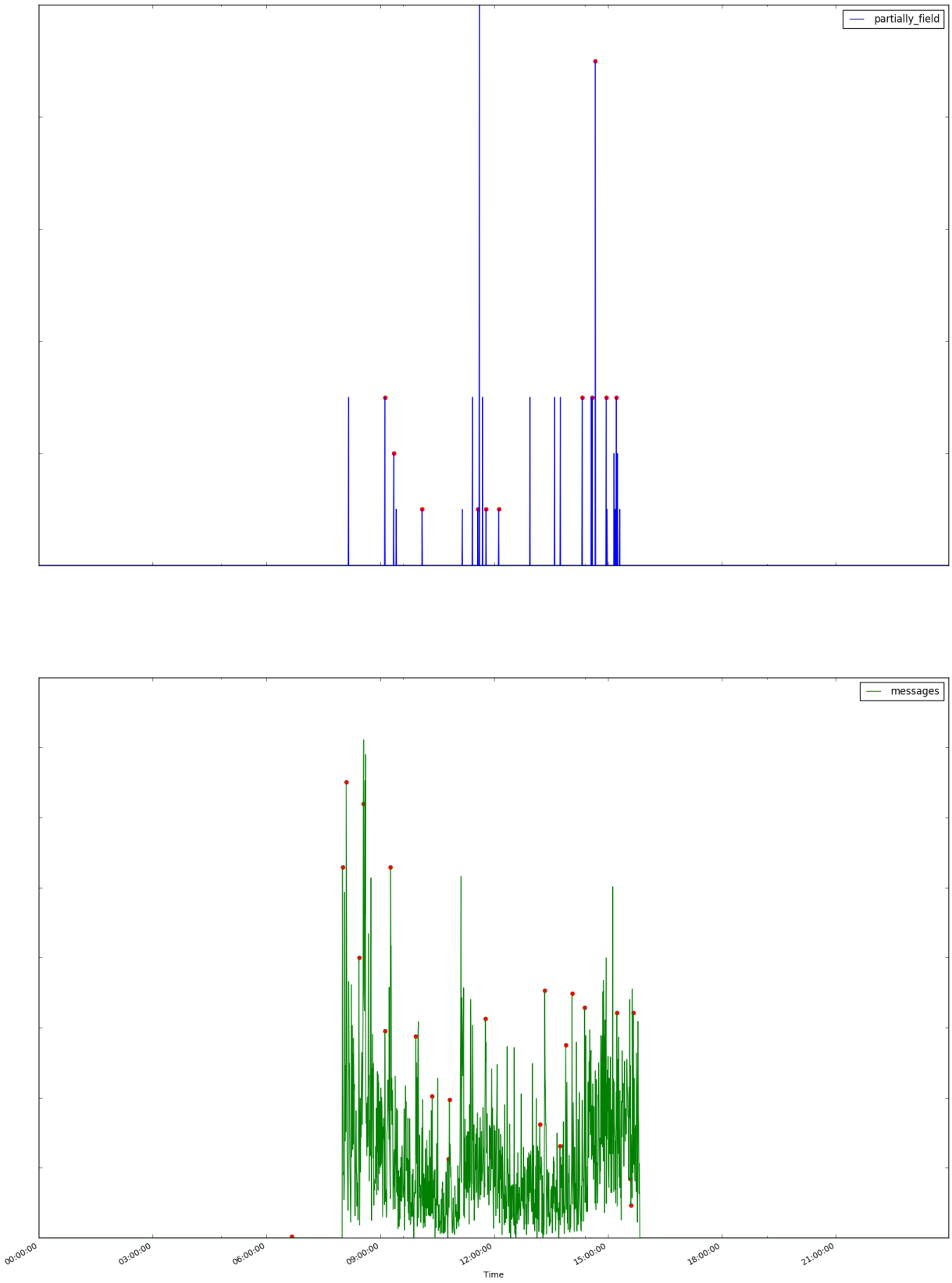

Figure 4.6: Exponential moving average with 30 seconds time window.

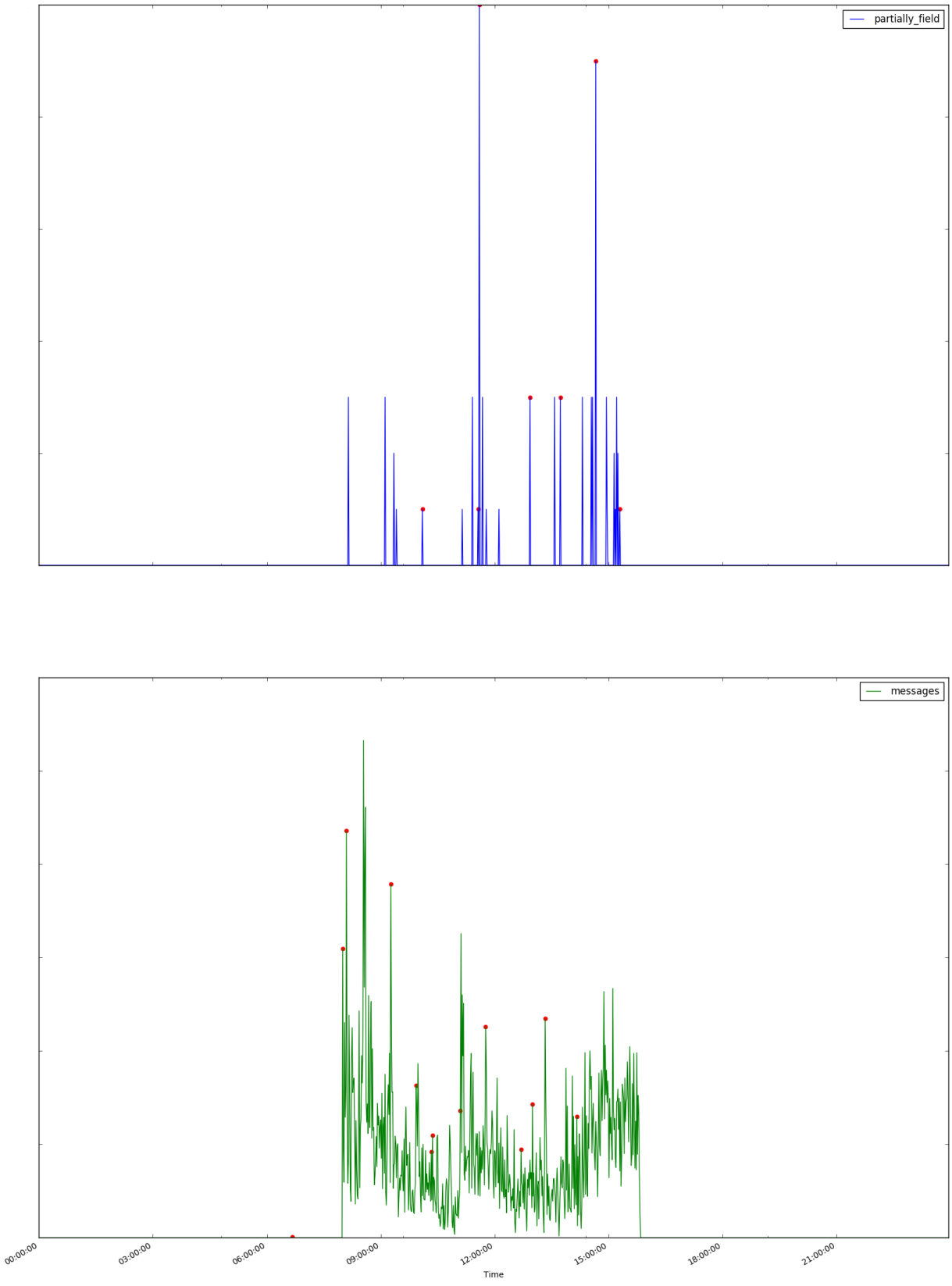

Figure 4.7: Exponential moving average with 1 minute time window.

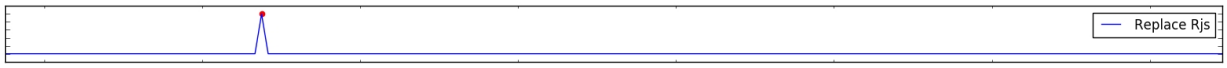

Figure 4.8: Example of a low variance feature.

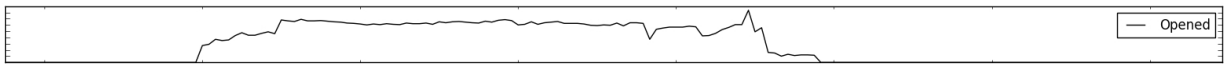

Figure 4.9: Example of a normal variance feature 1.

Figure 4.10: Example of a normal variance feature 2.

Figure 4.11: Example of a feature not suitable for anomaly detection 1.

Figure 4.12: Example of a feature not suitable for anomaly detection 2.

Chapter 5

Conclusions

In this chapter, the bachelor thesis is concluded, and suggestion for future research are provided.

The thesis has resulted in the first public researches in the domain of anomaly detection on the stock trading data which is not based on the data of some particular stock. It can be used by brokers as a framework or a baseline for building their anomalies monitoring solution.

Thus the main contributions are: the tool for converting trading data logs into format suitable for analysis, anomaly detection techniques applied on features which were generated using the developed tool, different models for anomaly detection were discussed and their performance was measured, response rate reduction methods were analyzed.

Features were analyzed and classified into the following classes: low variance, normal and not suitable for anomaly detection. Behavior of different algorithms on different feature classes was discussed. Low variance features work well with threshold-based method and moving average methods. Mean-based and HBOS-based models are good for normal features. Threshold-based method can be used for features not suitable for anomaly detection as a last resort filter.

Overall, it is possible to create well-performing model with a small false positive rate and high true positive rate. Using the proposed approach of splitting the day into logical parts and applying different algorithms on each part we were able to achieve false positive rate of 0.002106 and true positive rate of 0.967742, which is superior to other models tested (HBOS and mean-based). For mean-based and HBOS-based models increase of learning period corresponded with lower FPR and higher TPR. Increase of time window for these models resulted in slightly lower FPR and TPR. Best performing models were trained on 19 days.

Unfortunately, there is plenty of hard manual work. Mostly, in choosing time periods correctly and therefore models have to be designed for each client individually (unless they start/end the trading at the same time). There have to be some customizations based on empirical data. Ideally, the model also can be chosen for each feature individually.

Methods which were tested for response time reduction performed almost identically.

They yielded low false positive rate, high true positive rate, but at the same time high false discovery rate. That means that the signal doesn't indicate with high confidence that anomaly actually appeared, however if it does appear, it will get detected. In order to use methods tested for anomaly detection, there has to be some extra layer for post-processing, which will indicate if it's an actual anomaly.

5.1 Future Work and Improvements

Algorithms used for false-positives and response time reduction experiments are not mutually exclusive. In a complex system, they should be used together. They can detect different types of anomalies and can not be substituted one for another. The direction of building multi-layered anomaly detection system can be a topic for further investigations.

Investigation of different, perhaps, more elaborated anomaly detection techniques can yield good results in sophisticated models. Different advanced non-statistics-based unsupervised methods like neural networks can be researched.

It is worth to investigate ways to split different periods of the day automatically, based on general behavior pattern during this period. Therefore eliminating human factor in deciding the borders between different time-windows and improving the scaling and precision of the time-window based models.

Anomaly detection with labels can be researched to add another layer to the problem: classification of anomalies.

It is worth investigating how other near-real-time anomaly detection techniques perform on the trading data and if it is possible to incorporate data from previously learned behavior with the quickness of reaction for the local anomalies.

Methods which lower false discovery rate, while keeping true positive and false positive rates can be researched.

Bibliography

[HTF01] T. Hastie, R. Tibshirani and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2001. ISBN: 9780387952840. URL: https://books.google.cz/books?id=VRzITwgNV2UC.

[Lee06] Euromoney Institutional Investor PLC Lee Oliver. *Trading protocols: More Fix in FX*. 2006. URL: https://www.euromoney.com/article/b1321zg719lq20/trading-protocols-more-fix-in-fx (visited on 30/01/2018).

[HV07] Rob Hyndman and Andrey V. Kostenko. "Minimum Sample Size Requirements for Seasonal Forecasting Models". In: *Foresight: The International Journal of Applied Forecasting* 6 (Feb. 2007), pp. 12–15.

[CBK09] Varun Chandola, Arindam Banerjee and Vipin Kumar. "Anomaly Detection: A Survey". In: *ACM Comput. Surv.* 41.3 (July 2009), 15:1–15:58. ISSN: 0360-0300. DOI: 10.1145/1541880.1541882. URL: http://doi.acm.org/10.1145/1541880.1541882.

[GD12] Markus Goldstein and Andreas Dengel. "Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm". In: *Wölfl S, editor. KI-2012: Poster and Demo Track*. (2012), pp. 59–63.

[Eur14] Council of the European Union European Parliament. *Directive 2014/65/EU of the European Parliament and of the Council of 15 May 2014 on markets in financial instruments and amending Directive 2002/92/EC and Directive 2011/61/EU Text with EEA relevance*. 2014. URL: http://eur-lex.europa.eu/eli/dir/2014/65/oj (visited on 30/01/2018).

[Jam+14] Gareth James et al. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014. ISBN: 1461471370, 9781461471370.

[GZ15] Koosha Golmohammadi and Osmar R. Zaïane. "Time series contextual anomaly detection for detecting market manipulation in stock market". In: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (2015), pp. 1–10.

[Zak15] Valeriy Zakamulin. "Anatomy of Market Timing with Moving Averages". In: (Mar. 2015).

[GU16] Markus Goldstein and Seiichi Uchida. "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data". In: *PLOS ONE* 11.4 (Apr. 2016), pp. 1–31. DOI: 10.1371/journal.pone.0152173. URL: https://doi.org/10.1371/journal.pone.0152173.

[Gre16] The Financial Times Limited Gregory Meyer. *Trading: What happened when the pit stopped*. 2016. URL: https://www.ft.com/content/4d221b22-3dfb-11e6-8716-a4a71e8140b0 (visited on 30/01/2018).

[LTT16] Teema Leangarun, Poj Tangamchit and Suttipong Thajchayapong. "Stock Price Manipulation Detection Based on Mathematical Models". In: *International Journal of Trade, Economics and Finance* 7 (June 2016), pp. 81–88.

[ACU17] Mohiuddin Ahmed, Nazim Choudhury and Shahadat Uddin. "Anomaly Detection on Big Data in Financial Markets". In: ASONAM '17. Sydney, Australia: ACM, 2017, pp. 998–1001. ISBN: 978-1-4503-4993-2. DOI: 10.1145/3110025.3119402. URL: http://doi.acm.org/10.1145/3110025.3119402.

[CKO17] Dhruv Choudhary, Arun Kejariwal and Francois Orsini. "On the Runtime-Efficacy Trade-off of Anomaly Detection Techniques for Real-Time Streaming Data". In: *CoRR* abs/1710.04735 (2017).

[Ltd17] FIX Protocol Ltd. *FIX 4.2 Specification*. 2017. URL: https://www.fixtrading.org/standards/fix-4-2/ (visited on 30/01/2018).

[SA17] European Securities and Markets Authority. *MIFID II*. 2017. URL: https://www.esma.europa.eu/policy-rules/mifid-ii-and-mifir (visited on 30/01/2018).

Appendix A

Contents of enclosed CD

- `readme.txt` ..... the file with CD contents description
- `src` ..... the directory of source codes
  - `summary_view` ..... preprocessing tool sources
  - `thesis` ..... the directory of LaTeX source codes of the thesis
- `text` ..... the thesis text directory
  - `thesis.pdf` ..... the thesis text in PDF format