METHODOLOGY FOR MACHINE LEARNING-BASED MICRO-EVENT DETECTION IN NON-STATIONARY MULTIDIMENSIONAL TEMPORAL DATA

Artur Sokolovsky

Supervisors: Jaume Bacardit, PhD
Thomas Gross, PhD

School of Computing
Newcastle University

This dissertation is submitted for the degree of
Doctor of Philosophy

01.07.2022

I would like to dedicate this thesis to my wife Ana and daughter Adelaide who make my life brighter and give many reasons to smile and laugh.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 80,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 100 figures.

Artur Sokolovsky
01.07.2022

Abstract

The last two decades led to a significant increase in data availability. This is especially true for online communities. Namely, we have been seeing a significant activity and content amount increase within social networks, forum communities and electronic marketplaces. All these can be seen as platforms for online community interactions. Moreover, these platforms contain all the history of the interactions, creating a large footprint and allowing us to study more subtle structures within the data. While 20 years ago researchers only started thinking about topic detection and tracking in textual data sources (mostly limited by news feeds), now we have a huge variety of data feeds including Twitter, Stack Overflow, and Reddit, to name the most popular ones. Furthermore, these data sources contain hundreds of gigabytes of structured and free text data, that can be easily accessed and analysed.

For instance, over the last decade, many successful attempts were made to detect various event types in Twitter data in an automatic way. These included natural disasters, local concerts, celebrity-related events, collective events like pandemics, and so on. The more data we get access to, the more ambitious goals we can set. There are many activities in online communities that are not widely advertised. This is especially true for hacker forums, as well as forums dedicated to other illegal activities. Considering the growth of these communities, it becomes essential to perform automated analysis and risk assessment of such platforms and their trends.

The current work lays the basis for achieving this goal by introducing the notion of micro-event, as an event not detectable for a single data record. For instance, if there is a single tweet, one cannot judge whether it is related or not related to the micro-event. However, in the context of other tweets, it might be treated as related to the micro-event. This subtle nature of micro-events makes them incredibly hard to reliably detect in an automatic way. Hence, it is essential to design a generalisable methodology that would allow reliability, reproducibility and comparability when studying micro-events.

In the work, I propose a definition of micro-events, as well as a generalisable methodology for their discovery and classification. In the work, I discover that the definition, as well as the methodology, are suitable not only for textual communications but also generalisable to time series data. Since it is not feasible to get the labelled data for the above-mentioned

use cases, I design datasets and experiments to mimic the described settings as closely as possible.

Firstly, I apply the proposed methodology to detect FLOSS (Python packages) version release events in Stack Overflow data. The version releases are not explicitly mentioned or advertised in the data source, however, the events impact the community by introducing and deprecating packages' functionality. This makes the proposed experiment a good example of the micro-event detection task. Secondly, I adapt the proposed methodology to financial time series data, where market patterns align well with the introduced definition of micro-events. I introduce a machine learning-tailored market pattern, means for its automatic detection, and prediction of the price action scenarios after the event takes place.

Acknowledgements

I thank Jaume for teaching me the art of machine learning, fruitful discussions, guiding me through the whole journey and supporting me during the harder times.

I thank Thomas for introducing me to the world of statistics, as well as inspiring me to set more ambitious goals.

I thank Luca Arnaboldi for his support, fruitful discussions and an amazing mock-viva that helped a lot!

I am very grateful to my parents for their positive attitude to my studies and their support throughout. And, specifically, to my grandpa Michael, who encouraged me to sleep less and work more:)

Finally, I am very thankful to Newcastle University for providing a great environment for working on my program.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

"You know my method. It is founded upon the observation of trifles." — Arthur Conan Doyle, The Boscombe Valley Mystery - a Sherlock Holmes Short Story

1.1 Setting up the scene

The XXI century is the century when the amount of available data revolutionises society [1]. The so-called, "revolution" is driven among others by machine learning and AI [1]. The volumes and variety of data have been growing orders of magnitude over the last two decades due to more efficient crowdsourcing [2] and generally better community linkage [3]. We see that many industrial and business processes become data- and AI-driven with little-to-no involvement of humans [4, 5]. The fields where the outcomes are evident include advertisement [5], online communications [5] and financial markets [6]. The data in these fields has been intensively studied both in application-focused and fundamental science contexts [5, 7]. Considering the ever-growing complexity and volume of the data, new contexts emerge all the time. For instance, it becomes possible to identify properties in the data which were unrecognisable in the past [8].

The increasing connectivity of the society [3] is manifested by the rapidly-growing volume of interaction footprints. The examples of online community footprints are diverse and include social networks, forum-like textual communications, online review platforms, online stores, electronic marketplaces, etc [9]. There is no doubt that it is essential to understand how these communities function and develop methods for policing them [10]. One of the approaches to analysing and understanding online communities is topic detection and tracking (TDT) [11]. TDT is aimed at detecting events and topics in textual data sources, including news feeds and social networks. There are different kinds of events, including

events involving a tiny fraction of the community, collective events like epidemics, celebrity-related events, natural disaster events, and so on [12]. Their main limitation is that they are identifiable from a single unit of text, like a news article or a social network post.

To better understand this limitation, let us consider the following scenario: a purse is lost on bus route number A. The purse falls from the seat, gets opened and all its contents get spread through the bus. Passengers find this content and some of them post on Twitter about the discoveries. Before the purse falls down, it can be found using conventional TDT approaches, such as described by Sukel et al. [13]. However, the is a likelihood that the purse falls down and is not found by passengers but its contents are found. In such a case, conventional TDT approaches and event definitions would fail to discover the bus route. Addressing this limitation, in the current work, I introduce the definition of a so-called micro-event and propose a methodology for their detection and classification. The diagram illustrating the two scenarios is provided in Figure 1.1. In the current chapter, I use the notions of micro-event and event to distinguish ones not identifiable from a single data entry from the rest.

While the above-described example is simplistic and is of no interest from the application point of view, it helps communicate the notion of micro-event. A potentially impactful application of micro-events might include new and emerging threats detection in hacker forums, where the most advanced products are not publicly advertised but rather available to the trusted community members [14]. Naturally, the datasets of the most advanced malicious software that are made available to smaller hacker groups are not publicly available, hence designing an experiment for this specific application is not feasible. Hence, in the current work, I design datasets and experiments for micro-event detection in public financial and online communications data. In the past, this was not feasible either due to a lack of data. Nowadays, online community footprints provide massive volumes of structured free-text data, allowing to design experiments for micro-events detection.

In the current chapter, I first formally introduce events and micro-events, and put the latter in a wider context. Then, I discuss the challenges of micro-event detection and detail the scope of the thesis and contributions within the considered domains, as well as by setting the overall aim and objectives. Lastly, I conclude the chapter with the thesis structure and the list of studies published and submitted while working on it.

1.1.1 Events and micro-events

In the Cambridge International Dictionary of English “event” is defined as “anything that happens, especially something important or unusual” [15]. From the definition, one infers that the concept of the event can be applied to a wide range of contexts, depending on what

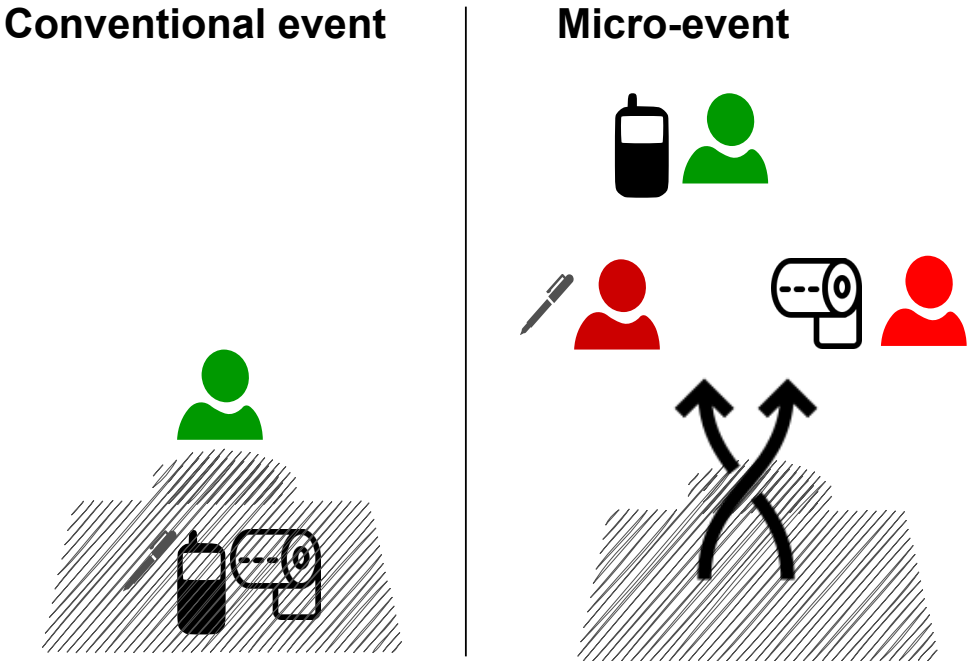


Fig. 1.1 Conventional events and micro-events scenarios of a purse lost on a bus route number A. Green and red icons indicate passengers who post and do not post on Twitter about an item discovery, correspondingly. When the purse is opened and its contents are discovered by different passengers, the detection of the event becomes not feasible by means of the existing topic detection and tracking approaches. The concept of micro-events introduced in the current work together with the methodology allows the discovery of micro-events.

"important" or "unusual" means for an individual. Hence, it is not surprising that the current body of knowledge contains a variety of event detection studies conducted at different scales and contexts [16].

I define micro-events as events that are not detectable from a single data entry. Moreover, a data entry cannot be associated with the micro-event on its own, but only in some context of other data entries. The data entry notion is broad here and might be a social network post, forum message, or even an electronic marketplace order.

While micro-events could be potentially treated as anomalies [17], their subtle nature would lead to a large false-positive rate when detected as anomalies, making the approach readily unsuitable for the scenarios considered in the current work.

Interestingly, the way events are defined in the TDT setting, they are limited to textual data footprints. However, the way micro-events are defined, this limitation is relaxed. In the current work, being guided by the proposed definition and methodology, I not only perform micro-event detection in forum-like textual communications but also demonstrate how micro-events can be detected and classified in the financial time series data with minimal methodological adjustments. This demonstrates the breadth of applications of the proposed definition of micro-events and the methodology of their detection.

As it was mentioned above, existing research on event detection focuses on the identification of events that can be associated with a single data entry [11]. The detection of weakly-manifested events, that can only be detected by looking collectively at multiple entries, has attracted far less to no attention in the events detection literature.[1] I see at least two reasons for that: i) conventional events are intuitively more impactful; and ii) research focused on conventional events has a lower risk of the null result.

1.1.2 Challenges

What are the factors making micro-event detection so challenging and risky for research? These include but are not limited by their weak manifestation, general data non-stationarity, as well as data complexity. These reasons are discussed below a bit more in detail.

Weak manifestation of micro-events

Even though micro-events are weaker-manifested than conventional events, there is no doubt that they might still have a huge impact on the whole community. For instance, financial markets are driven by large players but detection and understanding of their contributions is a big challenge [18]. There are at least two reasons for the weak manifestation,

[1] By the data entry I mean a piece of text, a message, a time series point, etc.

i) desire of the community member to stay anonymous or undetected, and ii) indirect relationship of the community member to the micro-event. An example of the first one is large players in financial markets who trade large volumes and want to minimise their impact on the market [18]. As the reader will see later, the second reason is relevant for the micro-event detection in forum-like communications performed in the current work.

Due to the weak manifestation, the performance of the machine learning models in the micro-event setting is relatively low. Hence, it is not clear, whether commonly accepted model explainability methods (like SHAP [19]) are reliable in this context. In the current work I follow one of the many potential ways of detecting micro-events. Other relevant approaches are discussed in the Background Chapter.

Data non-stationarity

When considering a developing online community, its characteristics change. For instance, the average activity of the members, the total number of posts per day, the breadth and depth of the discussed topics and the general nature of the interactions. Let us introduce the concept of this change more formally below. Stationary systems can be defined as having a constant unconditional joint probability distribution of its feature values over time, hence fixed mean and variance [20]. Non-stationary systems do not satisfy this condition. Definition of stationary and non-stationary systems varies across fields and authors by adding domain- and system-specific conditions, but the provided general definition is sufficient for the current work. The sources of non-stationarity in financial time series are constant qualitative and quantitative changes in market participants and all the external economic factors. In the case of forum-like textual communications, the non-stationarity is caused by internal community changes as well as external factors affecting the intensity and matter of the communications. As one can imagine, these changes require adjusting tools that are used for the community footprint analysis. For instance, one of the requirements is that the representation of the time step should be consistent and comparable across the analysed timeline.

Data multidimensionality

The number of variables used to describe the system defines the dimensionality of the analysed data. The more dimensions are present in the data, the higher the risk of finding spurious relationships between them [21]. On the other hand, the more data is available, the more likely it is to successfully analyse the system. On the lowest level, the financial time series is represented by a series of orders submitted, cancelled and executed by the market

participants. However, a common way of representing the system is through traded volumes, numbers of trades, and price changes over time [6]. More fine-grained classification of volumes and trades leads to even higher dimensionality of the data. Forum-like textual communications are represented by multiple communities and threads. The complexity of the footprint naturally dictates the higher dimensionality of the data representation. Considering the above-described non-stationarity, every dimension of the data representation should be made stationary and comparable across the considered time steps.

1.1.3 Objectives & aims

The overall aim of the work is to investigate the suitable means for micro-event detection and classification in the contexts of textual communications and financial time series, considering the non-stationarity and multidimensionality of the data sources. After better understanding the setting, propose a methodology for efficient micro-event detection & classification.

The specific objectives of the current thesis can be formulated as the following:

1. Define the micro-events in the context of online Q&A communities and financial markets;
2. Design a generalisable methodology allowing to work with weakly manifested micro-events in both domains;
3. Detect and classify the micro-events in forum-like communications data on the examples of Stack Overflow Q&A platform and financial time series;
4. Investigate the reliability of SHAP model explanations in the context of weakly manifested micro-events;
5. Detail the outcomes and limitations of the work.

In the later chapters, I state the more specific research questions that contribute to the stated overall aim and objectives. Below, I elaborate on the objectives with respect to the considered domains. Moreover, in Section 1.3, I relate every chapter to the stated objectives.

1.2 Thesis contributions

In the current section, I introduce the domains of interest more in detail, namely, forum-like textual data and financial time series. I list the contributions of the thesis to each of the

domains. Since the two settings have substantial differences, I identify research gaps and set aims separately for both.

1.2.1 Event detection in forum-like textual data

Event detection in textual data is a subfield of a larger research area of Topic Detection and Tracking (TDT). TDT has been an active research field for at least two decades [22]. It focuses on the identification and detection of events in text data, like news feeds and Twitter, as well as the investigation of how topics emerge in online communities and data sources. News articles or tweets in these datasets can be classified as related or not related to the event or topic. The common feature of these events is that they are followed by an observable response from the community, from which the event can be detected.

Until now, events that have been addressed in the Topic Detection and Tracking community were detectable from a single text entry [23–25]. Studying event detection in textual communications, I focus on software engineering (SE)-related forum data analysis due to the ample data availability, as well as the lack of event detection studies in the domain. Textual data in SE is present, among others, in the form of Question & Answer (Q&A) platform communications - Reddit and Stack Overflow (SO) to name two examples. These platforms have a large impact on the field because they are commonly queried for code snippets and solutions during the software development process.

I consider Free/Libre Open Source Software (FLOSS) version releases as micro-events in textual data and Stack Overflow Q&A communications as a platform where the associated communities interact and leave the footprint. I investigate whether FLOSS version releases can be associated with a characteristic change in the associated community interactions. In other words, I aim at detecting FLOSS version release events from SO message data.

Contributions

The contributions of the current work within the domain of TDT may be listed as follows:

- I introduce the concept of micro-event as well as the generalisable methodology for micro-event detection and classification, design a dataset specifically for micro-event detection and demonstrate how the methodology can be applied to it.
- I perform a feasibility analysis of the approach across a broad range of scenarios. Such elements as a set of considered features (predictors), different estimators, the type and length of the detection time window, and the type of events are investigated and discussed.

- Lastly, feeling the need for a better understanding of the micro-event concept, I introduce a synthetic forum-like data generation model, allowing the generation of synthetic datasets with controlled strength of the community response. It creates a synthetic data-driven basis for studying scenarios in which the micro-events can be detected.

1.2.2 Event detection in financial time series

Market events are the key means to evaluate changes in financial markets [6]. The study of these particular events allows traders to predict potential changes in market dynamics and permit the analysis of factors that may impact the profitability of trades. Being conditioned by the financial context, the notion of a conventional event here is narrower than in the TDT setting, however, is still very broad and includes natural disasters, economic news, political speeches, tweets, or even certain financial time-series patterns.

The micro-events in financial time series follow the original definition and are manifested as specific patterns of price and volume change detectable from multiple data entries.

Trading platforms offer the widest range of time series analysis tools, however since the analysis was done manually in the past, the existing methods are not necessarily optimal for the machine learning setting. Nevertheless, they are often adopted in automated trading systems off the shelf. This fact creates a knowledge gap in the field, as machine learning methods might have different requirements for the data representations and limitations in comparison to human intelligence. In the current work, I bridge the gap by introducing a new feature extraction approach and extending the volume profile market pattern to be suitable for the machine learning setting.

Contributions

The contributions of the current work to the field of financial time series analysis are the following:

- I describe all the components involved in the automated trading stack, including the challenges and complexity associated with each component.
- I introduce the concept of micro-event as well as the generalisable methodology for micro-event detection and classification, design a dataset specifically for micro-event detection and demonstrate how the methodology can be applied to it.
- I perform a statistically-backed feasibility study of the proposed approach for different time horizons, asset liquidity, method configurations, etc.

Due to the broad notion of events and, hence the varying objectives of the studies, both fields - TDT and financial time series suffer from poor research reproducibility and comparability. In the current work, I make the best effort to bridge this gap by introducing the methodology which allows for comparability and reproducibility of the results across studies.

1.3 Thesis structure

There are two types of chapters in the current thesis: the ones which lay a basis for the studies, and the technical ones. The rest of the thesis is structured as follows:

- **Chapter 2** introduces all essential information necessary to understand the work as well as links between the current thesis and the existing body of knowledge;
- **Chapter 3** communicates in detail the study of micro-event detection in the forum-like textual data [Objectives 1, 2, 3, 5];
- **Chapter 4** delivers the micro-event (pattern) extraction and classification method in the context of time series, guided by the methodology introduced in Chapter 2 [Objectives 1, 3, 5];
- **Chapter 5** builds on top of the Chapter 4 following the same methodology, expanding the idea of automatic micro-event detection to the newly introduced type of micro-event, comparing the two types of patterns from the classification performance perspective, assessing the optimal market conditions for the introduced micro-event type, and, last but not least, assessing the validity of using SHAP model interpretations in the context of micro-event detection and financial markets setting in particular [Objectives 3, 4, 5].
- In **Chapter 6** I discuss the results obtained from both domains with respect to the thesis objectives and overall aim and conclude the conducted work.

Chapters 3, 4 and 5 are technical and can be treated as separate studies and comprise of Aims, Material and Methods, Results, as well as individual Discussion and Conclusion sections.

1.4 Publications

The technical chapters of the thesis are based on the publications written by myself under the supervision of Jaume Bacardit and Thomas Gross or in collaboration with Luca Arnaboldi.

In the latter case, I present only work performed by myself. The papers used in the thesis are the following:

- **Chapter 3:** Sokolovsky A, Gross T, Bacardit J (2021) Is it feasible to detect FLOSS version release events from textual messages? A case study on Stack Overflow. PLoS ONE 16(2): e0246464. https://doi.org/10.1371/journal.pone.0246464.
- **Chapter 4:** Sokolovsky A, Arnaboldi L (2021) Machine Learning Classification of Price Extrema Based on Market Microstructure and Price Action Features. A Case Study of S&P500 E-mini Futures. ArXiv pre-print: 2009.09993.
- **Chapter 5:** Sokolovsky A, Arnaboldi L, Bacardit J, Gross T, (2021) Explainable Machine Learning-driven Strategy for Automated Trading Pattern Extraction. ArXiv pre-print: 2103.12419.

Chapter 2

Background

2.1 Introduction

In the current section, I provide all necessary information for understanding the methods, results, and findings of the current work. Since the research involves two distinct domains (TDT and financial time series), I first provide the background information common for both, then detail each of the domains separately. Namely, in Section 2.2, I introduce the concepts of supervised and unsupervised machine learning (ML), linear and non-linear ML estimators, classification and regression tasks, performance metrics, feature selection and hyperparameter tuning, and model analysis; in Section 2.4, I detail the statistical methods which allow addressing, certain limitations of ML - effect sizes, statistical tests and corrections for multiple comparisons. In the context of TDT (Section 2.5), I discuss state-of–of-the–the-art topic modelling approaches, synthetic text generation, and text difference metrics. For the financial time series (Section 2.6) I detail the way financial markets operate, then I introduce the concept of automated trading and its challenges, finally, I introduce ways of assessing automated trading system performance.

2.2 Machine learning

Machine learning term was defined in 1959 by Arthur Samuel as “the field of study that gives computers the ability to learn without being explicitly programmed”. [26]. The algorithms are provided with the data which they use to build or refine models and assess the quality of the models using specific performance metrics [27]. Machine learning can be used for distinguishing entries into classes or assigning them continuous variables [28].

Machine learning nowadays is used in most (if not all) data-driven fields, where one is expected to make decisions based on the data [29]. Concretely, it is used for data preprocessing, denoising, structuring, filtering, transforming, designing representations, etc. In the current section, one can characterise machine learning approaches as supervised and unsupervised, based on the target variable availability as well as introduce particular methods of both types used in the thesis.

Importantly, the other two types of machine learning approaches exist, that are not described in this chapter. Namely, reinforcement learning (RL) [30] and time series forecasting [31]. These approaches are not used in the current work hence not covered in the current chapter. In the dissertation, I do not explicitly perform time series forecasting but rather design labels and extract features from time series and timestamped text data for supervised classification.

2.2.1 Supervised machine learning

Supervised learning can be defined as a family of algorithms which learns to map an input to an output [32]. The learning is done on the training data, where the input-output pairs are defined and the algorithm aims at learning the existing mapping rules, which would hold for the unseen data. The latter is called *inference* data. When inference data constitutes a part of the original dataset, it is called *test* data. There are two common ways to set the supervised learning problems - regression and classification. The difference is in the form of output - regression models have a continuous target variable, and classification models use distinct classes as the target. In classification, the target is represented as a discrete variable.

Learning algorithms

Below I introduce the concept of estimator and linearity, then describe the algorithms used in the work. Estimators are data processing elements responsible for the modelling of the entries' target variables (and giving their estimate) based on input features (or observed data) [33]. The choice of the specific method differs depending on the size of the dataset, the number of features, and the expected relationships between them [34]. Moreover, one should also consider the complexity of the estimator itself, as it affects the overhead required to understand its output [35].

One of the many ways of classifying ML estimators is by their linearity [36]. Linear estimators use linear functions to separate the classes or fit the data relationships and usually require fewer resources, non-linear ones use non-linear solutions to find relations in the data and are more computationally demanding. Pronounced model non-linearity

often leads to finding spurious relationships in the data (also called high variance or over-fitting), hence poor generalisability to unseen data. Over-simplistic models might not be able to spot all present relationships in the data leading to the large bias of the fit (also called under-fitting) [37]. Linear estimators are much more transparent in terms of analysis and interpretation, while non-linear ones are often treated as a black box.

Hence, in the context of the current thesis, aiming to preserve transparency and good performance, I use both types of estimators. Namely: logistic regression and ensemble tree methods. The first one is an example of a linear estimator, the latter is a non-linear estimator.

Logistic regression One of the simplest estimators is built from the logistic regression. It is a model where the input features are fed into the model with adjustable coefficients, then the logistic function is applied to it to get the output. It is used for modelling binary variables, where the output is a float value in the range $(0,1)$, representing the probability of the output. If one applies some output threshold, the two classes can be distinguished based on the probability output [38].

The logistic function (called logistic or sigmoid curve) in its general form is defined as follows:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}, \tag{2.1}$$

where x_0 is the midpoint of the sigmoid, L is the maximum Y value of the sigmoid, k is the steepness of the curve. When applied in the estimator, more constrains are introduced. Namely, $x_0 = 0$, $k = 1$, $L = 1$, leading to the following function:

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{2.2}$$

Building a logistic model for n-dimensional $\mathbf{X}$, the following equation applies:

$$f(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n)}}, \tag{2.3}$$

where $f(\mathbf{X})$ is the probability of the positive class output. The model can be further extended to the multivariate version, where the number of outcomes is more than 2. In such a case, the probability of class Z output for K possible outcomes can be computed as the following:

$$Pr(Y_i = Z) = \frac{e^{\beta_Z \cdot \mathbf{X}_i}}{1 + \sum_{k=1}^{K-1} e_k^{\beta} \cdot \mathbf{X}_i} \tag{2.4}$$

Being very simple, this model is easy to interpret, additionally, there is a stack of well-established statistical methods for the model analysis, including the goodness of fit, outlier detection, influential cases, residuals analysis, etc. [39]. The variety of available analysis methods comes from the fragility of the model - there are many factors that might affect its fit quality. For instance, the linear dependency between the features (also called observed variables), different scales of feature values, and outliers. Importantly, these factors do not have that large impact in the case of more complex machine learning models. In the current work, logistic regression is used as a well-established powerful statistical tool in event detection in textual data.

Decision trees A decision tree can be considered as a predictive modelling approach, used in data science and machine learning in particular [40]. The training is performed by searching for the conditions to split the training entries into groups in a hierarchical way. Most decision tree building algorithms are greedy in nature but not all of them [41]. The hierarchies create decision paths, allowing to separate the entries into classes. The output of the model is obtained by following a number of conditional splits, depending on the values of the input features. The sequence of the splits leading to a leaf is called a decision path. The diagram of a simple decision tree is provided in Figure 2.1. The trees

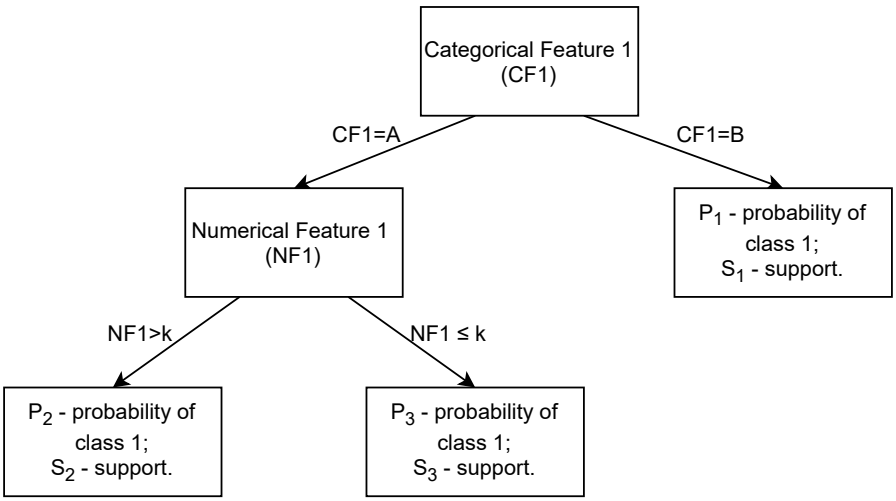


Fig. 2.1 An example of a decision tree for an arbitrary binary classification task. Two splits are demonstrated - for CF1 and NF1. Also, there are 3 leaves representing the tree decisions with the associated supports and class probabilities.

can be used for both classification and regression (called regression trees) tasks. Regression might require a more complex tree structure as the regression output often has more bins than classification. Generally, the number of output bins depends on the size of the training

dataset, the complexity of the problem, the feature space size, and the distribution of the outputs.

While simple decision trees are directly interpretable, in real-life scenarios, the number of splits might reach hundreds, depending on the feature space. Hence, at some point, the model becomes infeasible to manually assess and interpret [42]. Decision trees are an example of an unstable model, where a tiny change in the numerical feature value might lead to a completely different decision path. This instability makes it a great fit for ensembles of estimators, where multiple estimators are fitted on different versions of the dataset and their outputs are aggregated.

Ensembles By definition, ensemble estimators take outputs of multiple base estimators and aggregate them. The base estimators can be anything in theory, but decision trees are often used as an example of a relatively simple unstable estimator [43]. Another reason why tree-based approaches are hugely popular is that they are directly interpretable. In the ensemble setting several different trees are trained and used in unison to come up with the result. The simplest and most known case of this is Random Forest. Random Forest operates by constructing a multitude of decision trees simultaneously. The features and training entries are bootstrapped (sampled with replacement with uniform probability distribution) to diversify the information exposure of the single trees [44]. However, being guided by randomness, bootstrapping is not the most optimal way of fitting individual trees.

Improved robustness and performance are provided by boosting algorithms. In boosting, the base estimators are fitted sequentially with the following ones being more focused on the more complex entries. The final contributions of the base estimators are obtained either from their performance on the validation data or goodness of fit. One of the first boosting algorithms, as we know them, was AdaBoost [45], this work presented the concept of combining the output of the boosters into a weighted sum that represents the final output of the boosted classifier. Following on from this technique two other techniques were introduced - XGBoost [46] and LightGBM [47], both libraries have recently gained a lot of traction in the machine learning community for their efficacy, and are widely used nowadays. In this category, the most recent algorithm is CatBoost [48]. CatBoost is highly efficient and less prone to bias and over-fitting than its predecessors. CatBoost was specifically proposed to expand issues in the previous approaches which lead to target leakage (disagreement between the actual and predicted target distributions), which sometimes led to over-fitting. This was achieved by using ordered boosting, a new technique allowing independent training and avoiding leakage. Since boosting algorithms are robust, and efficient and often provide

close to state-of-the-art performance, I use CatBoost for classification problems in both domains. Additionally, in the TDT domain, I compare CatBoost to Random Forest.

Artificial neural networks Neural networks (NNs) are a biologically-inspired family of estimators. There are two general contexts in which the neural networks are studied: i. the biological and natural sciences approach, where the focus is to model the NNs of living systems and understand them better, and ii. making the best use of the models for problem-solving tasks [49]. In the current thesis, I only use the networks in the latter context.

An elementary unit of the NN is a neuron, which is represented by a mathematical function (called activation function). In the simplest case, the neurons are assembled in layers, where all outputs from layer k are fed into all the neurons in layer $k+1$ with varying weights (representing the strength of the connection). The outputs are multiplied by weights and summed, then the activation function is applied. The diagram representing the neural network model is provided in Figure 2.2 This architecture is called a fully connected feed-forward neural network. There is a wide variety of topologies that can be represented as networks with many of them known for decades. For example, Markov Chain - is a stochastic process represented by a set of states, where the probability of the following state is defined by the current state only, as well as its extension to Markov Chain Monte Carlo (MCMC) models that sample the original chain getting the probabilities distributions [50]. Another example is the Boltzmann machine - is a symmetrically connected network of neuron-like units that make stochastic decisions about its activity [51]. The learning is performed by updating weights between the units.

More recently discovered topologies involve connections between far-positioned layers, like ResNet [52] or even more complex connectivity as in Attention networks [53]. The further development of attention networks are transformers that differentiate the input data by significance for the model. Due to the parallel architecture of the neural networks and the simplicity of the base units (neurons), these models scale very well - they are commonly fitted using GPUs (Graphical Processing Units), which are capable of working with hundreds and thousands of threads in parallel. There are many NN architectures nowadays having up to trillions of neurons with the largest ones used for natural language processing tasks [53].

NNs are known to provide state-of-the-art performance in high-dimensional problems with complex relationships between features, like computer vision and natural language processing - to name two. The model training is done by adjustment of the weights through a backpropagation algorithm - the weights are changed proportionally to their contributions to the incorrect output. In the current dissertation, I use a pre-fitted transformer-based GPT2 neural network [54] for generating synthetic data.

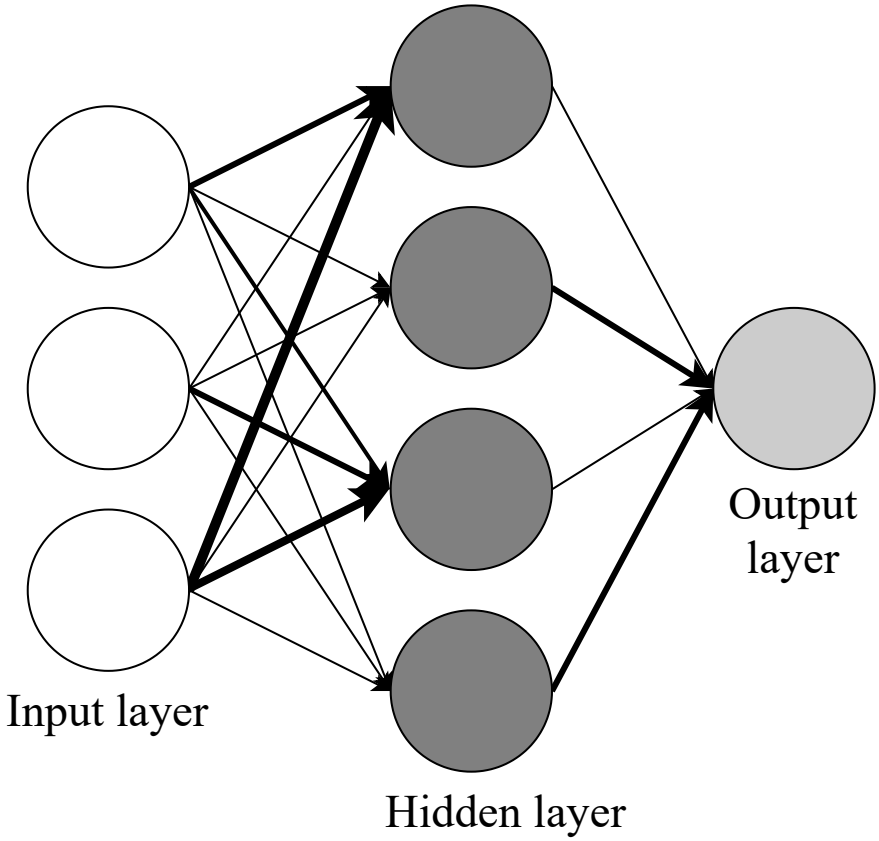


Fig. 2.2 A simple neural network architecture with a single hidden layer. The data flows from the left to the right. The thickness of the arrows represents different weights (can be seen as impact) of the source neuron to the target neuron. The input layer does not perform any transformation on the data. The transformations are introduced in the hidden and output layer.

2.2.2 Unsupervised machine learning

Unsupervised models focus on different types of tasks, e.g., finding aggregations of data points or frequent patterns in the data. They do not require labelled data to make use of it. Depending on the objective, the formalism may vary. Common examples are clustering of the unlabelled data or learning to reproduce the structure of the input data. Consequently, there might be different outcomes of such models - clustered data, which domain experts analyse and make use of, or models which learned the structure of the data and are capable of performing new tasks, for example, NLP domain, text translation, question answering, text generation, etc. Another example is topic modelling, wherein in the training phase, a model identifies topics in the text and assigns topics to the previously unseen texts in the inference phase.

As one can see, supervised and unsupervised approaches have very distinct applications and work in different settings. Both of them may be of use for event detection tasks. For instance, supervised learning is good for classifying entries into relevant or irrelevant to the event, while unsupervised learning may be used for synthetic data generation with the purpose of better understanding of the system, or extracting topic modelling features from the text. Below I overview of approaches that are used in the thesis.

Learning algorithms

All the data representations in the SO data analysis are based on topic modelling. The underlying assumption is that any relative changes in topics might be indicative of a micro-event. At the same time, these changes will unlikely be spotted by lower-level features, like Bag-of-Words. Below, I discuss two methods for topic modelling which are used in the current work: LDA and hSBM. LDA is a gold-standard topic modelling method, that is robust, scalable, and gives a relatively small number of topics as the result of optimisation. hSBM in its turn is a novel network-based approach that gives a large number of topics in the output. When choosing these methods, I wanted to offer comparability (LDA) as well as make sure that I am not missing on the novelty (hSBM) or the limitations of the feature space size.

Please note that when conducting the research, robust deep learning (DL) language model implementations were not available, hence it was not feasible to implement the topic modelling based on DL vector representations. At the time of designing experiments, the chosen models were considered as offering the best performance among topic modelling methods.

LDA Latent Dirichlet Allocation (LDA) topic modelling - is a generative model widely used in Natural Language Processing [55]. The model requires Bag-of-Words encoded text[1] as an input and outputs a distribution of topics present in the text. Each topic has a set of associated tokens. In the model training phase, the topics are automatically defined based on the word co-occurrences. Concretely: a word subset, occurring across multiple texts (messages, documents, posts, etc.) defines a latent topic. As an example: the words "oranges" and "fruits" are seen together more often than "oranges" and "transactions", while "transactions" are often found in the context of "banks". In case of a sufficient number of documents containing these subsets of words, the model would detect a topic for fruits, represented by "fruits" and "oranges"; and a financial topic, defined by the words "banks" and "transactions". An interpretation of topics is usually done manually, but the quality of the topics can be assessed in an automatic way.

[1] First, one gets a list of unique words in the corpus of text, then uses as a vector to encode the text units.

hSBM The hierarchical Stochastic Block Model - is a novel approach based on finding communities in complex networks and topic modelling [56]. Concretely: it represents the dataset as a bipartite graph of posts and words. The model outperforms LDA leading to better topic models. Also, hSBM does not require setting the number of topics allowing the model to find them naturally. We used this method to obtain a larger feature space and ensure that our feasibility analysis covers the dimensionality aspect. We computed the per-time step representations by taking topic means across posts in the time step, the same way as for LDA.

2.3 Machine learning pipeline

While machine learning algorithms are the heart of the data analysis pipelines, it is equally important to prepare the data, ensure pipeline consistency, take care of data-related biases, etc. Below, I introduce common machine learning pipeline concepts and best practices. Namely, I describe data preprocessing, feature selection, hyperparameter tuning, model evaluation, and model interpretation steps in the context of the conducted research. The workflow diagram is provided in Figure 2.3.

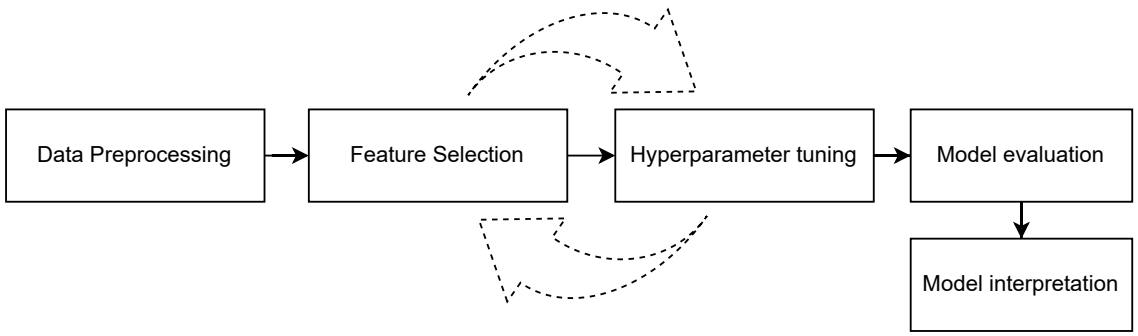


Fig. 2.3 Machine Learning workflow diagram. Feature selection and hyperparameter tuning steps can be performed in different orders and even iteratively as an optimisation task.

2.3.1 Data preprocessing

To make use of the data, it is necessary to design the experiment based on the expectations of the practitioner - the scientific hypothesis. Deriving the hypothesis, one should be aware of its assumptions and limitations. It is also essential to be familiar with the mechanics

behind the data collection as they largely define the limitations of the raw data. After the hypothesis is formulated, the dataset is designed - one identifies the output variable as well as the available raw features for building the feature space. After the feature space is built, one is aware of the dimensionality and value distribution, hence it is feasible to choose the estimator. Depending on the estimators used, different initial data processing steps might be required. For instance, data normalisation, standardisation, removal of outliers and capping [39]. For instance, logistic regression is known to be affected by outliers, dependent data entries and multicollinearity between the features [39]. More complex estimators, like ensembles, have way fewer restrictions and all limitations of the logistic regression do not apply to them. However, data preprocessing might still affect the final performance of the model.

Since data preprocessing is data- and problem-specific, I describe the dataset design and its preprocessing in the corresponding sections of the technical chapters.

2.3.2 Feature selection

Considering any machine learning model, one cannot infinitely increase the number of input features (also called feature space) as this would lead to increasingly sparse data entries in the considered feature space. The impact of large input feature spaces is well-known and studied [57]. One of the ways of mitigating the effects of the large feature space is by applying feature selection methods.

The objective of such methods is to choose the best suitable features from the large feature space and remove the ones bringing no new information. The choice can be guided by the feature variance, statistical tests, goodness of fit measures, performance on unseen data, etc. One can distinguish three different classes of feature selection methods:

- Filter - model-agnostic methods that perform the selection based on the properties of the features, like variance or correlation with the target. They are usually computationally efficient and relatively robust to overfitting [58].

- Wrapper - model-specific methods, evaluating performance on the subsets of features. They are capable of detecting feature interactions. However, due to their increased complexity, they are prone to overfitting. Finally, due to the number of feature subset variants, their evaluation might require significant computational resources [59]. One of the common examples is recursive feature elimination which is described below.

- Embedded - feature selection embedded into the machine learning model. This class might possess characteristics of both filter and wrapper methods. The most common examples are Decision tree and LASSO [60].

In the current work, it is essential to take feature interactions into account, hence I use Recursive feature Elimination with Cross-Validation (RFECV) [61], which is a commonly adopted standard for wrapper feature selection methods in the machine learning community. The method works as the following: the model is fitted to the full set of features, cross-validated, then user-defined k of the least important features are dropped, and the model is fitted again. This is repeated until there is a single feature in the model. The best subset is chosen based on the cross-validated performance of the model. Depending on the problem, various performance metrics can be used in the cross-validation phase. The pseudo-code for the RFECV algorithm is provided in Algorithm 1:

Algorithm 1 RFECV algorithm pseudo-code

Require: **N** features > 0;
p - minimum size of the feature subset (user-defined);
k - number of features removed from the subset per iteration (user-defined);
for a feature subset $\mathbf{S_i}$ of sizes $\in$ [**N**..**p**] with a step **k do**:
1. Train the model on the $\mathbf{S_i}$ feature subset;
2. Compute the performance metric of the cross-validation dataset;
3. Calculate feature importances;
4. Remove **k** least important features from the subset;

end for

2.3.3 Hyperparameter tuning

It is important to distinguish between model parameters and hyperparameters. Referring to hyperparameters, one usually means the configuration of the model - its complexity, loss function, optimisation algorithm, regularisation, etc. Model parameters are usually learned from the data during training. Since hyperparameters vary depending on the model, there are no strict guidelines on how to set them up for a specific problem. Hence, the process of tweaking the hyperparameters is usually data-driven - the performance of different model configurations is compared on a validation dataset. Since it is often not feasible to try all possible model configurations, there are various ways of finding the optimal set of hyperparameters. The simplest ones are grid-search and random search, where one either iterate over a pre-defined set of configurations or randomly chooses configurations from

the search space [62]. In case the models are computationally intensive, more sophisticated approaches, based on Bayesian search and assessment of the configurations of the partially fitted models [63]. Since models considered in the current work can be fitted within minutes and hyperparameter optimisation is not the focus of the thesis, I use the grid-search approach in all experiments.

2.3.4 Model evaluation

In the machine learning world, models usually need extra data for exploratory tests, model selection, feature selection, and hyperparameter tuning. Hence, the available data needs to be split into batches. Depending on the nature of the data, this can be done in many ways. For instance, a commonly used approach is K-Fold cross-validation - the method splits the data into training and test batches in K folds [64], test data from the fold sums to the available dataset with no overlaps. Depending on the variant of the K-Fold, additional conditions may be introduced, like class balance within folds. Moreover, there are more advanced ways of cross-validation, like nested K-fold. In its nested form, the cross-validation is done internally for each fold (as described above), hence the test data of all the folds cover the whole dataset multiple times [65].

K-Fold cross-validation is not applicable when there is a notion of time in the data and, hence, risk of introducing the look-ahead bias. In this case, the cross-validation is done sequentially, with the preserved temporal component - in this case, the training data is taken from earlier times than the test data [66]. Since in both domains (financial time series and TDT) there is a notion of time, I use the time-aware cross-validation approach I use throughout the experiments.

2.3.5 Model interpretation

Explainable machine learning (or explainable AI) is an active research area across many different disciplines [67]; however, the community has yet to reach a consensus on how to achieve perfect understanding as several challenges arise [68–70]. It is generally understood that by focusing on more understandable machine learning algorithms, such as logistic regressions, and with careful feature selection, one can greatly improve understanding. While this is a very active area in certain domains, such as medicine [71], comparatively little research has been applied to financial time series analysis.

Generally speaking, one can achieve explainability in AI in three ways [72]: 1) using more understandable algorithms, 2) reverse engineering estimator to understand how it comes to a decision, and/or 3) domain-specific adjustment of the input entries and feature

design. Whilst the first approach is more desirable of the two, as explainability comes inbuilt, there is often a trade-off between using simpler, more understandable models that may be less accurate, and more complex (less understandable) models that may well be highly accurate [73]. The second approach is gaining traction in recent years, focusing on using: i. Visualisations, ii. Natural language explanations and iii. Explanations by example [72]. The latter is based on selecting particular data entries and explaining the model decisions based on them. Example-based explanations lead to optimal input entries and feature space. Applied to financial time series, this means careful selection of the events (time series patterns) as well as the use of the most relevant features, leading to less convoluted model explanations [72].

Before introducing particular methods of model interpretation, it is necessary to define the vocabulary. Namely, it is crucial to distinguish the interpretability and explainability of the estimators. Interpretability is usually associated with white-box models, whose decisions can be interpreted explicitly and by design. Examples of interpretable models are logistic regression, decision trees, k-nearest neighbours, and ensemble methods, depending on the base estimator. Explainability is associated with the black-box models, which are not feasible to interpret explicitly and external tools involving certain approximations are needed [74]. The most common examples are neural networks, however, one might think of many models becoming harder to interpret after a certain level of complexity is reached. Hence, these terms should be used with care.

In the current dissertation (Chapter 5) I showcase a combination of novel state-of-the-art machine learning techniques and statistical methods to create effective data analysis pipelines that can both potentially be used in live settings while still being potentially by a practitioner.

Below I will detail the model interpretation methods common for the ML community, namely feature importance and local explanations, including LIME and SHAP. In Section 2.4 I additionally communicate model interpretation techniques common for the statistical assessment of the linear models.

Feature importance

For most of the ML estimators, one can obtain the feature importance of a fitted model [75]. The importance represents how useful the feature is for fitting to the considered dataset. The importance is assessed based on an estimator-specific criterion, hence should be treated as one of the ways of ranking features [76]. Feature importance is a useful approach to model interpretation, however, its limitation is that feature importance does not allow to assess of the contributions of features on a per-entry basis (as in the case of local explanations).

Local explanations

Local explanations address the limitation of the feature importance. They allow getting feature contributions for predicting a single entry. Local explanations are model-agnostic and in general, treat the model as a black box. There are two widely used local explanation methods - SHAP [77] and LIME [78]. Both these approaches create an interpretable surrogate model which is trained to replicate the target model outputs for the specified inputs. SHAP uses a game-theoretical model and LIME uses a linear LASSO model. Then, there is an assumption that if the outputs agree, then the feature contributions should be similar between the surrogate and the target model as well. This is quite a strong assumption, making local explanation methods highly susceptible to adversarial examples [79]. In the current dissertation, I use SHAP feature explanations as one of the ways of explaining the model outputs. Also, I assess its suitability for financial time series analysis by comparing it to the explicit model interpretations.

2.4 Statistical approach - ensuring generalisability & comparability

Machine learning is a rapidly emerging field of research, it already offers a set of incredibly powerful tools for data analysis. While ML methods usually offer state-of-the-art performance, it does not focus on statistical evaluation of the results. Hence, one might take the best of the two worlds by integrating the performance of the ML methods and statistically supporting the findings by using well-established statistical methods.

In the current section, I describe the set of tools that can be used as a part of the data processing pipeline in integration with machine learning methods. Namely, I introduce effect size as a way of model-agnostic feature ranking, analysis of linear models by assessing its goodness of fit and odds ratios, hypothesis testing, and correction for multiple comparisons.

2.4.1 Effect sizes

Generally, the effect size is a measure for calculating the strength of a statistical claim. The claim might be based on the mean difference, correlation of continuous features, the association of ordinal features, etc. [80] Larger effect sizes indicate that there is a larger difference between the treatment (method) and the control sample. Reporting effect sizes is considered a good practice when presenting empirical research findings in many fields [81–83]. Two types of effect sizes exist: relative and absolute. Absolute ones provide a raw

difference between the two groups and are usually used for quantifying the effect of a particular use case. Relatives are obtained by normalising the difference by the absolute value of the control group.

Variance explained measures the proportion to which a mathematical model accounts for the variation in the data. One of the most common ways to do this in the context of effect sizes is making use of Pearson Correlation [84], or its squared version, known as R-squared [85]. These measures allow assessing the proportion of variance shared by the two variables.

Another approach is to look at the differences in sample means, using a standardisation factor. Popular approaches include Cohen's d [86], which calculates the difference between two sample means with pooled standard deviation as the standardisation factor. However, it was found that the standard deviation may be biased as the standardisation factor, meaning that when the two means are compared and standardised by division as follows $\frac{u_1-u_2}{SD}$, if the standard deviation SD is used it may cause some bias and alternative standardisation may be preferred. This is rectified in Hedge's g [87] method, which corrects the bias using a correction factor when computing the pooled standard deviation. A further extension that can be added on top of this correction is to use av or rather an average variance instead of variance. The extension accounts for the correlation between the compared samples. The corrected measures are referred to as Cohen's d_av and Hedge's g_{av} [88, 89].

The effect size for categorical variable associations checks the inter-correlation of variables and can evaluate the probability of variables being dependent on one another. An example of this is the chi-squared test [90], also effective on ordinal variables. This test assesses the significance of the differences in feature category distributions between the tested groups by comparing them to the χ^2 distribution.

Another way of assessing effect size for ordinal variables is by using Cliff's Delta [91] as an effect size measure. It is also applied when the sample values are distributed non-normally. Cliff's delta is computed by first defining the delta function as:

$$\delta(x,y) = \begin{cases} +1, x > y \\ -1, x < y \\ 0, x = y \end{cases} \tag{2.5}$$

Then, using the defined function, one computes the relationship between all entries of the compared groups $\mathbf{X}$ and $\mathbf{Y}$ of sizes n and m as the following:

$$\delta = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \delta(x_i, y_j) \tag{2.6}$$

To generalise the findings for the samples to the population, it is advised to report .95 confidence intervals (CIs), representing the range where the effect size for the population might be found with the .95 probability [92].

When discussing effect sizes, their interpretation in different fields varies depending on the commonly observed differences in the effects between the test groups [93]. For instance, in social and medical sciences, it is common to consider Hedge's g effect sizes above 0.2, 0.5 and 0.8 as small, medium and large, respectively [89]. To my knowledge, there are no established effect size thresholds in the fields of financial time series and event detection in textual data. Hence, throughout the dissertation, I use the 0-threshold indicating the absence of the effect size and contribute to the establishment of domain-specific thresholds by reporting the effect size values. Interpreting the data, I am guided by the confidence intervals. I use effect sizes for two purposes - to quantify the effectiveness of the method (financial time series) as well as assess to what extent the classes are linearly separable by the considered features (textual data experiments). In both cases, I compute confidence intervals aiming to check the generalisability of the findings.

2.4.2 Model analysis

When considering statistical models, there is a need to investigate the model behaviour not just to interpret the decisions made, but also to check the validity of the model. Namely, whether it learns anything from the data, if it is affected by outliers, data non-linearity or excessively large feature space. Hence, there is a set of tools to do that, which I review below. Since from the statistical tools, I use only logistic regression, I focus on its analysis.

Goodness of fit

There are different ways of assessing the goodness of fit for linear models, including Akaike's Information Criterion, Bayesian Information Criterion, pseudo-R-squared, Log-likelihood ratio test, etc.

The log-likelihood ratio (LR) test is a way of statistical verification of the significance of the model fit. In this setting, the model is compared against its intercept version and the t-statistic is used to evaluate the significance [94]. While answering the question about significance, being a statistical test, it does not quantify the difference between the models. For that purpose, one might use different tools, like pseudo-R-squared.

Pseudo-R-squared is the concept derived from the R-squared measure of linear regression for the assessment of the fraction variance explained by the model. Or, also an improvement of the fitted model over the intercept (null) model. Since logistic regression

is a classifier, one cannot compute the R-squared between the actual data labels and the model output. Consequently, there is a need to assess the goodness of fit of the model differently. Many variants of the pseudo-R-squared scores exist. Among the most popular are McFadden (and its adjusted version), Nagelkerke, Cox-Snell, and Tjur [95]. It is a good practice to report multiple ones as they are affected by different aspects of the data, hence contrasting different aspects of the data. Their values usually lay within the $[0,1]$ range and the larger values mean a better fit of the estimator. However, there are measures adjusted for the number of features, which can have negative values as well. Also, there are ones that never reach 0 or 1 (like Cox-Snell).

Variance inflation factors

Variance inflation factors (VIFs) quantify the collinearity in linear models. And collinearity (or multicollinearity) is a phenomenon in which an input feature can be predicted from other input features by a linear model with a certain level of success. The danger of collinearity is that tiny changes in the data might lead to large changes in the model fit [96]. As advised by Andy Field et al., VIFs exceeding 10 indicate the presence of strong collinearity in the model, and at this point, the model is considered unreliable [39].

Data linearity

The linearity assumption of logistic regression is that there is a linear relationship between any continuous input feature and the logit of the output ($log(p/(1-p))$, where p is the output probability). To check whether the assumption is fulfilled, one fits the model with interaction features designed as features multiplied by its log transformations. If the interaction features are significant in the fitted model, the assumption is violated [39].

Influential outliers

It is known that linear model fits are susceptible to influential observations [97]. There are various ways of detecting them - from visual analysis of the added variable plots [98] to running statistical tests, like the Bonferroni outlier test [99].

2.4.3 Hypothesis testing

Another core component of the statistical assessment is hypothesis testing, which is a form of inferential statistics. With hypothesis testing, one aims to determine whether the findings are generalisable to the population. There are many ways to do that, depending on the

sample size, relationships between the entries and groups, domain-specific requirements, etc. Often the hypothesis testing is designed in a form of a general linear model [100]. Since we cannot empirically confirm the hypothesis, the general concept is that one defines the null hypothesis, which is considered the opposite of the alternative hypothesis and checks it for rejection. Strictly speaking, its rejection does not confirm the alternative hypothesis, but excludes its opposite, contributing to the likelihood of the alternative hypothesis being valid. Popular approaches include t-test [101], ANOVA [102], Wilcoxon [103], and many more, depending on the considered setting.

A t-test is a type of inferential statistic used to determine if there is a significant difference between the means of two groups, which may be related in certain features. The basic functioning is that you take a sample from two populations, and establish the null hypothesis for which the sample means are equal, it then calculates the mean difference and assesses the probability of it being observed by chance instead of due to an actual effect. If the mean difference is statistically significant, one rejects the null hypothesis. However, the t-test relies on several assumptions: 1) that the data is continuous, 2) that the sample is randomly collected from the total populations, 3) that the data is normally distributed, and 4) that the data variance is homogeneous [104]. This makes the t-test not suited to the analysis of small samples, where normality and other sample properties are hard to assess reliably.

An approach that doesn't face the same limitations is the Wilcoxon test [103]. The advantage of this approach is that instead of comparing means, it repeatedly compares rank differences, this means it will check the arithmetic average of the indexed position within a list. This type of comparison is applicable for paired data only and done on individual paired subjects, increasing the power of the comparison. However, a downside of this approach is that it is non-parametric. A parametric test is able to better observe the full distribution and is consequently able to observe more differences and specific patterns, however, as we saw with t-tests, they rely on stronger assumptions and are sometimes impractical. In the current work, I use the Wilcoxon test for hypothesis testing in the financial time series analysis, as well as the Log-likelihood ratio test to evaluate the significance of the model fit in the textual data experiments.

Throughout the thesis I encode the hypotheses in the following way: H_{0X} and H_{1X} correspond to null and alternative hypotheses, respectively, for research question X.

2.4.4 Correction for multiple comparisons

The more inferences are made (or hypotheses tested), the more likely erroneous inferences are to occur. Multiple comparisons arise when a statistical analysis involves multiple simultaneous statistical tests, each of which has the potential to produce the discovery, of

the same dataset or dependent datasets. Hence, the overall probability of the discovery increases. This increased chance should be corrected. Some methods are more specific, but there exist a class of general significance level α adjustments.

Examples of these are the Bonferroni Corrections [105], Holm-Bonferroni corrections [106] and Šidák Corrections [107]. The general idea follows from the following: given that the p-value establishes that if the null hypothesis holds what is the likelihood of getting an effect at least as large in your own sample. Then if the p-value is small enough, you can conclude that your sample is inconsistent with the null hypothesis and reject it for the population. So the idea of the corrections is that to retain a prescribed significance level α in an analysis involving more than one comparison, the significance level for each comparison must be more stringent than the initial α.

In the case of Bonferroni corrections, if n multiple tests are performed, we ensure that its p-value is less than $1.0 - \alpha/n$, then we can conclude, as previously, that the associated null hypothesis is rejected. Holm-Bonferroni corrections apply varying corrections depending on the ranked p-values from lowest to highest as $1.0 - \alpha/(n+1-k)$, where k is the rank of the experiment p-value. One can notice that for the first-ranked p-value Holm-Bonferroni correction is the same as Bonferroni. The following p-values have smaller significance levels α.

I correct the significance level for multiple comparisons in all the experiments across the dissertation. Namely, I use Holm-Bonferroni corrections for the textual data experiments and Bonferroni corrections for the financial time series data. The choice is motivated by a potentially weaker signal in the textual data as well as larger assumptions in the experiment design.

2.5 Topic detection and tracking

The current section introduces the concept of topic detection and tracking as well as highlights the research approaches in the field. Moreover, it details works in the field related to the current dissertation.

Initially, topic detection and tracking were limited to detecting and following events in news streams. Later, after Twitter started gaining popularity, the area benefited from the Twitter data [108]. Currently, these two streams are being developed in parallel with a certain overlap in methods and approaches [11].

One can distinguish two general types of event detection:

- Retrospective Event Detection (RED) - analysis of data from the past to discover new, unknown events;

- New Event Detection (NED) - usually done in an online fashion with a stream of data.

The field of event detection is very broad and there are various definitions of events in the literature. Their classification can be done by the following set of properties:

- Topic - there are topic-specific and general events.
- Geographic location - there are events, relevant for particular locations, like traffic information, concerts, etc. and fewer location-specific events, like financial reports, online entertainment project releases, etc.
- Time scale - depending on the event, the time horizon might be from hours to weeks and months.
- Reaction - not every event can be associated with a characteristic change in the data source. There might be different reasons for that, and depending on the methodology it may be addressed differently.

The event detection approaches may be also described as Feature Pivot or Document Pivot [11]. The first one involves a certain representation of documents/posts/messages within a time window with changes in the representation indicating the event. The Document Pivot focuses on the classification of the documents as related or not related to a particular event.

In terms of machine learning, the detection challenges can be treated as the following:

- Supervised Clustering - the entries are classified in a supervised fashion, then clustered [11];
- Semi-supervised Clustering - a small subset of the dataset is labelled and used to train the classifier, then the classified entries are clustered [11];
- Unsupervised Clustering - conventional clustering of entries is applied [109];
- Classification - the methods based purely on the classification of entries [11];
- Anomaly Detection - the discovery of entries that are anomalous for the considered dataset. In the current context, anomalies are expected to be observed within the topic, vocabulary, language, etc. [11].

Document Pivot approaches are developed for conventional event detection tasks and are not applicable to the micro-events introduced in the current work. The feature pivot approach is commonly used for collective event detection [110] and is applicable to the context of micro-event detection. I adopt the feature pivot approach and adjust it to spot subtle changes in the time steps.

2.5.1 Related research

A known example of an event detection system on Twitter is TwitInfo proposed by Marcus et al. [111]. The system performs clustering by topic and implies that peaking activity in a certain topic is a consequence of an event. Then the most relevant tweets are obtained using a keyword-based ranking. The ranking limits the generalisation of the TwitInfo to micro-events as single text units cannot be associated with the micro-events.

Another relevant topic is collective event detection, for example, flu epidemics. Collective events require multiple posts for detection [112, 113]. At the same time, each post can be successfully labelled as related or not related to the event (like in the study by Aramaki et al. [114]), distinguishing collective events from the introduced notion of micro-events.

Multiple instance learning (MIL)

It should be noted that it is not uncommon in TDT studies to use a temporal representation of textual data [115]. In NED, the temporal component is used either directly or implicitly as there it is essential to be aware of time when making the data available to the model. The current work does not diverge from this principle and uses the notion of time when processing the data and designing data representations.

Finally, the approach proposed in the thesis has its commonalities with Multiple-instance learning (MIL). MIL is defined as a supervised learning problem where instead of input-output pairs the model is provided with sets of entries associated with a particular class. In its simplest form, if a set is labelled negative, all its instances are negative, however, if the label is positive, there is at least one positive instance in it [116]. MIL use cases include scenarios where the effect is present but its cause is not clear. It has been used in a range of domains, like object detection [117], audio event detection [118], text categorisation [119], etc. The classification process usually involves an initial instance-level classification step either in a sequential way, using attention [120] and convolution neural networks [121], or in an independent way, using estimators like SVMs [122]. Then, there is a pooling procedure, summarising the output for all the elementary entries [118]. The schematic of the MIL workflow is provided in Figure 2.4.

MIL's approach is similar to the proposed in the current thesis in considering multiple instances as bags. However, it operates under the assumption that the elementary instances can be labelled which is not the case in this work.

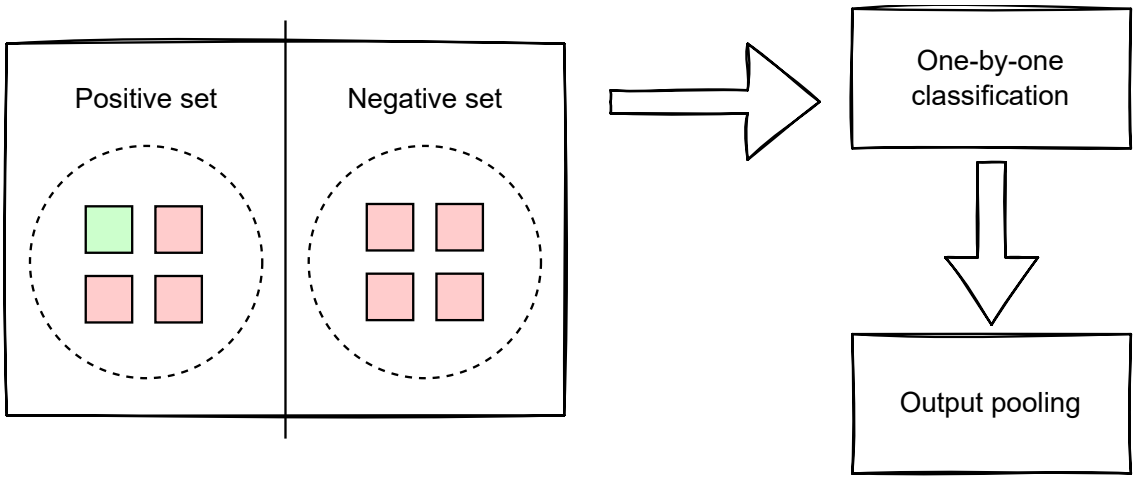


Fig. 2.4 Multiple instance learning workflow diagram.

Topic modelling

One of the methods used for detecting events is topic modelling. A well-established technique for that is Latent Dirichlet Allocation (LDA) [55], where changes in topics across time windows might indicate an event. Usually, dynamic or temporal implementations of LDA are used [123] for this purpose. However, they require certain adjustments, like incremental refitting of the model, and often do not have a scalable implementation available. Gerlach et al. recently proposed a stochastic block model-based method (hSBM) which outperforms LDA [56]. In the current work, I use both LDA and hSBM approaches to obtain text unit representations.

Applying the LDA topic modelling approach, I follow a well-established policy of model parameter optimisation. I feel that it is necessary to relate the existing works on SO data analysis using LDA to the way I approach the problem. Below, I overview of three works from the perspective of LDA application - by Barua et al. [124], Yang et al. [125] and Abellatif et al. [126].

First, is a study by Barua et al. on the analysis of topics and trends in the Stack Overflow community [124]. The study applies LDA as a technique for topic extraction, focusing on the empirical analysis of single messages as well as the investigation of trends. When assigning topics to posts, the authors use a probability threshold, not considering topics with probabilities below 0.1. It should be noted that even though it helps manual post interpretation, ML models might benefit from less probable topics as these may still contain valid information. The second study is by Yang et al. [125] on the analysis of security-related topics in SO data. The authors perform optimisation of the number of topics using a genetic algorithm with a Silhouette coefficient as an objective function. While LDA itself is a

computationally intensive method, the genetic algorithm adds an extra layer of complexity to it. It is justifiable in the research setting, but such an approach is unlikely to be used in the live setting. The study is focused on topic analysis rather than event detection. Finally, there is a recent study by Abellatif et al. analysing the chatbot community of Stack Overflow [126]. Abellatif et al. manually associate increased posting activity with two technology releases. It is infeasible to investigate these observations statistically due to the small sample size, hence in the following chapter I formalize the challenge by defining more subtle events of similar nature and proposing a dataset with a statistically sound pipeline for investigating them.

2.6 Financial time series

The current section introduces the concepts of financial markets mechanics, approaches to automated trading and its components, as well as highlights the research approaches in the field. Moreover, it details works in the field related to the current dissertation.

The financial market is an inevitable part of the modern world economy. This is a mechanism allowing speculation and exchange of almost any goods, including currencies (Forex), company shares in the form of stocks, options, futures contracts, metals, grains, dairy products, meat, energy, digital assets (blockchain products), and many more.

The classical financial market may be represented as a double-auction, where both buyers and sellers compete for the deal. The reason for that is the fact that the same indistinguishable asset is owned by many parties, as well as many parties, are interested in purchasing the asset. Here I do not consider more exotic markets, where the asset units are distinguishable as this is out of the scope of the research conducted for this thesis.

The mechanics behind the double-auction varies depending on the particular asset traded. Since the research of the thesis is conducted on two Chicago Mercantile Exchange (CME) Futures markets, I limit the explanations of the mechanics to this particular type of asset. The footprint of the double-auction is represented as time series of price changes, sent/cancelled orders, executed orders, as well as numbers of traded contracts. Having said that, one seems that we are dealing with multidimensional time series. Single-dimensional time series are known to be analysed using autoregressive models like AR, ARMA, SARIMAX, etc. [127]. While there are attempts to use these models for forecasting financial time series [128], more complex models are often required. Among other methods, forecasting attempts are made using deep reinforcement learning [129]. In the current work, I do not use these approaches as not aim to directly predict the future price. Instead, I focus on the classification of future price behaviour scenarios. The adopted approach is aimed at

simplifying the problem as much as possible without any loss of utility in the studied context. Operations of purchasing or selling assets on the financial markets are called investing or trading, depending on the time scales of the activity. To improve the readability of the work, I provide a glossary of the financial markets-related terms in Table 2.1.

Terms	**Definitions**
Futures contract	provide the means to trade a commodity (instrument) at a predetermined price at a specific time in the future.
Tick	represent a single movement upward or downward by a specific increment in the price for a specific instrument (e.g. 0.25$ for S&P futures).
Bar	used to identify a window of interest-based on some heuristic, and then aggregate the features of that window. It may contain several features and it is up to the individual to decide what features to select, common features include: *Bar start time*, *Bar end time*, *Sum of Volume*, *Open Price*, *Close Price*, *Min* and *Max* (usually called High and Low) prices, and any other features that might help characterise the trading performed within this window
Price Range Bar	the bar formed using the price action heuristic. Namely, the bar is considered completed when the difference between the price extrema equals N ticks. Where N is user-defined and depends on the trading frequency.
Volume	refers to the number of traded contracts (or shares) for a particular instrument
Volume profile	refers to the volume traded per price visualised as a vertical histogram for a range of prices over a certain time range
Liquidity	how rapidly stocks may be traded without affecting the market price. Has an impact on whether you are able to get the desired instrument at your choice of price (sell or buy)
Volatility	degree of variation for the price of a given instrument over a period of time.
Trading	the buying (long) and selling (short) of a financial instrument
Trading platform	is software that you use to conduct your trading. Allows for the centralised management of instruments and positions
Time & Sales	a set of features provided real-time for each trade executed in an exchange. Features include: *volume*, *price*, *direction*, *date*, and *time*

Order Book	the list of orders used by a trading venue to keep track of offers and bids by buyers and sellers for a particular instrument. These are then matched in specific order to execute a trade
Flat Market	is a stable state in which the range for the broader market does not move either higher or lower, but instead trades within the boundaries of recent highs and lows.
Trending Market	shifts in the market towards a raise or decrease in price compared to expected highs or lows. Used to buy and sell at the point where one is most likely to gain profit
Long positions	owning the asset for a time period with the expectation that the asset will go up in price
Short positions	if the expectation is that the price will decrease over time, you can short an asset to profit from its decreasing value.
Actionable ML	In the context of machine learning and more specifically algorithmic trading, actionability refers to the ability to act upon a prediction. This may directly relate to understanding the reason behind the prediction, in turn, allowing you to make informed decisions on how to act upon it.
Take-profit	an order that specifies a price at which the trade is closed with profit. The order remains open until the price is reached
Stop-loss	an order placed at a specific price that gets closed if the price lowers beyond a certain amount. This is meant to reduce potential losses incurred if the desired price is not reached.

Table 2.1 **Glossary**: This glossary contains some essential definitions used throughout Chapters 4 and 5 of the thesis. Sometimes similar definitions are reintroduced in specific contexts to centre the discussion.

Most current-day trading (with positions opened for less than 24 hours) is done electronically, through various available applications. Market data is propagated by the trading exchanges and handled by specialised trading feeds to keep track of trades, bids and asks by the participants of the exchange. Different exchanges provide data in different formats following predetermined protocols and data structures. Finally, the dataset is relayed back to a trading algorithm or human to make trading decisions. Decisions are then relayed back to the exchange, through a gateway, normally by means of a broker, which informs the exchange about the wish to buy (long) or sell (short) specific assets. This series of actions rely on the understanding of a predetermined protocol that allows communication between various parties. Several software tools exist to ensure that almost all these steps are done

for you, with the decisions made being the single point that may be uniquely done by the individual. After a match is made (either bid to ask or ask to bid) with another market participant, the match is conveyed back to the software platform and the transaction is completed. In this context, the main goal of ML is to automate the decision making in this pipeline.

When constructing algorithmic trading software or Automatic Trading Pipeline (ATP), each of the components of the exchange protocol needs to be included. Speed is often a key factor in these exchanges as a full round of the protocol may take as little as milliseconds. Therefore, to construct a robust ATP, time is an important factor. This extra layer adds further complexity to the machine learning problem. A diagram of what an ATP looks like in practice is presented in Figure 2.5.

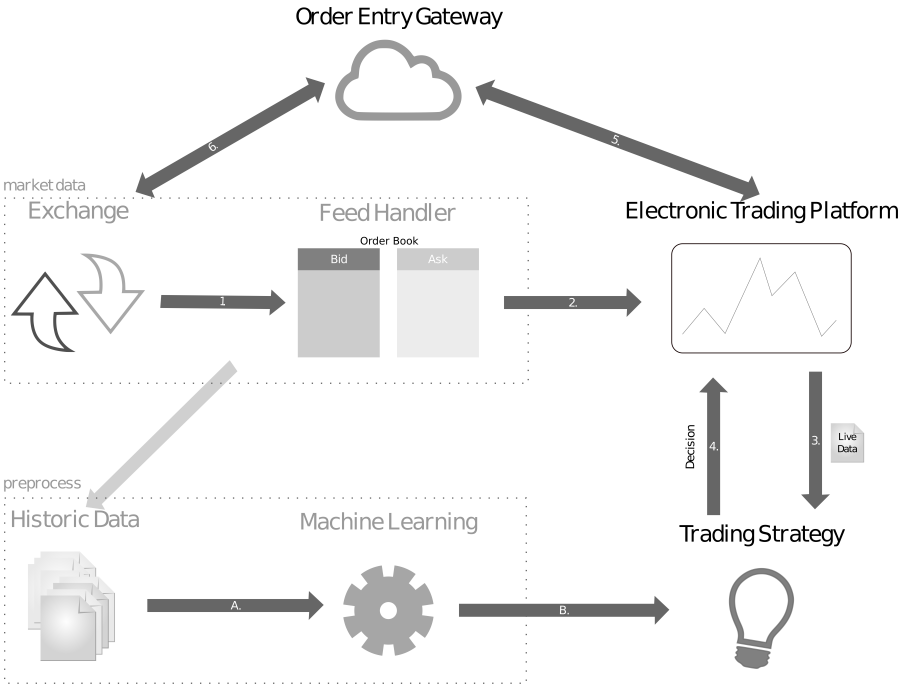


Fig. 2.5 Full overview of Automated Trading Platform components

In Figure 2.5 it can be observed that the main ML component is focused on training of the decision making and strategy. This is by no means a straightforward feat as successful strategies are often jealously guarded secrets, as a consequence of potential financial profits. Several different components are required, not the least analysing the market to establish components of interest. Historical raw market data contains unstructured information, allowing one to reconstruct all trading activity, however, that is usually not enough to establish persistent and predictable price action patterns due to the market non-stationarity. This characterisation is a complex process, which requires guidance and domain understanding.

While traditional approaches have focused on trying to learn from the market time series over the whole year or potentially across dozens of years, more recent work has proposed the usage of data manipulation to identify key events in the market [130], this advanced categorisation can then become a focus for machine learning input to improve performance.

This methodology focuses on identifying the states of a financial market, which can then be used to identify points of drastic change in the correlation structure, whether positive or negative. Previous approaches have used these states to correlate them to worldwide events and general market values to categorize interesting scenarios [131], showing that using these techniques the training of the strategy can be greatly optimised. At the same time, there is a lack of research proposing a full-stack automated trading platform and evaluating it using statistical methods.

2.6.1 Financial data types

There are several different types of financial data, and each of these has a different role in financial trading. They are widely classified into four categories: i. *Fundamental Data*, this kind of data is formed by a set of documents, for example, financial accounts, that a company has to send to the organisation that regulates its activities, this is most commonly accounting data of the business, ii. *Market Data*, this constitutes all trading activities that occur, allowing you to reconstruct a trading book, iii. *Analytics*, this is often derivative data acquired by analysing the raw data to find patterns, and can take the form of fundamental or market analytics, and iv. *Alternate data*, this is extra domain knowledge that might help with the understanding of the other data, such as world events, social media, Twitter and any other external sources.

2.6.2 Market states

In their seminal work Munnix et al. [130] first proposed the characterisation of market structures based on correlation. Through this, they were able to detect key states of market crises from raw market data. This same technique also allows the mapping of drastic changes in the market, which correspond to key points of interest for trading. By using k-means clustering, the authors were able to predict whether the market was approaching a crisis, allowing them to react accordingly and construct a resilient strategy. Their successful results initiated a lot more research in this area, where methods more advanced than correlation like random matrix theory [132] and copulas [133] were used to assess the market states. Their way of analysing a market as a series of states proved to be a winning strategy allowing for more focused decision making and improving the understanding of the market. In the

context of the current work, the crises and the drastic changes of the markets can be seen as events. Following on from this same approach I seek to characterise the market as a series of points of interest (micro-events) and understand whether the market structure allows their classification into different price action scenarios. This constitutes the initial stage of ATP or preprocessing, and with their approach, several steps of manual intervention are still required.

2.6.3 Market data preprocessing

To prepare data for processing, the raw data is structured into predetermined formats to make it easier for a machine learning algorithm to digest. There are several ways to group data, and various different features may be aggregated. The main idea is to identify a window of interest-based on some heuristic, and then aggregate the features of that window to get a representation, called *Bar*. Bars may contain several features and it is up to the individual to decide what features to select, common features include *Bar start time*, *Bar end time*, *Sum of Volume*, *Open Price*, *Close Price*, *Min* and *Max* (usually called High and Low) prices, and any other features that might help characterise the trading performed within this window. The decision of how to select this window may be a make or break for your algorithm, as it will mean you either have good *useful* data or data not representative of the market. An example of this would be the choice of using time as a metric for the bar window, e.g. take n hours snapshots. However, given the fact there are *active* and *non-active* trading periods, one might find that only some bars are actually useful using this methodology. In practice, the widely considered way to construct bars is based on the number of transactions that have occurred or the volume traded. This allows for the construction of informative bars which are independent of timing and get a good sampling of the market, as it is done as a function of trading activity. There are of course many other ways to select a bar [6], so it is up to the prospective user to select the one that works for their case.

2.6.4 Local price extrema

In mathematics, an extremum is any point at which the value of a function is the largest (maximum) or smallest (minimum). These can either be local or global extrema. At the local extremum, the value is larger/lower at immediately adjacent points, while at a global extremum the value of the function is larger than its value at any other point in the interval of interest. If one wants to maximise their profits theoretically, their intent would be to identify an extremum and trade at that point of optimality, i.e., the peak. This is one of the many ways of defining the points of optimality.

As far as the algorithms for an ATP are concerned, they will often perform with *active* trading, so finding a global extremum serves little purpose. Consequently, local extrema within a pre-selected window are instead chosen. Several complex algorithms exist to do so, with use cases in many fields such as biology [134]. However, the objective is actually quite simple: identify a sample for which neighbours on each side have lower values for maxima and higher values for minima. This approach is very straightforward and can be implemented with a linear search. In the case where there are flat peaks, which means several entries are of equal value the middle entry is selected. Two further metrics of interest are, the prominence and width of a peak. *The prominence of a peak* measures how much a peak stands out from the surrounding baseline of the near entries, and is defined as the vertical distance between the peak and lowest point. The *width of the peak* is the distance between each of the lower bounds of the peak, signifying the peak duration. In the case of peak classification, these measures can aid a machine learning estimator to relate the obtained features to the discovered peaks. This avoids attempts to directly relate the properties of narrow or less prominent peaks to wider or more prominent peaks. These measures allow for the classification of good points of trading as well as giving insight as to what led to this classification with prominence and width.

Historically, there has been an intuition that the changes in market price are random. By this it is understood that whilst volatility peaks are linked to certain events, it is not possible to extract them from raw data. Despite this, volatility is still one of the core metrics for trading [135]. In an effort to statistically analyse price changes and break down key events in the market Caginalp & Caginalp [136], propose a method to find peaks in the volatility, representing price extrema. The price extrema represent the optimal point at which the price is being traded before a large fluctuation. This strategy depends on the exploitation of a shift away from the optimal point to either sell high or buy low. The authors describe the supply and demand of a single asset as a stochastic equation where the peak is found when the maximum variance is achieved. Since the implied relationship of supply and demand is something that will hold true for any exchange, this is a great fit for various different instruments. In a different context, Miller et al [137], analyse Bitcoin data to find profitable trading bounds. Bitcoin, unlike more traditional exchanges, is decentralised and traded 24h a day, making the data much smoother and with less concentrated trading periods. This makes the trends harder to analyse. Their approach manipulates the data in such a way that it is smoothed, through the removal of splines, this seeks to manipulate the curves to make its points more closely related. By this technique, they are able to remove outliers and find clearer points of fluctuation as well as peaks. The authors then construct a bounded trading strategy that proves to perform well against unbounded strategies. Since Bitcoin has more

decentralised access, and by the very nature of those investing in it, this also reduces barriers to entry, making automated trading much more common. This means that techniques to identify bounds and points of interest in the market are also more favoured and widely used.

2.6.5 Derivation of the market microstructure features

A market microstructure is the study of financial markets and how they operate. Its features represent the way that the market operates, how decisions are made about trades, the price discovery process and many more [138]. The process of market microstructure analysis is the identification of why and how the market prices will change, in order to trade profitably. These may include, 1) the *time between trades*, as it is usually an indicator of trading intensity [139] 2) *volatility*, which might represent evidence of good and bad trading scenarios, as high volatility may lead to an unsuitable market state [140], 3) *volume*, which may directly correlate with trade duration, as it might represent informed trading rather than less high volume active trading [141], and 4) *trade duration*, high trading activity is related to greater price impact of trades and faster price adjustment to trade-related events, whilst slower trades may indicate informed single entities [142]. Whist several other options are available they are often instrument-related and require expert domain knowledge. In general, it is important to tailor and evaluate your features to cater to the specific scenario identified.

One such important scenario to consider when catering to prices is whether the price action is caused by aggressive buyers or sellers. In an Order Book, a match implies a trade, which occurs whenever a bid match asks and conversely, however, the trade is only ever initiated by one party. In order to dictate who is the *aggressor* is in this scenario (if not annotated by the marketplace), the tick rule is used [143]. The rule labels a buy initiated trade as 1, and a sell-initiated trade as -1. The logic is the following an initial label l is assigned an arbitrary value of 1 if a trade occurs and the price change is positive, then $l = 1$ if the price change is negative, and $l = 0$ and if there is no price change l is inverted. This has been shown to be able to identify the aggressor with a high degree of accuracy [144].

2.6.6 Automated trading systems

An automated trading system is a piece of code that autonomously trades in the market. The goal of such constructs is the identification of a market state in which a trade is profitable, and to automatically perform the transaction at that stage. Such a system is normally tailored for a specific instrument, analysing unique patterns to improve the characterisation. One such effort focusing on Forex markets is, Dempster & Leemans [145]. In this work, a technique using reinforcement learning is proposed to learn market behaviours. Reinforcement

learning is another area of machine learning concerned with how software agents ought to take actions in an environment in order to maximize the notion of cumulative reward. This is achieved by assigning positive rewards to desired actions and negative rewards to undesired actions, leading to an optimisation towards actions that increment rewards. In financial markets, this naturally corresponds to profitable trades. Using this approach, the authors are able to characterise when to trade, perform an analysis of associated risks, and automatically make decisions based on these factors. In the more recent work, Booth et al. [131], describe a model for seasonal stock trading using an ensemble of Random Forests. This leveraged variability in seasonal data to predict the price return based on these events taking place. Their random forest-based approach reduces the drawn-down change of peak to through events. Their approach is based on domain knowledge of well-known seasonality events, usual approaches following this technique find that whilst the event is predictable, the volatility is not. So their characterisation allows for predicting which events will lead to profits. The Random Forests are used to characterise features of interest in a time window, and multiple of these are aggregated to inform the decision process. These ensembles are then weighted based on how effective they are and used to inform the decision with higher weights having more input. Results across fifteen DAX assets show increases in profitability and in prediction precision.

As can be seen, statistical and machine learning techniques have been successfully applied in a variety of scenarios, proving effective as the basis of automatic trading and identification of profitable events. This makes the further investigation into more advanced machine learning techniques a desirable and interesting area. In the current thesis, I expand on these previous concepts to seek new ways to characterise the market via the micro-events.

Trading strategy

In the current thesis, due to the data time frames, I only consider strategies for day trading. Active trading seeks to gain profit by exploiting price variations, to beat the market over shorter holding periods. Perhaps the most common approach is trend-based strategies. These strategies aim to identify shifts in the market towards a raise or decrease in price and sell at the point where they are likely to gain profit. The second common approach is called the flat strategy. Unlike trending markets, a flat market is a stable state in which the range for the broader market does not move either higher or lower, but instead trades within the boundaries of recent highs and lows. This makes it easier to understand changes in the market and make a profit with a known market range. The role of machine learning in both these strategies is to predict whether the market is entering a state of flatness or trending respectively.

To evaluate the effectiveness of the trading strategy, the Sharpe Ratio is used. This is a measure for assessing the performance of an investment or trading approach and can be computed as the following:

$$S = \frac{R_p - R_f}{\sigma_p}, \tag{2.7}$$

where R_p & R_f correspond to the portfolio and risk free returns, respectively, and σ_p is a standard deviation of the portfolio return. While the equation gives a good intuition of the measure, in practice, its annualised version is often computed. It assumes that daily returns follow the Wiener process distribution, hence to obtain annualised values, the daily Sharpe values are multiplied by $\sqrt{252}$ - the annual number of trading days. It should be noted that such an approach might overestimate the resulting Sharpe ratios as returns auto-correlations might be present, violating the original assumption [146].

2.6.7 Backtesting

In order to test a trading strategy, its evaluation on historical data is performed to assess the profitability. While it is possible to do so on real market data, it is generally more favourable to do so on historical data to get a risk-free estimation of performance. The notion is that a strategy that would have worked poorly in the past will probably work poorly in the future, and conversely. However, as you can see, a key part of backtesting is the risky assumption that past performance predicts future performance.

Several approaches exist to perform backtesting and different things can be assessed. Beyond testing trading strategies, backtesting can show how positions are opened and the likelihood of certain scenarios taking place within a trading period. The more common technique is to implement the backtesting within the trading platform, as this has the advantage that the same code as live trading can be used. Almost all platforms allow for simulations on historical data, although it may differ in form from the raw data one may have used for training. For more flexibility, one can implement their own backtesting system in languages such as Python or R. This specific approach enables for the same code pipeline, that is, training the classifier to also test the data, allowing for much smoother testing. Whilst this will ensure the same data that is used for training may be used for testing, it may suffer from differences in the trading software that might skew the results. Another limitation of this approach is that there is no connection to the exchange or the broker, there will be limitations on how order queues are implemented as well as the simulating of latency which will be present during live trading. This means that the identification of slippages, which is the difference between where the order is submitted by the algorithm and the actual market entry/exit price, will differ and impact the order of trades.

2.7 Conclusion

In the current chapter, I have set up the scene for the later-described studies. At this point, the reader should have a feeling of the major design decisions taken throughout the dissertation. "Topic detection and tracking" together with "Financial time series" sections give an idea about the sources of the data complexity and non-stationarity by drafting the underlying mechanisms for the domains of financial markets and forum-like communications. I believe that they also allow the reader to appreciate the complexity of the problem of micro-event detection and classification.

"Machine learning" section describes only a tiny fraction of the available breadth of machine learning methods and their application contexts. At the same time, it suggests a need for an informed choice of machine learning algorithms.

It is evident that when tackling such a challenging problem as micro-event detection & classification, it is necessary to take all precautions for obtaining unbiased results. "Machine learning pipeline" and "Statistical approach - ensuring generalisability & comparability" sections provide a set of essential tools and practices for rigorous research in the considered setting. Statistical tools support that idea further, allowing one to achieve a higher level of interpretability (logistic regression, odds ratios, variable coefficients, pseudo-R-squared), generalisability (effect sizes with confidence intervals) and reproducibility (hypothesis testing) of the results.

Chapter 3

Event Detection in Forum-like Textual Data

Topic Detection and Tracking (TDT) is a very active research topic within the area of text mining, generally applied to news feeds and Twitter datasets, where topics and events are detected. The notion of "event" is broad, but typically it applies to occurrences that can be detected from a single post or message. As it was mentioned in the earlier chapters, none to little attention has been drawn to what I call "micro-events", which, due to their nature, cannot be detected from a single piece of textual information. From the previous chapter one can infer that TDT is commonly done on textual data. Hence, to avoid introducing too many degrees of freedom at once, I first attempt detecting micro-events from textual communications data. The chapter investigates the feasibility of micro-event detection on textual data using a sample of messages from the Stack Overflow Q&A platform and Free/Libre Open Source Software (FLOSS) version releases from Libraries.io dataset. I build pipelines for detection of micro-events using three different estimators whose parameters are optimised using a grid search approach. I consider two feature spaces: LDA topic modelling with sentiment analysis, and hSBM topics with sentiment analysis. The feature spaces are optimised using the recursive feature elimination with the cross-validation (RFECV) strategy.

In the conducted experiments, I investigate whether there is a characteristic change in the topics distribution or sentiment features before or after micro-events occur, and thoroughly evaluate the capacity of each variant of the proposed analysis pipeline to detect micro-events. Additionally, I perform a detailed statistical analysis of the models, including influential cases, variance inflation factors, validation of the linearity assumption, pseudo R^2 measures and no-information rate. After linear models are analysed, I apply non-linear models, such as Random Forests and CatBoost. The proposed pipeline, that consists of

statistics and machine learning elements, proves to be a robust and thorough approach to micro-event detection and is reused throughout the thesis. Finally, in order to study the limits of micro-event detection, I design a method for generating micro-event synthetic datasets with similar properties to the real-world data and use them to identify the micro-event detectability threshold for each of the evaluated classifiers. The background to the experiments is provided in Chapter 2, Section 2.5.

The idea of the research was conceptualised in discussions with Thomas Gross and Jaume Bacardit. Both my supervisors contributed by providing feedback on the experiment design, methodology, and writing of the original paper.

3.1 Materials and methods

In the current section, I state the research question of this chapter, formulate the null and alternative hypotheses, introduce data sampling strategy, label design, data preprocessing steps, and, finally, means for hypothesis assessment and model analysis.

3.1.1 Research gap & aims

As I already mentioned in the introductory chapter of the dissertation, relatively little attention has been paid to micro-events detection, both in textual data processing and financial time series analysis. The current chapter fills the gap with a detailed feasibility study of the micro-events in forum-like textual communications.

Aiming to investigate, whether micro-events can be detected in the software engineering texts, I formulate the research question in the following way:

RQ: Is it feasible to associate FLOSS version releases with a characteristic change in the associated community interactions in a form of textual communications on a Q&A resource - Stack Overflow?

To formally evaluate the research question, I formulate the null and alternative hypotheses:

H_{01}: An event is not associated with a change in the topic distribution or sentiment, representing textual communication of the community.

H_{11}: There is an associated change in the topic distribution or sentiment, representing the textual communication of a community and the event.

The significance level for the study is $\alpha = 0.05$. The multiple comparisons are accounted for using Holm-Bonferroni corrections. Each type of dataset (Selenium, Django, Multiple) is considered a separate experiment family. I apply the corrections when judging the sig-

nificance of the models or model features - each case is explicitly mentioned in the Results section. Moreover, I have made the data and the code of the study available via Zenodo [147] and Newcastle University data storage [148].

3.1.2 Data description and sampling strategy

I use two data sources in the study: stackoverflow.com and libraries.io. Both of them comply with Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) allowing data sharing and adaption. Older posts of StackOverflow comply with the earlier versions of the license (2.5 and 3.0), still allowing adapting and sharing the data. The datasets are downloaded from the libraries.io and archive.org web pages, where they were originally shared by the owners of the resources. At every stage of the study, I fully comply with the terms and conditions of stackoverflow.com, libraries.io and archive.org web-resources.

The study sample consists of a subset of packages related to the Django web framework, selected based on the presence of the associated discussions on the SO platform and having associated event entries in the libraries.io 1.4.0 dataset [149][1]. The proposed sample allows the investigation of two dataset configurations - a single package SO data sample with associated version releases, and multiple packages.

I manually obtain the initial list of packages from djangopackages.org. This is done in full compliance with the web page terms of use. Since I aim to ensure the theoretical possibility of event detection for every package, I require the package-associated SO community to be large enough and active. I perform initial filtering of the packages by the number of package followers on Github - all packages with <1000 followers were dropped. In the next step, I filter packages by the number of SO posts associated with them - communities with a number of messages < 1% of the number of posts associated with the Django package are dropped. I choose Django as a reference package as it is a well-established package with a large community and package updates. Moreover, Django relies on a number of other packages which might have a positive performance impact on the multiple-package datasets.

I download a complete data dump from Stack Overflow, version June 2018, then filter the messages by checking the presence of the selected package names in the body and tags of the messages. Finally, I split the dataset into training and test sets by using the first 60% of messages (chronologically) for training and the remaining 40% for test. The dataset design is illustrated in Figure 3.1.

[1] Libraries.io collects, among other information, release dates of FLOSS packages.

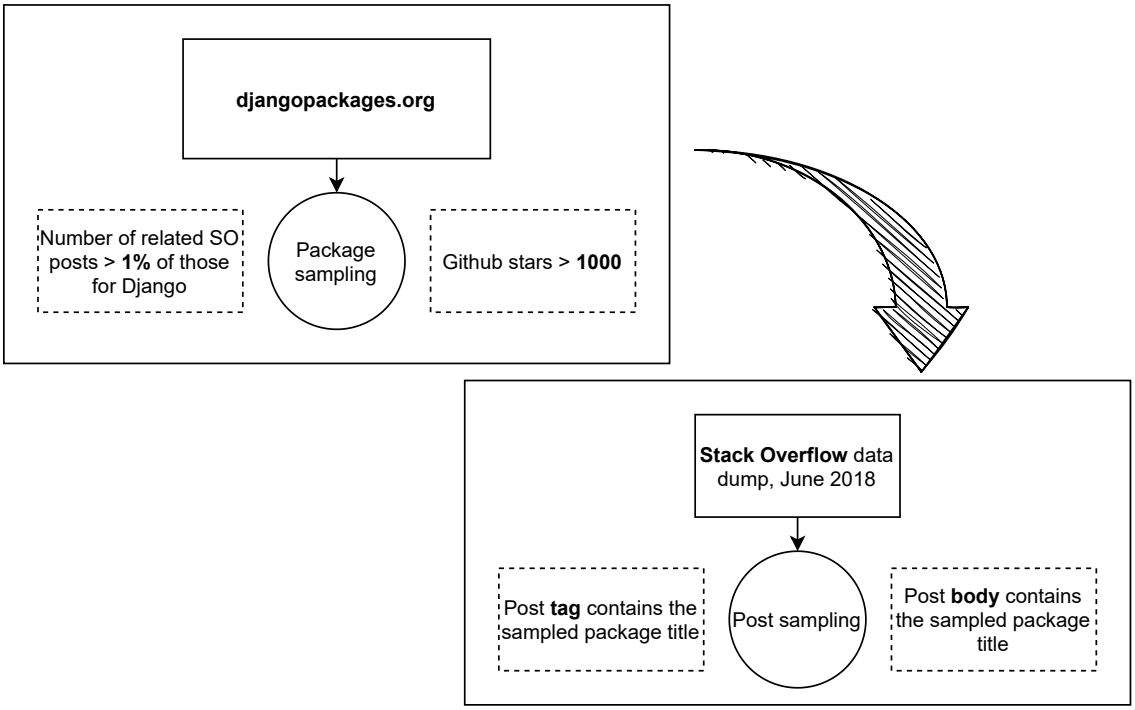


Fig. 3.1 The figure illustrates the dataset design steps. First, the packages are sampled based on popularity and SO community presence, then the relevant SO posts are sampled. The dashed lines indicate the sampling conditions.

Throughout the chapter, I use the following dataset naming convention: [package][type of event][time-step]. I study the datasets of 3 types: Multiple (dataset including messages and releases of 7 different packages), Django and Selenium. The event types are major, minor and patch updates. And the time-steps are either event-based or calendar week-based (c.w.).

3.1.3 Dataset design

Class labels

From the list of release dates provided by libraries.io, I extract three types of FLOSS version releases: patch, minor and major updates. When identifying the event types, I apply the Semantic Versioning 2.0.0 convention [2]. These three package types are used for creating separate datasets. The major limitation of the dataset is that there is no guarantee that every package maintainer follows the current convention. At the same time, the dataset covers all the public package manager version releases.

[2] https://semver.org/#semantic-versioning-200

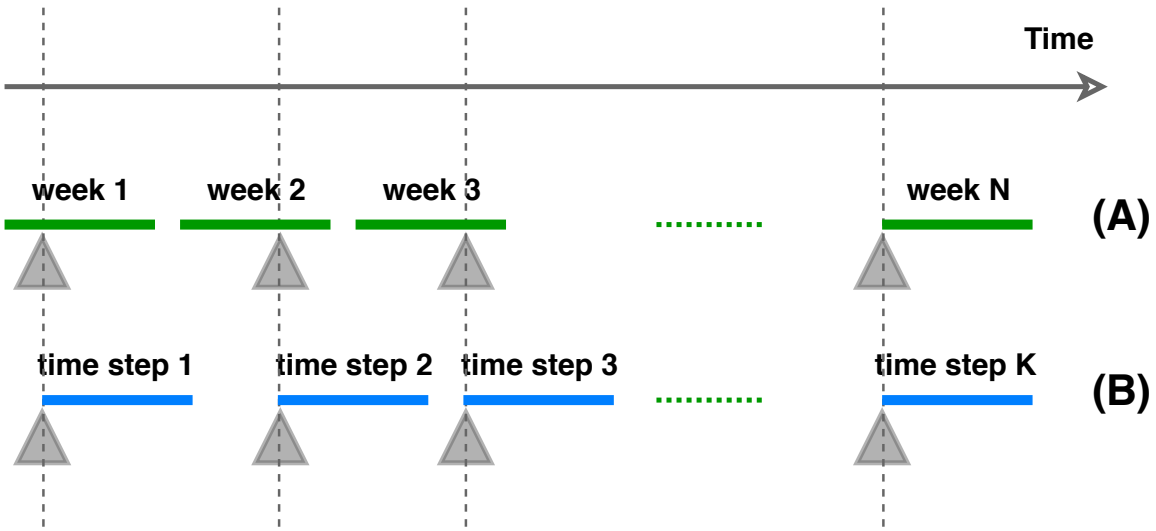


Fig. 3.2 The figure illustrates two designs of time-steps - calendar week-based (A), and event-based (B). Grey triangles indicate events that take place at the beginning of event-based time-steps and at any moment of calendar week-based time-steps.

Datasets

I use the two largest packages (by community size) to generate the single-package datasets, namely, Django and Selenium. Additionally, 7 packages (listed in Figure 3.6) were used to generate the multiple-packages dataset. I distinguished all three types of updates for the multiple-packages dataset and only patch and minor updates for the single-package datasets as major updates are too sparse to use them in the single-package datasets. Finally, I have designed two types of time-steps.

Time-steps

The first design is based on calendar weeks - every calendar week is a single time-step. Events occur on any day of the time-step (Figure 3.2, (A)). If multiple events occur in the same time-step, only the most significant (major>minor>patch) is considered while labelling. This ranking is based on the assumption that major updates might include characteristics from the patches, and the minor ones include the properties of the patches. The ranking is necessary to avoid interference of the effects from the different update types. The control time-steps are those in which no event of any type occurred. This approach is aimed to study prior and posterior characteristic changes in communication.

The second time-step generation is event-based. Each event occurs on day 0 of the time-step (Figure 3.2, (B)). Whenever there are multiple events with less than 7 days of the gap, time-steps overlap (i.e. the time-steps share messages). As control time-steps, I defined

the second week of 14-day intervals with no event occurring within that period. Because of this restrictive definition, messages of certain dates (first week of 14 day periods) could not be used for either type of time-step and had to be dropped. This design decision minimizes the noise characteristic changes caused by prior events.

The more days a time-step consists of, the fewer entries one gets in the dataset. Based on the frequencies of the events and common sense, I set the time-step length to 7 days. It allows on average 3.5 days for a developer to relate to the event considering the calendar week granularity of the time-steps, and 7 days - for the event-based approach. Daily time-steps were studied as well (the experiments are not reported, but available in the code supplement), however, they would require much faster responses from the community and more prolific communication.

3.1.4 Preprocessing

Prior to applying the NLP tools, the code snippets and HTML tags are removed from the body of the messages. Initially, 3 techniques are applied to design the features: sentiment analysis, LDA topic modelling, hSBM topic modelling. The central analysis of this chapter is conducted using two feature spaces: LDA with sentiment analysis and hSBM with sentiment analysis. These feature spaces allow investigation of both small- and large-dimensional experiment setups, and at the same time give the best trade-off between the dimensionality and the contained information. The features are extracted on a per-message basis with further grouping by the time-step.

Sentiment analysis features are generated using the NLTK python package [150], specifically using the Vader method [151]. There are 4 features generated for each message: negative, neutral, positive and compound components. The latter is a 1d representation of the sentiment. I include sentiment analysis in every feature space, due to its compactness. The sentiment features are coded as "sentiment_<component>", where the components are negative, neutral, positive and compound.

Latent Dirichlet Allocation (LDA) topic modeling is described in the Background Chapter, Section 2.2.2.

I optimize and fit the model on the training part of the dataset and compute fixed-dimensional vectors for posts in the whole dataset. Each dimension represents the probability of the post belonging to a particular topic. To get the per-time-step representations, I compute per-topic averages across the messages in the time-step.

To apply LDA, I choose the gensim package with a robust and scalable implementation of the method [152]. I built the vocabulary from the lemmatised messages with included bi- and tri-grams. The topic number is optimised on the training dataset. Coherence measure

(C_V) was used to find the optimal number of topics. I compute the coherence for 3 random-seeded models to ensure the result is not randomness-biased. Finally, I choose a number of topics using the Elbow method heuristic.

To verify the soundness of the obtained LDA topics, I manually inspect the top 5 messages associated with each topic, ranked by the gap in the probabilities between the target topic and the next closest one. Finally, showing the general theme of the discussions, I assign names to the LDA topics. The feature names are encoded as "lda_topic__<topic name>".

Hierarchical Stochastic Block Model (hSBM) is described in the Background Chapter, Section 2.2.2. I use a default configuration of the hSBM not limiting the number of topics. Since the approach is novel, computationally more intensive than LDA and there is no well-supported implementation, I assume that due to a large number of topics, randomness does not affect the results as largely, as in the case of the LDA. Finally, due to the large final number of topics, I do not interpret them, setting their names to the order numbers from the list they are obtained.

I do not conduct any formal analysis of other feature spaces (like Bag-of-Words and TextRank) for two reasons:

- Their dimensionality is up to 2 orders larger in comparison to the used approaches, making the models susceptible to the curse of dimensionality [57];
- I assume that lower-level features, such as Bag-of-Words (BoW), contribute to the output as subsets. And discovering feature interactions in the setting where the number of features is orders larger than the number of entries is bound to detect spurious interactions and, consequently, leads to interpretation errors.

Standardisation As I wanted to make the feature values comparable across time-steps, we standardised them as the following:

$$z = \frac{x_i - \mu}{\sigma}, \tag{3.1}$$

where μ is the sample average and σ is the sample standard deviation. Both values are obtained from the training partition of the data.

3.1.5 Analysis pipelines

I perform the data analysis using three different estimators - Logistic Regression (LR), Random Forest (RF) and CatBoost (CB) [153]. I select those to cover a rigorous statistical approach (LR), a well-known machine learning approach (RF) and a robust state-of-the-art machine learning method (CB). For the Logistic Regression, I perform a full stack of model

investigation techniques recommended by Field et al. [39] to ensure that the obtained models are statistically reliable. Both Random Forest and CatBoost are based on decision trees and there is a well-established set of model interpretation tools, which I apply in the current chapter.

Uniform methodology

Due to the nature of micro-events, their detection is expected to be challenging. Hence, there is a need for robust methodology that would allow not only measuring performance reliably but also analyse the causes of poor performance. I propose such a methodology (Figure 3.3). It is applicable to all the considered estimators in terms of feature selection, model optimisation, fitting, and performance assessment. I ensure an as uniform pipeline design as possible between the three estimators. However, the model analysis differs since the approaches are model-specific. The proposed methodology is applicable to textual communications and financial time series data, as it will be demonstrated in the following chapters.

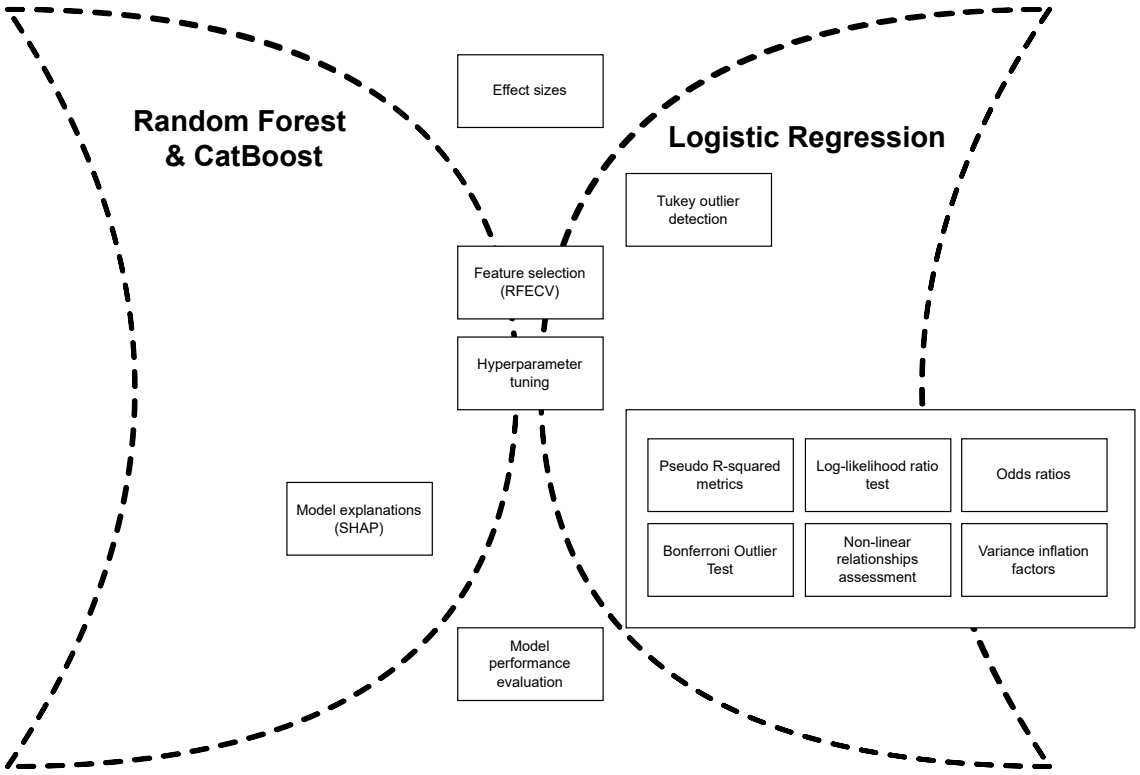


Fig. 3.3 The diagram illustrates a proposed sequence of steps for non-linear (Random Forest and CatBoost) and linear (Logistic Regression) models. Elements belonging to both and placed outside of the "wings" are relevant for both types of models.

Below, I justify every step of the proposed approach.

Description of the Logistic Regression pipeline

Effect sizes I compute effect sizes to get a model-agnostic quantitative measure of the phenomenon magnitude. The benefit of the method is the ability to directly compare the strengths of different phenomena irrespective of the models used in the study. I use Cliff's Delta [91] as an effect size measure, due to a non-normal feature values distribution. R *effsize* package is used to compute the effect sizes. Additionally, I report the .95 confidence intervals (CIs). Finally, I correct the CIs for multiple comparisons.

Outliers Prior to fitting LR models, I cap outliers using the Tukey method [154], as they are known to unduly affect the Logistic Regression model results. I use R $boxplot.stats()$ implementation for this purpose. I obtain the capping thresholds from the training partition of the data and cap the whole dataset.

Feature selection I perform feature selection using recursive feature elimination with cross-validation (RFECV), which is described in the Background, Section 2.3.2.

Parameter tuning and model fitting The only tunable parameter in the *R glm* (generalised linear model) implementation of LR is class weights. However, uneven class weights affect the base probability of the model as well as skew output distribution. Consequently, it decreases the interpretability of the model and makes any finding inaccurate and inapplicable to the population. I proceed with even class weights to avoid the mentioned issues.

After fitting the model, the Log-Likelihood Ratio (LLR) test is performed to check the significance of the model fit for the optimal feature space. It is important to note that no test data is used to conduct the test and it does not say anything about the model performance.

Model analysis The assessment of the model quality involves the following procedures:

1. I estimate the goodness of fit using Tjur, Nagelkerke, Cox-Snell, and Adjusted McFadden pseudo R^2 measures.

2. I compute odds ratios of the model and plot them as a forest plot with the .95 confidence intervals. The intervals are corrected for multiple comparisons. Since the data are standardised, one standard deviation change in the feature value leads to the odds ratio multiplicative change in the base probability: for the base probability p and the

odds ratio value d, a standard deviation feature value change results in the updated probability of $p \times d$.

3. To assess the model quality, I compute Variance Inflation Factors (VIFs). They allow spotting potentially redundant features in the model. As advised by Andy Field et al. [39], VIFs exceeding 10 indicate the unreliability of the model.

4. Since logistic regression operates under the assumption of linear relations between inputs and outputs, I verify whether it holds by adding $log(f) \times f$ features to the model and checking their significance [39].

5. To make sure there are no influential outliers in the fitted model, I apply the Bonferroni Outlier Test on the fitted model. It allows spotting influential cases significantly changing the behaviour of the model.

Model Performance Since the presence of the event is a minority in most datasets and the majority in the Multiple packages dataset, I compute the mean of PRAUC, treating events as a positive and a negative class. This metric is used across the study. Additionally, I use PRAUC as a performance metric in permutation tests. Lastly, I provide alternative metrics - ROCAUC and F1-score, to allow a better understanding of the results.

Description of the Random Forest and CatBoost pipelines

Feature selection RFECV takes the feature ranking after fitting a subset of features. N worst-ranked features are removed to perform the next iteration of the method. For the LDA feature space, the features were removed one-by-one, hSBM feature space is larger and I drop 10% of the bottom-ranked features to make the experiments computationally feasible. At this point, I use the default estimator parameters, as both RF and CB are known to be rather unpretentious in terms of parameter tuning.

Parameter Tuning To tune the model parameters, I use a Grid Search approach with a 2-folded time series split cross-validation. I optimize the number of trees, maximal tree depth and class weights. For CatBoost I additionally optimize the tree L2-regularisation and use of the temporal dimension (binary variable).

Model Analysis For RF and CB estimators, I generate SHAP values and visualise feature impacts on a per-entry basis using the shap python package implementation [77]. Taking

into account feature values and their impact on a per-entry basis, SHAP provides theory-grounded and more reliable feature importance estimates in comparison to the built-in sklearn and catboost packages feature importance.

Model Performance For CB and RF, I compute the same set of performance metrics as for the LR.

3.1.6 Synthetic data generation

As it was previously mentioned, micro-events are potentially harder to detect than conventional events. Moreover, to the best of my knowledge, there is no dataset offering ground truth for micro-event detection. In the current chapter I propose a way to design a micro-event dataset. However, in the proposed dataset, it is not feasible to quantify the impact of factors influencing the detection performance. I assume that one of the major factors impacting the micro-event manifestation strength is the fraction of the community engaged in it. The larger the fraction, the more obvious changes it causes in the data. Being guided by this assumption, I generate a synthetic dataset with the objective of understanding what strength of the reactions in textual communications can be reliably detected by the considered means.

In the current section I first briefly describe the data generation process, then I get into detail in the Deep Learning Text Generator subsection. The sequence of the synthetic data study phases is shown in Figure 3.4.

Please note, for the experiments on synthetic datasets, I could not evaluate the hSBM feature space because the available software implementation of this method was not able to handle the size of our dataset.

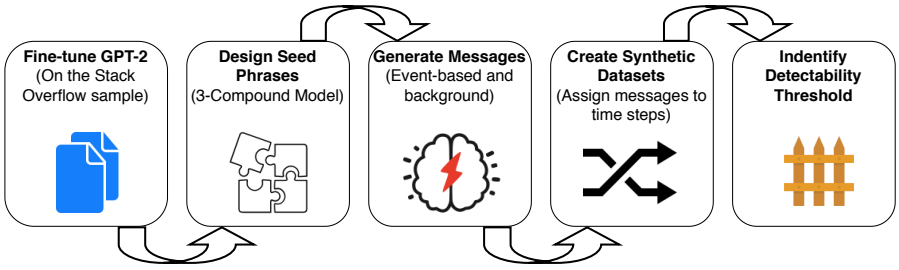


Fig. 3.4 The sequence of steps for the synthetic dataset.

When generating the synthetic data, I separately generate control and event-related messages. After the messages are generated, I form time-steps by bagging messages of two

types in a variable proportion. To assess the model performance, I follow the same feature design procedure as in the case of SO data.

The process of data generation can be seen as a set of steps:

1. Generate background messages;
2. Generate event-related messages;
3. By bagging messages from (1) and (2) I create time-steps with positive labels;
4. I bag messages from (1) to create negatively labelled time-steps.

I assume that in the real world scenario only a fraction of messages are relevant to an event - by changing the proportion of the two message types in step 3) I assess this assumption.

Synthetic Data Analysis Pipeline

After generating the data, I aim to find the detectability threshold for the micro-events. I define the threshold as a minimal fraction of event-related messages, for which micro-events can be detected with a significant outcome of the permutation test. I perform the same feature selection, optimisation, and performance assessment for the SO datasets and synthetic data.

Additionally, to reduce the influence of randomness, I repeat the experiments multiple times for each dataset configuration by reshuffling messages across the time-steps.

Deep Learning Text Generator

The posts are generated using a small version of GPT-2 OpenAI neural network [54] with 117M neurons. The original pre-trained version of GPT-2 is fine-tuned on the multiple packages sample of Stack Overflow data.

Message types The background (or control) messages, having no relation to an event, are generated from an empty context - no seed phrase is provided to the model. To generate event-related messages, I propose an approach to inject events into the data. In the proposed approach the community values are represented by 3 compounds: rules, people and products (Figure 3.5). Changes in any of these compounds might lead to changes in community communications. These are abstract entities, specific to every community.

I generate event-related messages from a set of seed phrases, used as a context for the generator network.

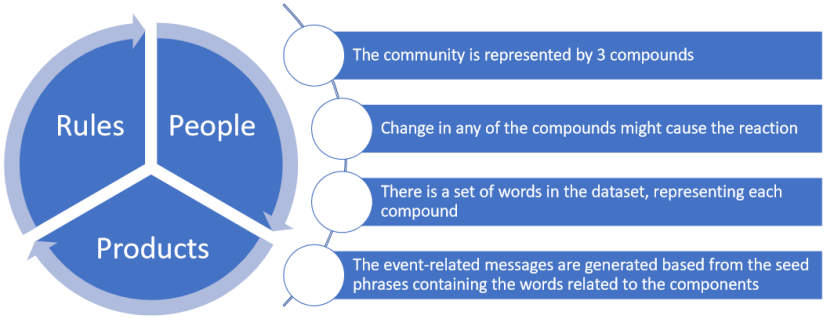


Fig. 3.5 Three compound model is used for the generation of the event-related messages. It assumes that the textual representation of the community in the forum-like platforms consists of 3 compounds and any change in them (event) might lead to the reaction. This model is used to generate event-related messages.

The seed phrases are designed on the basis of the real-world dataset, used for the generator fine-tuning. Entities representing the three compounds (rules, people and products) have to be identified in the Stack Overflow dataset. For that purpose, I use the Word2Vec model [155], fitted on the Google News dataset. I expect this model to be generic and cover a wide range of topics including IT.

I obtain the closest (cosine similarity) 1.7k nouns for each of the components from the SO dataset. Since the events are represented by changes in the compounds, I also take 1k closest verbs representing addition and removal operations.

Finally, I obtain the seed phrases for generating event-related messages by concatenating the selected nouns and verbs. For the generation, I use a random set of 100 seed phrases. I use the NLTK POS tagger for the Part of Speech identification.

The proposed approach is the first step towards synthetic data generation for micro-event detection tasks, hence there is a lot to understand. At this stage, I expect that the proposed synthetic data generation method is generalisable to other textual communications datasets. However, it cannot be applied directly to time series data. I provide a detailed description of the generator optimisation and ways of validating the synthetic data quality in Synthetic data generator optimisation.

3.2 Results

3.2.1 Data sample

Stack Overflow data

In the single package experiments, I focused on Django and Selenium, as they account for around 85% of the messages. Table 3.1 shows the packages (and the number of associated messages) included in the study sample. Moreover, Figure 3.6 represents chronologically the events associated with these seven packages. Black strips at the top part of the figure represent the events. I provide this figure to support the reproducibility of the study.

Table 3.1 Number of posts per package after filtering by the package name in the post's tag or body. The total number of posts in the Stack Overflow platform data dump of 06/2018 is **65049182**.

Package	Number of posts
Django	475760
Selenium	210498
Sentry	60874
Django-rest-framework	24852
celery	20864
Hypothesis	19401
Gunicorn	14602
Total (unique)	**826851 (777812)**

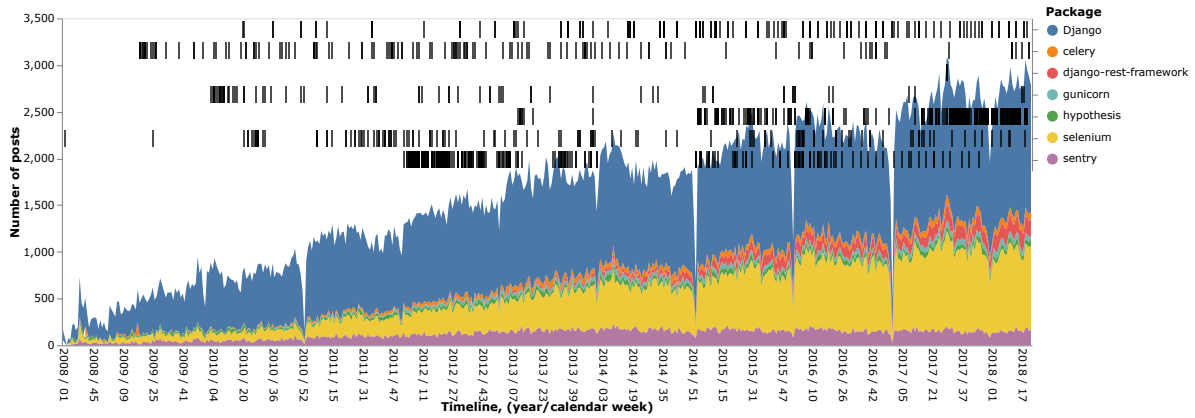


Fig. 3.6 The studied sample of the posts and the events per package for the available time range. The number of posts is provided in a per-week fashion. The packages are stacked vertically. The spike drops take place in the New Year's Eve periods.

The posts before 27 July 2015 belong to the training set, after the date - to test.

Synthetic Data

The optimised generator was used to obtain 1109150 background messages and 154680 event-related messages.

I generated 15 instances of each dataset configuration with event-related message fractions in the range from 0.1 to 0.45 with a step of 0.05. Each synthetic dataset consists of 335 time-steps, to match the number of time steps in the SO dataset, ranging from 171 to 708 time-steps. The ratio of positive to negative time steps was set to 0.25 to mirror the considered real-world scenarios (the mean event fraction in the SO datasets is 0.26).

The details on how the messages and the time steps were generated are provided in the Synthetic data generation section.

3.2.2 SO datasets results

Summary of the results

I list the performance summary of the three estimators (Table 3.2) obtained by fitting and tuning on the training dataset batch and assessing performance on the test batch of the dataset. I provide more metrics in Performance of estimators. Feature selection leads to larger feature spaces in the case of CB and RF estimators - this is observed for the hBSM feature space (Table 2 of Performance of estimators), it is less obvious from LDA feature space results (Table 1 of Performance of estimators). Once the corrections for multiple comparisons are applied within each experiment family, the permutation tests reveal 7 significant models in total. Interestingly, 6 of them belong to the multiple packages dataset. Considering the PR-AUC metric, one sees that the models' performance in most cases are only marginally better than the baseline of 0.5.

To further analyse the results and understand the limitations of the models, in this section, I focus on a particular dataset as a case study: Selenium package, patch updates, event-based time steps dataset. I chose this dataset based on the goodness of fit of the LDA features space. Concretely: I assessed Adjusted McFadden and Tjur R^2 measures (see The goodness of fit measures of Logistic Regression models). I did not consider multiple packages datasets with event-based time steps due to their violation of the independence assumption of the logistic regression - a message may be included in multiple time steps simultaneously, as described in section 3.1.3. Also, when considering the goodness of fit, I restrict myself to LDA feature space due to the artificial behaviour of some metrics caused by the large dimensionality of hSBM feature space.

Table 3.2 CatBoost, Random Forest and Logistic Regression model performance is reported. I evaluate the three estimators on all the datasets. PR-AUC and f1-score metrics are computed as averages of events being positive and a negative class. P.test columns contain p-values of the permutation tests over 1000 permutations. After applying Holm-Bonferroni corrections the threshold for the top 1 model is 0.0014. The significant entries are marked with a star (*). More metrics are reported in Performance of estimators.

	hSBM feature space								
Dataset	**CatBoost**			**RF**			**LR**		
	PRAUC	P.test	F1	PRAUC	P.test	F1	PRAUC	P.test	F1
Multiple major event-based	0.54	1.00	0.16	0.52	1.00	0.16	0.52	0.76	0.20
Multiple minor event-based	0.50	0.01	0.51	0.75	<0.001*	0.49	0.50	0.90	0.36
Multiple patch event-based	0.50	0.81	0.35	0.50	0.88	0.41	0.51	<0.001*	0.49
Django minor event-based	0.50	0.26	0.50	0.55	<0.001*	0.47	0.51	0.01	0.41
Django patch event-based	0.52	1.00	0.46	0.54	1.00	0.47	0.51	0.15	0.42
Selenium minor event-based	0.52	1.00	0.46	0.50	1.00	0.46	0.50	1.00	0.46
Selenium patch event-based	0.50	1.00	0.44	0.51	1.00	0.44	0.50	1.00	0.44
Multiple major c.w.-based	0.50	1.00	0.46	0.50	1.00	0.46	0.51	1.00	0.47
Multiple minor c.w.-based	0.50	0.03	0.48	0.51	0.29	0.45	0.51	0.01	0.53
Multiple patch c.w.-based	0.51	0.14	0.51	0.51	<0.001*	0.40	0.51	0.02	0.46
Django minor c.w.-based	0.51	0.04	0.46	0.51	1.00	0.45	0.50	0.54	0.45
Django patch c.w.-based	0.50	1.00	0.48	0.50	1.00	0.48	0.50	1.00	0.48
Selenium minor c.w.-based	0.51	1.00	0.47	0.56	0.07	0.35	0.51	0.10	0.47
Selenium patch c.w.-based	0.52	0.44	0.45	0.52	1.00	0.47	0.50	0.08	0.54
	LDA feature space								
Dataset	**CatBoost**			**RF**			**LR**		
	PRAUC	P.test	F1	PRAUC	P.test	F1	PRAUC	P.test	F1
Multiple major event-based	0.56	0.07	0.58	0.57	0.06	0.58	0.40	0.88	0.49
Multiple minor event-based	0.51	<0.001*	0.45	0.59	0.19	0.52	0.45	0.93	0.45
Multiple patch event-based	0.51	<0.001*	0.46	0.70	0.01	0.51	0.50	0.98	0.46
Django minor event-based	0.51	0.50	0.46	0.53	0.02	0.44	0.51	0.36	0.43
Django patch event-based	0.54	0.15	0.55	0.50	0.30	0.51	0.47	0.44	0.47
Selenium minor event-based	0.51	1.00	0.45	0.51	1.00	0.47	0.46	0.98	0.47
Selenium patch event-based	0.50	0.81	0.46	0.50	0.03	0.46	0.52	0.46	0.45
Multiple major c.w.-based	0.53	0.18	0.54	0.50	0.23	0.50	0.48	0.80	0.48
Multiple minor c.w.-based	0.51	<0.001*	0.37	0.51	0.22	0.50	0.54	0.19	0.52
Multiple patch c.w.-based	0.50	0.40	0.52	0.51	0.49	0.46	0.46	0.89	0.32
Django minor c.w.-based	0.50	1.00	0.44	0.50	1.00	0.44	0.50	0.52	0.44
Django patch c.w.-based	0.50	0.21	0.53	0.51	1.00	0.44	0.48	0.86	0.47
Selenium minor c.w.-based	0.50	1.00	0.45	0.51	1.00	0.48	0.48	0.72	0.48
Selenium patch c.w.-based	0.51	1.00	0.46	0.51	1.00	0.46	0.52	0.21	0.47

Effect sizes As a model-agnostic feature analysis, I build a forest plot of the effect sizes, sorted by the range of the confidence intervals (CIs) (Figure 3.7).

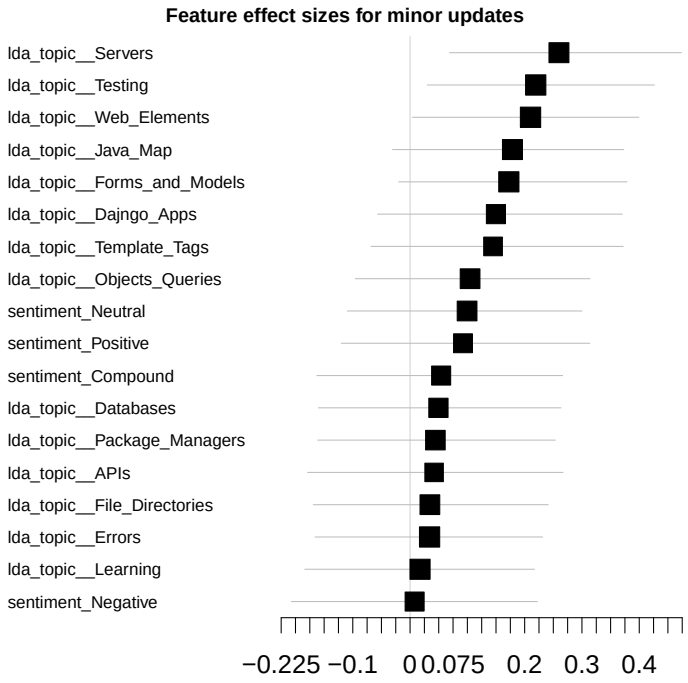


Fig. 3.7 Effect sizes computed for the Selenium package, minor updates, event-based time steps dataset, LDA feature space. Cliff's Delta is used as the effect size measure to account for the non-normal distribution of the feature values. The error bars account for 0.95 confidence intervals (CIs). Features, whose lower CI bound is greater than 0 are considered to be statistically significant. Interpreting the CIs - there is a 0.95 probability that the effect size computed on the population is in the bounds of the CI.

The error bars illustrate the 0.95 CIs allowing assessment of the significance of the features beyond the considered sample. There are 18 features in total - 14 LDA topics and 4 sentiment-related ones. At this point, there are only 3 features with significant effect size - Servers, Testing and Web Elements LDA topics.

I do not report the effect sizes of hSBM feature space due to its dimensionality leading to large CIs and, consequently, the insignificance of all the features.

Logistic regression analysis

Model fitting analysis I have fitted the logistic regression model (Tables 3.3 and 3.4) and computed their goodness of fit measures. After the feature selection step, there are 13 features in the case of LDA feature space. After correcting for multiple comparisons, 5 of them are significant - Intercept, Template Tags, Testing, Forms and Models and Servers LDA topics. Based on Tjur R^2 one can see that around 16% of the variance is explained, other R^2 measures support that statement. It should be noted that pseudo R^2 typical values tend to be

smaller than linear regression R^2. Concretely: McFadden suggests that McFadden pseudo-R^2 values of 0.2-0.4 correspond to an excellent model fit [156]. Overall, the pseudo R^2 values do not show any anomalies. Correcting for multiple comparisons, the Log-Likelihood test outcome indicates that the model fit is significant. hSBM feature spaces model contains only two features including the intercept. Even though the topic feature is significant, R^2 and LLR test metrics indicate a poor model quality and close to no variance explained (Table 3.4).

Table 3.3 LR model fit with its assessment of the goodness of fit. The model was fitted on Selenium package, event-based time steps, minor updates dataset, LDA feature space, with the update events as a dependent variable. The subset of features was selected using the RFECV method with a step of 1. After applying Holm-Bonferroni corrections, significant p-values are marked with a star (*).

Predictor	**Estimate**	**Std. Error**	**Z-value**	**Pr(> |z|)**	**VIF**
(Intercept)	-1.68	0.19	-8.94	< 0.001*	
lda_topic__Web_Elements	0.44	0.33	1.34	0.179	3.93
lda_topic__Package_Managers	0.28	0.18	1.58	0.114	1.15
lda_topic__Template_Tags	0.58	0.18	3.13	0.002*	1.35
lda_topic__Testing	1.14	0.36	3.19	0.001*	5.37
lda_topic__Dajngo_Apps	0.38	0.18	2.17	0.03	1.22
lda_topic__Errors	0.41	0.22	1.82	0.068	1.75
lda_topic__Forms_and_Models	0.57	0.18	3.11	0.002*	1.24
lda_topic__Servers	0.57	0.19	3.07	0.002*	1.44
lda_topic__File_Directories	0.31	0.19	1.65	0.099	1.27
lda_topic__Objects_Queries	0.36	0.18	1.96	0.05	1.18
sentiment_Positive	0.58	0.24	2.42	0.015	2.03
sentiment_Compound	-0.49	0.25	-1.98	0.048	2.19
Fit Measurements					
LLR Test Chi^2	50.8	Observations		329	
Log Likelihood	-147	Null model Log Likelihood		-173	
LLR Test p-value	<.001	Degrees of freedom		13	
AIC	321	Adj. McFadden R^2		0.07	
Null model base probability	0.22	Cox-Snell R^2		0.14	
		Nagelkerke R^2		0.22	
		Tjur R^2		0.16	

I compute the odds ratios of the models (Figs 3.8 and 3.9) to assess per-feature contributions to the model output. The error bars correspond to 0.95 CIs. Since the data is standardised, one standard deviation change in the feature value leads to the event probability change multiplicative of the feature's odds ratio. Significant features are ones whose CIs do not cross a vertical line at $X = 1.0$.

Table 3.4 LR model fit with its assessment of the fit quality. The model was fitted on Selenium package, event-based time steps, minor updates dataset, hSBM feature space, with the update events as a dependent variable. The subset of features was selected using the RFECV method, where 10% of features were removed per iteration. After applying Holm-Bonferroni corrections, the significant p-values are marked with a star (*).

Predictor	**Estimate**	**Std. Error**	**Z-value**	**Pr(> |z|)**	**VIF**
(Intercept)	-1.29	0.13	-9.54	< .001*	-
hsbm_topic_310	0.24	0.12	1.98	0.048*	-
Fit Measurements					
LLR Test Chi^2	3.92	Observations		329	
Log Likelihood	-172.87	Null model Log Likelihood		-171	
LLR Test p-value	0.048	Degrees of freedom		1	
AIC	346	Adj. McFadden R^2		0.0	
Null model base probability	0.22	Cox-Snell R^2		0.01	
		Nagelkerke R^2		0.02	
		Tjur R^2		0.01	

Let us interpret the feature impact on the model output on the example of the LDA feature space. If considering significant features with odds ratios around 2 and the model base probability equals 0.22, one standard deviation increment in any of the features leads to the event probability increasing twice - to 0.44. Due to the intercept odds ratios being far from neutrality (1.0), the positive model output might require multiple features activated.

Validating model assumptions Logistic regression algorithm assumes there is a linear relation between features and the output. I assess that assumption by extending the feature space with interaction terms and checking their significance. For the considered dataset configuration, LDA feature space, I find that Errors LDA topic ($F \times log(F)$) is significant - the linearity criterion is violated. I observe this for the hSBM feature space as well - the non-linear feature of topic 310 is also significant. I conclude that there is a non-linearity present in both cases and they cannot be detected using the LR model. RF and CB do not have any constraints on the data linearity and are capable of detecting the effects missed by the LR.

None of the Variance Inflation Factors (VIFs) crosses the threshold of 10 (Table 3.3), meaning there are no superfluous features in the model. Since there is only one topic feature in the hSBM case, no VIF is computed. Bonferroni Outlier Test did not discover any significant outliers, meaning that there are no influential cases in the training data.

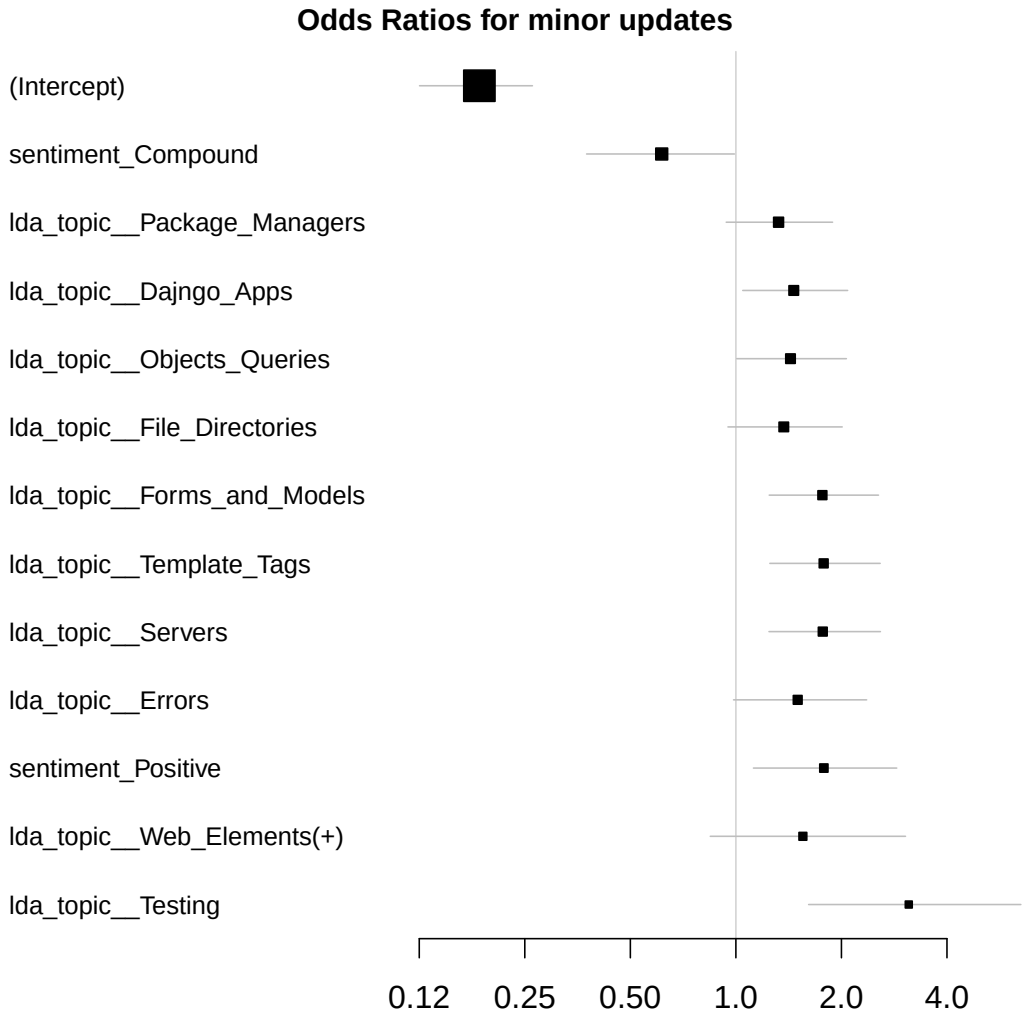


Fig. 3.8 Odds Ratios computed for the model fitted on Selenium package, minor update events, event-based dataset, LDA feature space. The X-axis is logarithmic and the features are sorted ascendingly by the confidence interval range.

Random Forest and CatBoost analysis

I tuned model parameters using a grid search approach with time series cross-validation and report the optimised parameters in Table 3.5.

I plot SHAP values with respect to the positive class output for RF and CB, both feature spaces in Figs 3.10-3.13.

The optimised LDA feature space consists of 6 and 4 features for RF and CB, respectively. More often smaller feature values contribute towards a negative class output. Furthermore, there are 3 common features in the models. Moreover, they affect the output in the same

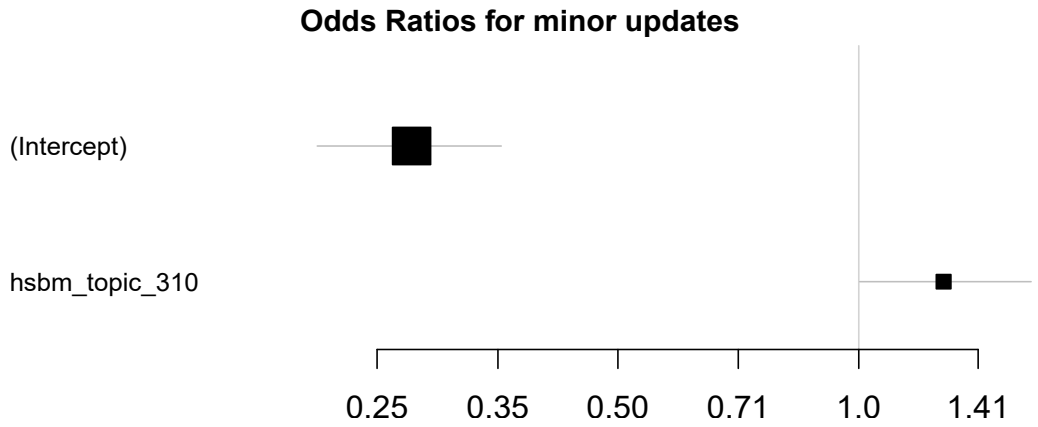


Fig. 3.9 Odds Ratios computed for the model fitted on Selenium package, minor update events, event-based dataset, hSBM feature space. The X-axis is logarithmic and the features are sorted ascendingly by the confidence interval range.

Table 3.5 The table shows the final parameters of Random Forest and CatBoost estimators for the Selenium package, minor updates, event-based time step dataset. "s-balanced" value accounts for "subsample-balanced" type of class label balancing. Depth corresponds to the maximum depth of a decision tree. Temporal nature is a binary variable, indicating whether the temporal nature of data should be considered when fitting the model. The last two parameters were optimised only for CatBoost.

	hSBM feature space		**LDA feature space**	
	Random Forest	CatBoost	Random Forest	CatBoost
Depth	8	6	8	6
Trees number	200	200	50	500
Class balance	s-balanced	balanced	balanced	balanced
L2 Regularisation	-	3	-	7
Temporal nature	-	True	-	True

way. For most of the features, the maximum impact towards a negative output is stronger than the maximum impact towards a positive - this is shown by larger absolute SHAP values to the left from zero value (X-axis). This leads to a more probable negative class output of the model (agrees with the LR model). When I obtain a confusion matrix, the majority class is predicted for all entries in both models. From the odds ratios of LR and SHAP values, one can see that the Testing topic has the largest impact on the model output from all the LDA topics for all three estimators. The nature of the impact is similar as well - larger feature values contribute towards a positive output. Forms and Models topic (which is significant in the LR model) is also present in all the models. Its impact is shifted for CB and RF - there are a

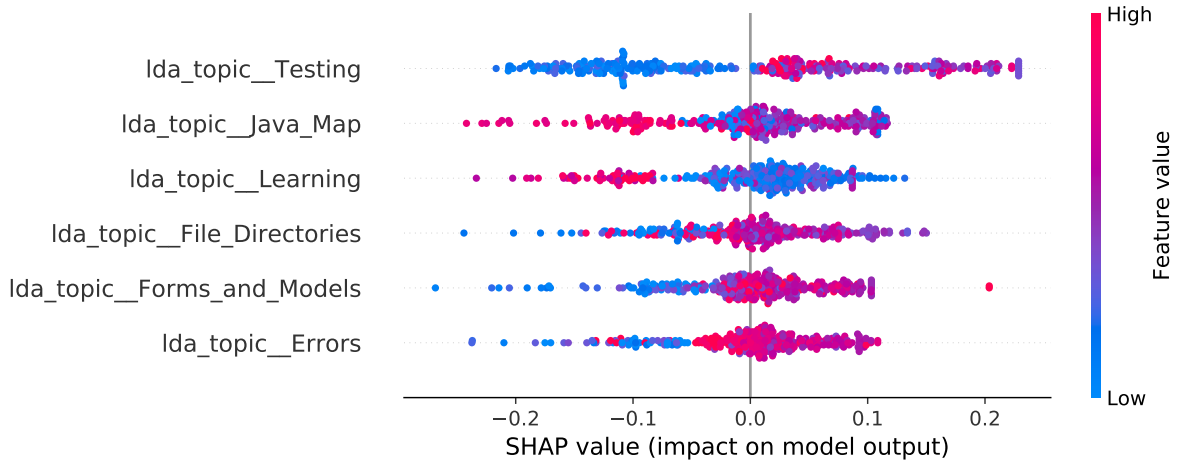


Fig. 3.10 The figure illustrates the influence of the features on the output of the Random Forest model fitted on the Selenium package, minor updates, event-based time steps dataset. Colour encodes the feature value and the X-axis represents the impact of the feature in a particular case. The RF features partially overlap with the LR - there are 4 common features, 2 of which are significant in the LR. There are 3 features overlapping with the CB model.

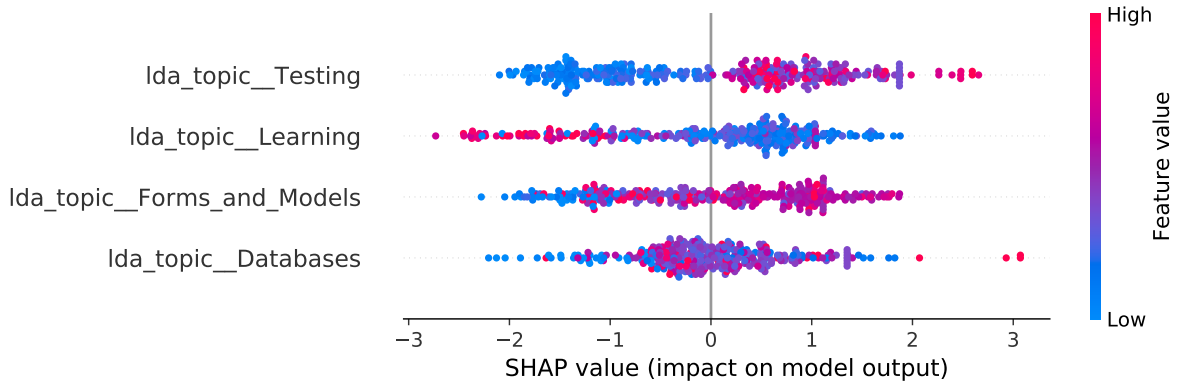


Fig. 3.11 The figure illustrates the influence of the features (SHAP) on the output of the CatBoost model fitted on the Selenium package, minor updates, event-based time steps dataset. Colour encodes the feature value and the X-axis represents the impact of the feature in a particular case. There are two features that overlap with the LR model and 3 features overlapping with the RF model.

number of entries where middle to large values of the feature cause no impact or contribute towards a negative class output.

Considering hSBM feature space, the optimised feature subsets are large, hence in Figs 3.12 and 3.13, we show only the top 10 features for each model. Due to a large number of hSBM topics, it is not feasible to interpret all of them. Consequently, later in the text,

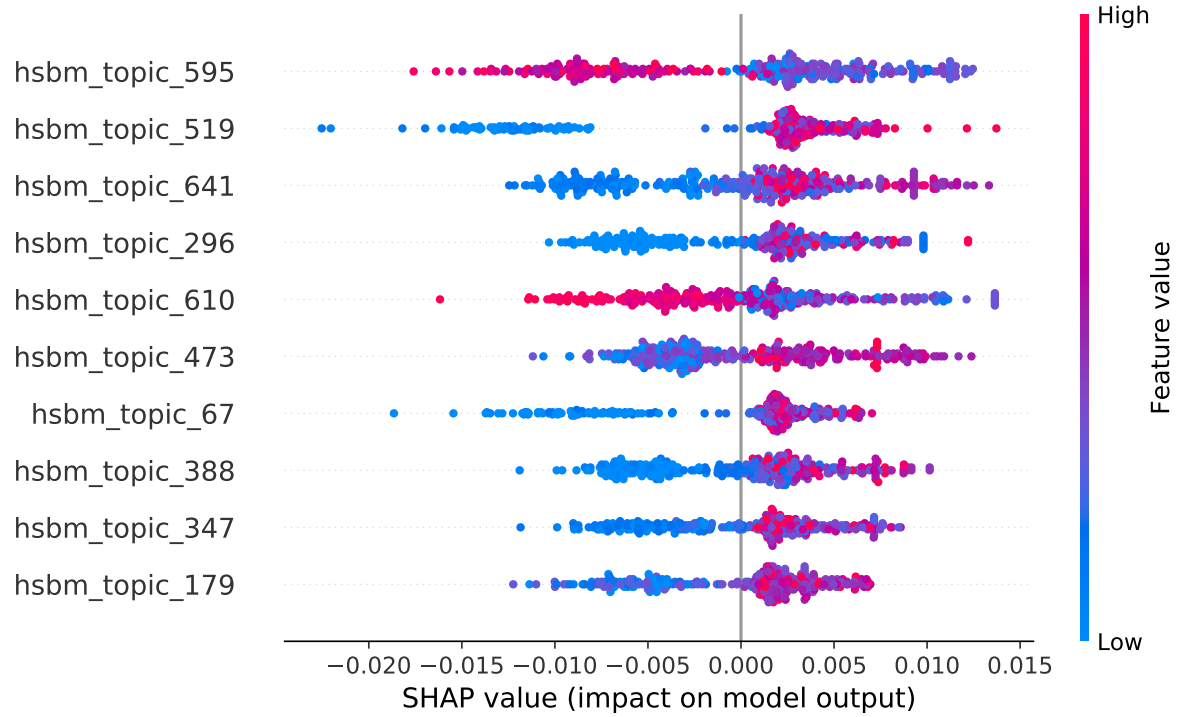


Fig. 3.12 The figure illustrates the influence of the features (SHAP) on the output of the Random Forest model fitted on the Selenium package, minor updates, event-based time steps dataset. Colour encodes the feature value and the X-axis represents the impact of the feature in a particular case. Only the top 10 features from the feature space are shown.

I interpret the top 3 impactful features for each model. Overall, there are 441 features overlapping between RF and CB models.

LDA topics interpretation

LDA was optimised on the training dataset and 14 topics setting has the optimal coherence C_V score based on the Elbow heuristic, as shown in Figure 3.14.

Below I interpret significant features of the LR model and all the features of RF.

File Directories LDA topic - the topic contains messages on the operations with files and file systems, like accessing, storing, and downloading. The top 4 characteristic words extracted from the LDA model are: 'file', 'image', 'directory', 'folder'.

Template Tags LDA topic - communication on Django template language, use of tags in the context of this language. The tokens are 'template', 'view', 'use', 'url'.

Forms and Models topic - posts on Django models and forms. These two notions are related in a way that models provide access to databases and forms are used to input the data into

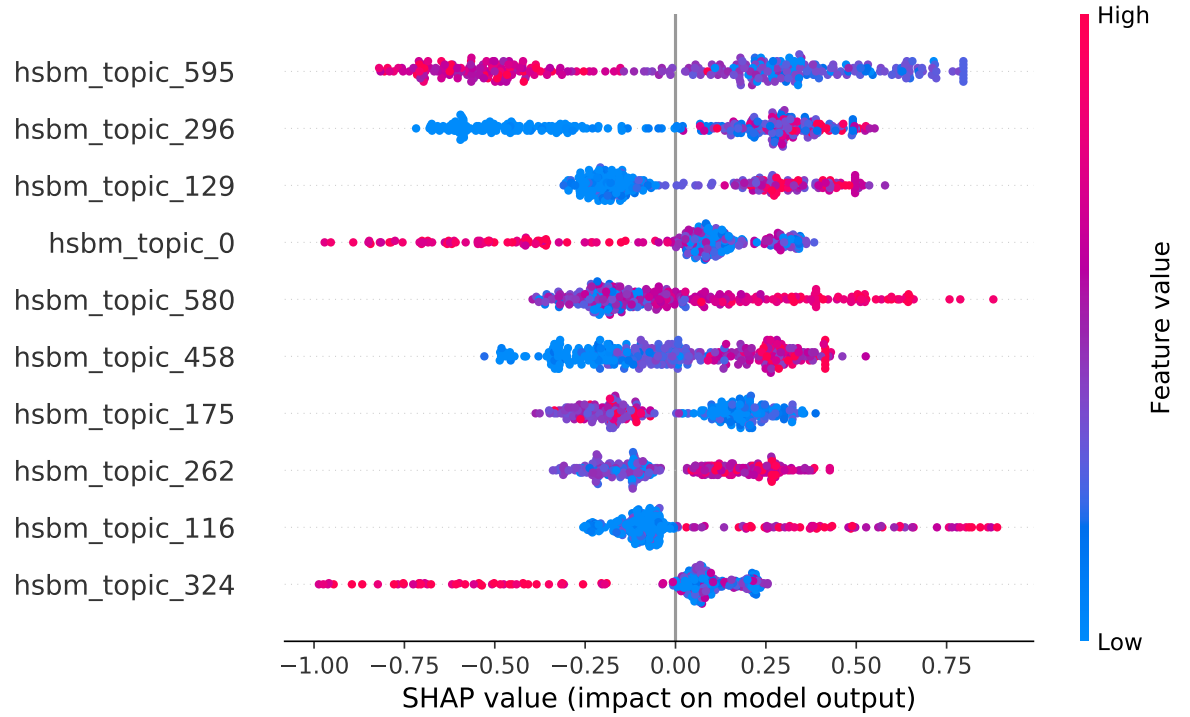


Fig. 3.13 The figure illustrates the influence of the features (SHAP) on the output of the CatBoost model fitted on the Selenium package, minor updates, event-based time steps dataset. Colour encodes the feature value and the X-axis represents the impact of the feature in a particular case. Only the top 10 features from the feature space are shown.

the databases. The characteristic tokens are 'model', 'form', 'field', 'class'.

Java Map topic - communication on Java map structure and its aspects. The characteristic words are 'value', 'list', 'use', 'string'.

Learning topic - posts on coding skills development and IT education. The characteristic tokens are 'use', 'good', 'time', 'would'.

Errors topic's characteristic words are 'try', 'error', 'get', 'work'.

Testing topic's characteristic words are 'test', 'use', 'selenium', 'run'.

Servers topic's characteristic words are 'django', 'app', 'server', 'use'.

I did not expect to see Learning and Java Map topics in the list of selected features of RF, since they seem to be less related to the Selenium package dataset. In this sense, LR model optimisation leads to a more expected feature subset.

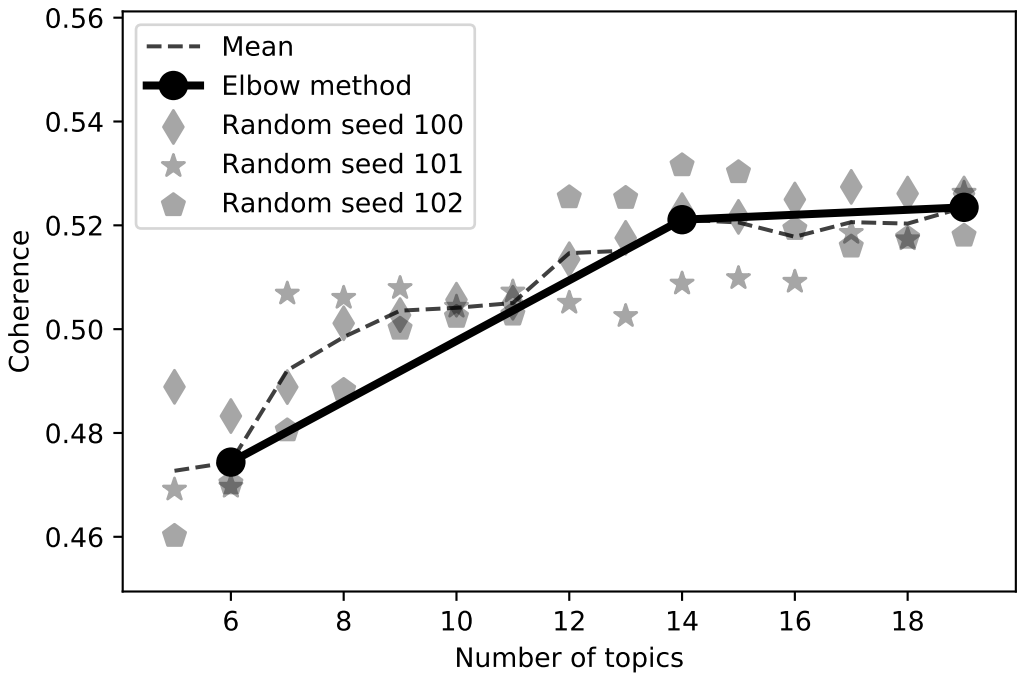


Fig. 3.14 LDA topic number optimisation is visualised. The figure demonstrates coherence measures computed for varying number of topics, for LDA models randomly initialised with 3 random seeds, as well as mean coherence across the models. Elbow heuristic is applied using the mean coherence across the models. As an outcome, 14 LDA topics model is used throughout the study.

hSBM topics interpretation

I interpret the top 3 features of CatBoost and Random Forest models, hSBM feature space. The interpretation is done by providing their characteristic tokens with associated probabilities in Table 3.6.

Table 3.6 The table provides characteristic tokens with associated probabilities for top 3 features of RF and CB models fitted on hSBM feature space. Since topic 595 is common for both models, there are 5 features interpreted in total.

Topic 595		**Topic 296**		**Topic 129**		**Topic 519**		**Topic 641**	
Token	Prob.	Token	Prob.	Token	Prob.	Token	Prob.	Token	Prob.
attribute	0.28	setting	0.98	team	0.28	driver	0.45	run	0.089
item	0.18	settingsi	0.0015	game	0.15	protractor	0.097	test	0.064
insert	0.10	settingsand	0.0011	player	0.11	phantomjs	0.088	com	0.051
parent	0.087	yourproject	0.0010	story	0.11	headless	0.076	server	0.047

One might notice that Topic 641 looks similar to the LDA Testing topic with 2 characteristic tokens overlapping ('test' and 'run'). Additionally, Topic 641 contains a common 'server' token with LDA Servers.

3.2.3 Synthetic data results - response strength analysis

I assess the performance of the three estimators on the synthetic data for a varying fraction of event-related posts. I perform permutation tests for all instances and plot the results as a scatter plot (Figure 3.15). Error bars show 0.95 confidence intervals for the multiple instances of the same configuration. The colour indicates the maximum p-value among the 15 instances of the same configuration - the darker, the smaller the p-value is. One can see that the first significant p-value is at 0.25 fraction of the event-related messages. The significance of fractions 0.05 and 0.01 is reached by CB and RF, respectively.

In terms of the absolute performance, all estimators perform comparably - CIs overlap, consequently there is no significant difference in their performance.

3.3 Discussion

3.3.1 Results interpretation

I found 13 significant fits of LR models based on the Log-likelihood Ratio test after applying the corrections for multiple comparisons. At the same time, I observed 7 models with significant permutation test results in the SO data. The null hypothesis can be rejected -

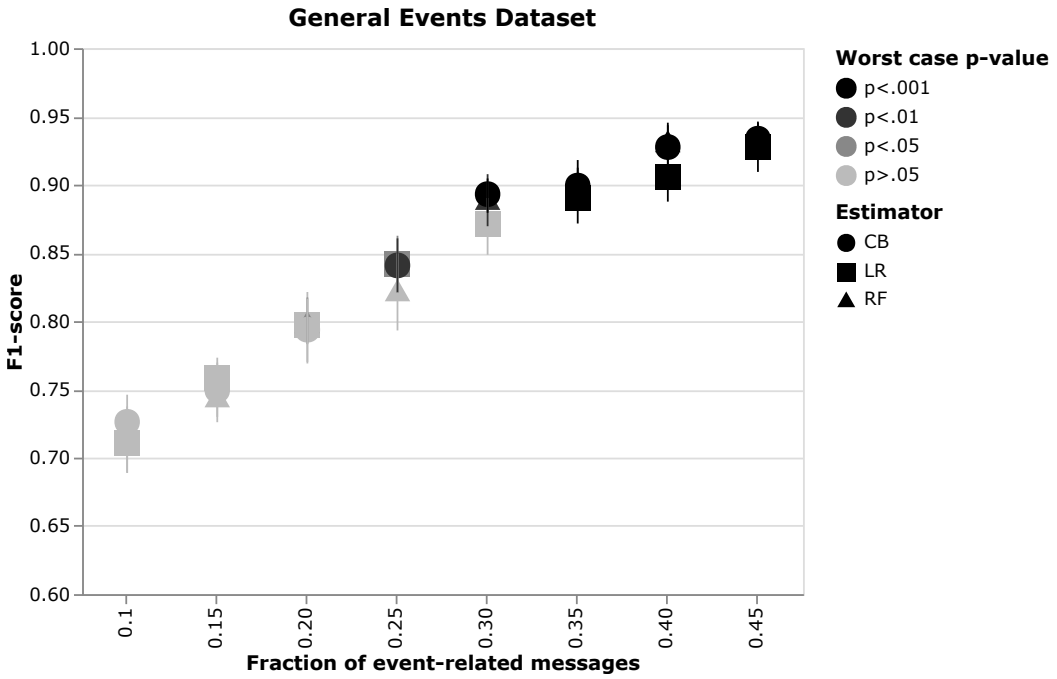


Fig. 3.15 Performance of CatBoost (CB), Random Forest (RF), and Logistic Regression (LR) estimators against the fraction of the event-related messages. The experiments were conducted on a synthetically generated dataset. The performances are means over 15 randomly initialised dataset instances of the same configuration. The illustrated p-values are obtained from the permutation tests (1000 permutations) and are the maxima (as the worst case) over 15 random initialisations. The error bars account for 0.95 confidence intervals computed from the random initialisations.

we have found a statistically significant change in the topics distribution when events take place. This can be already stated on the basis of the significance of the model fit. Moreover, from the assessment of the significance of the predictions, I state that the fitted models generalize to the unseen data. Even though the tests indicate significant results, the absolute model performance is rather weak. I feel that there is a need for model improvement before it is used for solving real-world challenges.

3.3.2 Limitations

Time step design

The event-based time steps might violate the independence assumption for the logistic regression model in case there are multiple updates of the same type taking place within the time interval of a time step. It leads to using some fraction of posts in multiple entries, making the entries dependent. The violation does not allow to rely on the logistic regression

model results in the case of the multiple packages dataset. However, this setting can be used in the machine learning approach with its own benefits, such as more data points and a narrower defined setting in comparison to the calendar week-based time steps.

The proposed time step representation has a low risk of encountering ethical issues because it does not require labelling individual messages and hence it can better preserve the privacy of individual message contributors.

Response lag

In this work, I assume that the reaction can be observed within up to 7 days after the event took place. This assumption was empirically derived from the sparsity of the events-posts as well as my understanding of the software engineering workflows. Moreover, when designing reference time steps, one has to assume that the reaction to the most recent event is not in the data any longer. For the event-based time steps, I assume that day 8 after the event would belong to the reference time step. The assumption becomes unrealistically strict when considering daily time steps. Depending on the available length of the time series, the nature of events, and the activity of the community, the time step window can be optimised.

It should be noted that the two suggested time step designs are aimed at two different temporal natures of the reactions. The event-based design is aimed at posterior reactions to events, and the calendar week based is aimed at both prior and posterior. It should be taken into account if comparing the results.

NLP tools

I am aware of the limitations of the sentiment analysis tools [157, 158], as well as part-of-speech recognition tools [159] applied to the SE domain. Indeed, it makes SE texts challenging for the NLP tasks. I used a classical NLTK Python package POS tagger, and NLTK Vader sentiment analysis over other potentially more powerful methods like deep learning attention-based [160] and manual labelling-based [161] to ensure the best generality possible. The deep learning approach is computationally intensive, especially when applied to large datasets. The manually labelled dataset is available for SE texts, however, there is no guarantee that there is a similar tool in other domains. Concluding: there is definitely space to extend the demonstrated approach, which I discuss below.

When generating event-related synthetic messages, I assume that the three community components are represented by nouns and the change operations - by verbs. Depending on the community and its language, it might not hold.

Generator fine-tuning

To fine-tune the generator, I use multiple packages post sample, which contains both event-related and not related entries. To generate event-related and background messages separately, I use the seed phrases, designed based on the 3-Compound model. This way I overcome the requirement of manual labelling of messages, making the approach scalable and applicable to the content where possibilities for manual labelling are limited.

Design decisions

Multiple design decisions were made throughout the study. Ideally, all the applied thresholds should be further optimised to find an optimal configuration and evaluate their influence on the final result. For instance, the 1% threshold for the inclusion of package-related messages into the dataset might be suboptimal. However, its optimisation would require an additional correction for multiple comparisons. At the same time, this decision does not affect the single-package dataset configurations. I have done the optimisation where it was computationally and statistically feasible.

The number of LDA topics in the above-mentioned papers by Barua et al. [124], Yang et al. [125] and Abellatif et al. [126] is 40, 30 and 12, respectively. This is the same order as obtained in the result of the optimisation in the current work.

3.3.3 Implications for practitioners

In the current state, the approach requires adjustment for applying to real-world problems. Concretely: model performance at this point is unsatisfactory for reliable event detection and might require improvement.

There would be several benefits from the reliable detection of micro-events motivating the continuation of this work, as it would: i. enable software developers to observe the event-related community interactions. ii. On historical data, the developers would be able to make better-informed decisions when choosing dependencies for their projects (i.e. identify problematic dependencies). iii. Finally, it would boost the feedback loop between users and the developers - the event-related interactions can help measure a release's success.

More broadly, the approach might benefit other settings, like detection of emerging scams, illegal products, fraud schemes, etc. Generally, any setting with micro-events and linked forum-like textual communications might make use of the current study.

3.3.4 Future work

As it was mentioned above, the pipeline can be improved, for example by adjusting the NLP tools to the domain. Using domain-agnostic tools, I have demonstrated that micro-event detection is possible in the challenging domain of SE. Considering the improvements, there are more advanced extended LDA models, such as Author-Topic LDA, which links authors, topics, documents and words [162], LDA with Genetic Algorithm, acting on multimodal data [163] or LACT acting on the source code [164]. I feel that specifically for SE, source code analysis might contribute towards model performance improvement.

In the study, I have investigated the pooling through average across the messages within the time step. There might exist more perspective approaches, using, for instance, clustering prior to pooling to get more fine-grained statistics. Finally, it might be worth investigating more in detail other time window designs, as well as adding an additional lag between the event and reference time windows.

As an alternative data representation, deep learning unsupervised models can be used. For instance, variable input size autoencoders [165], and context-aware word embeddings from transformer-based deep learning architectures [54].

Finally, the synthetic dataset generation pipeline may be improved by looking into more advanced ways of comparing real-world and synthetic data. Potential directions are the application of alternative distance measures in parallel with the used ones and the use of human intelligence for assessment.

3.4 Chapter conclusion

The contribution of the chapter is twofold, i) a feasibility study of the micro-events detection on the example of FLOSS version releases and ii) micro-event detection methodology that is generalisable to different types of data, as it will be demonstrated in the following chapters. A significant performance of the proposed approach for micro-event detection was demonstrated and null hypothesis was rejected.

I lay out a detailed analysis to understand model decisions. The experiments on synthetic datasets help understand the limitations on the detectability of the studied micro-events. The analysis of the features contributing to the fitted models can help better understand the nature of the predicted events, and contribute insight for the monitoring and management of the FLOSS ecosystem health. Finally, I identify a series of limitations in the message/time step representations, models and data preparation that lead to several potential lines of future work.

I have performed the detection of micro-events in one of the most challenging domains of textual data. In the following chapters I demonstrate that the notion of micro-event and the proposed methodology are generalisable to other data modalities, like financial time series.

Chapter 4

Event Detection in Price Extrema Patterns of Financial Time Series

As the previous chapter showed, detecting micro-events is challenging in textual data. In this chapter, I move to the financial time series, which presents a very different scenario. Events are often straightforward to observe in financial time series, through volatility changes. However, a new challenge emerges: predicting how these events will impact future price behaviour. Hence, I identify two scenarios of price action and perform micro-event detection and classification.

The chapter introduces a way of automatic extraction of micro-events represented by price patterns from the financial time series. Then, I design a classification pipeline predicting the price movement scenario after the event takes place. The micro-events are defined as price extrema with specific properties. Price extrema is a commonly traded pattern. However, to the best of my knowledge, there is no study presenting a pipeline for automated detection, classification, statistical and profitability evaluation. The current chapter addresses this gap by adjusting the previously proposed methodology (Chapter 3) and presenting a deep analysis of the matter supported by statistical tools which allow generalisability to unseen data and comparability to other studies. The background to the experiments is provided in Chapter 2, Section 2.6.

4.1 Material and methods

In the current chapter, I demonstrate how off-the-shelf machine learning methods can be applied to financial markets analysis and trading in particular. Taking into account the non-stationary nature of the financial markets, I narrow down the number of considered

scenarios (and, hence, non-stationarity) by considering only specific subsets of the time series. I propose a price-action-based way of defining the points of interest (micro-events) and performing their classification. Concretely: I identify local price extrema and predict whether the price will reverse (or 'rebound') or continue its movement (also called 'crossing'). For demonstration purposes, I set up a simplistic trading strategy, where I trade the price reversal after a discovered local extremum is reached as shown in Fig. 4.1. I statistically assess the choices of the feature space and feature extraction method. In the simulated trading, I limit the analysis to backtesting and do not perform any live trading. In the Discussion section I address the limitations of such an approach. Also, I share the reproducibility package for the chapter [166, 167].

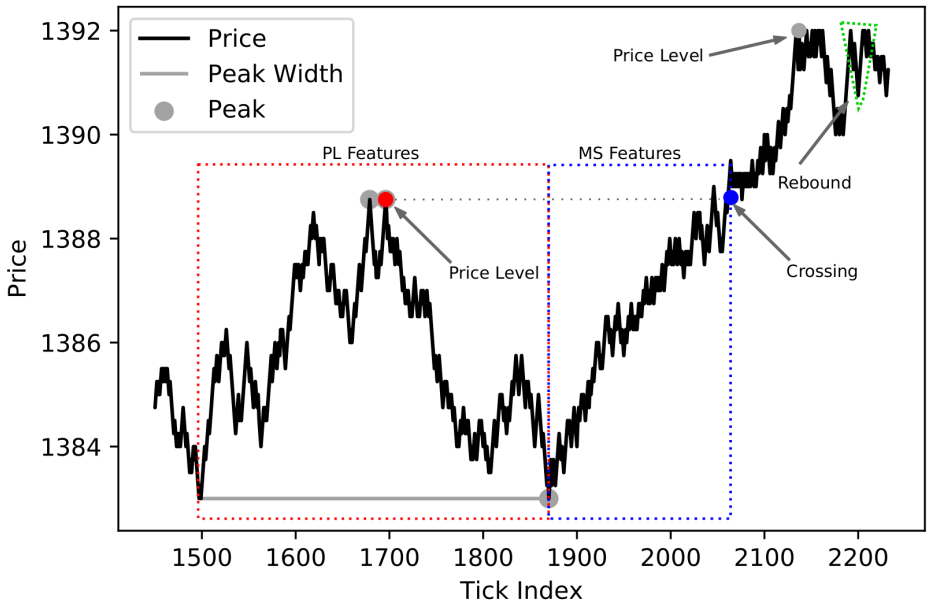


Fig. 4.1 The figure illustrates what data we use for the 2-step feature extraction: price level (PL) and market shift (MS) components. It also demonstrates peaks, peak widths, as well as rebound and crossing labelling.

In the section, I first formally introduce the aim, then describe the datasets and pre-processing procedures. Then, I outline the experiment design comprising of entries labelling and setting up the classification task, designing features, evaluating the model performance, evaluating the results statistically, interpreting model results and, finally, simulated trading.

4.1.1 Research gap & aims

For the current chapter I identify the research gap as twofold: i. lack of studies making use of the existing statistical tools for formal assessment of the findings; and ii. to the best of my

knowledge, there was no attempt to investigate micro-event detection and classification in the context of financial markets.

The aim of the chapter is comprised of multiple aspects: i) demonstrate a way to automatically extract the price patterns ii) propose a two-step way of feature extraction and statistically assess its validity, iii) build a classifier of the events and assess its performance, iv) interpret the classifier decisions, v) propose an automated trading system and check the significance of its performance, iv) demonstrate benefits of the statistical methods in the context of financial time series analysis.

I formulate the research questions and statistical hypotheses which are later evaluated as follows. As the first research question, I aim to assess whether the estimator is capable of fitting to the data given the considered feature space. I believe, it is a necessary initial step, as without this information it would be hard to judge the following findings.
RQ1: Is it feasible to classify the extracted price extrema using the proposed feature space and CatBoost estimator better than the baseline precision?

Assuming that there is value in the proposed data representation, the baseline precision is obtained from an estimator with an always positive class output.

As the second research question, I aim to investigate if the proposed 2-step method for feature extraction gives any benefit in comparison to using any of the extraction steps alone.
RQ2: Does the use of the 2-step feature extraction improve the extremum classification performance with respect to the individual steps?

I provide more details and formulate the null and alternative hypotheses in Section 4.1.5.

4.1.2 Data

In the current chapter, I use S&P500 E-mini CME futures contracts ES(H-Z)2017, ES(H-Z)2018 and ES(H-U)2019. Which correspond to ES futures contracts with expiration in March (H), June (M), September (U) and December (Z). I operate on ticks data with associated Time&Sales records statistics, namely: bid and ask volumes and numbers of trades, as well as the largest trade volumes per tick. I consider a tick to incorporate all the market events between the two price changes by a minimum price step. For the considered financial instrument the tick size is $0.25.

4.1.3 Data pre-processing

I sample the contracts data to the active trading periods by considering only the nearest expiring contracts with the conventionally accepted rollover dates. Namely, the samples end on the second Thursday of the expiration month - on that day the active trading is usually

transferred to the following contract. This decision ensures the highest liquidity, and, due to the double-auction nature of the financial markets, stable minimum bid-ask spreads [168].

For the classification, I consider the two simplest scenarios of the price behaviour after it reaches the local extremum - reversal and extremum crossing. When labelling the entries, I require from 4 to 15 ticks price movement as a reversal (or rebound) and only 3 ticks for the extremum crossing. The labelling approach allows to study of a range of configurations of the reversals and investigate how the configurations affect the performance of the models. At the same time, these ranges are well within the boundaries of the intraday price changes.

An essential part of the proposed experiments is the detection of the price extrema. The detection is performed on a sliding window of the streamed ticks with a window size of 500. I capture peaks with widths from 100 to 400 (the peak width is defied in Chapter 2, Section 2.6.4). The selected widths range serves three purposes: i) ensures that we do not consider high-frequency trading (HFT) scenarios which require more modelling assumptions and a different backtesting engine; ii) allows to stay in intraday trading time frames and have a large enough number of trades for analysis; iii) makes the price level feature values comparable across many of the entries.

4.1.4 Experiment design

In the experiment design I make use of the methodology proposed in Chapter 3. Namely, I use it for consistent integration of the machine learning and statistical methods, as well as reporting the results. The reused components are described below and justified. Together, they ensure comparability and reproducibility of the study as well as allow analysing weakly-manifested effects. Since in the current and the following chapter the baseline is defined differently than that in Chapter 3, the logistic regression component of the methodology in omitted.

Classification task

In order to incorporate machine learning into the automated trading system, I design a binary classification task, where the labels correspond to price reversals (positives) and crossings (negatives). Due to the label design, there is more flexibility in take-profit and stop-loss sizes when trading reversals (up to 15 ticks versus 3 ticks) - this explains their positive labelling.[1] The pseudo-code for labelling is provided in Algorithm 2.

[1] The vocabulary is provided in Table 2.1.

Algorithm 2 Price level labelling

Require: Price level (**PL**) exists and is reached;
  if PL is minimum **then**
    if (current price + 3 ticks) $\leq PL$ **then**
      Label as **crossing**;
    else if (current price - [4..15] ticks) $\geq PL$ **then**
      Label as **reversal**;
    else wait;
    end if
  end if
  if PL is maximum **then**
    if (current price - 3 ticks) $\geq PL$ **then**
      Label as **crossing**;
    else if (current price + [4..15] ticks) $\leq PL$ **then**
      Label as **reversal**;
    else wait;
    end if
  end if

Feature design

To perform the extrema classification, I obtain two types of features: i) designed from the price level ticks (called price level (PL) features), and ii) obtained from the ticks right before the extremum is approached (called market shift (MS) features) as I illustrate in Fig. 4.1. I believe (and statistically test it) it is beneficial to perform the two-step collection since the PL features contain properties of the extremum, and the MS features allow to spot any market changes that happened between the time when the extremum was formed and the time it was approached for the second time and traded.

Considering different extrema widths, varying amounts of the data do not allow to use it for classification in its raw format as most of the algorithms take fixed-dimensional vectors as input. I ensure the fixed dimensionality of a classifier input by aggregating per-tick features by price. The aggregation is performed for the price range of 10 ticks below (or above in case of a minimum) the extremum. This price range is flexible - 10 ticks are often not available within ticks associated with the price level (red dashed rectangle in Fig. 4.1) in this case I fill the empty price features with zeros (called padding). I assume that in most cases the further the price from the extremum the less information relevant to the classification it contains. Hence, considering the intraday volatility of ES, I believe that the information beyond 10

ticks from the extremum is unlikely to improve the predictions. If one considers larger time frames (peak widths), this number would need to be increased.

PL features are obtained from per-tick features by grouping by price with the *sum*, *max* or *count* statistics. For instance: if one is considering volumes, it is reasonable to *sum* all the aggressive buyers and sellers before comparing them. Of course, one can also compute *mean* or consider *max* and *min* volumes per tick. If following this line of reasoning, the feature space can be increased to very large dimensions. I empirically choose a feature space described in Tables 4.1 and 4.2 for Price Level and Market Shift components, respectively. Defining the feature space I aim to make the feature selection step computationally feasible. Too large feature space might be also impractical from the optimisation point of view, especially if the features are correlated.

To track the market changes, for the MS feature component I use 237 and 21 ticks and compare statistics obtained from these two periods. Non-round numbers help avoid interference with the majority of manual market participants who use round numbers [6]. I also choose the values to be comparable to our expected trading time frames. No optimisation was made on them. I obtain the MS features being 2 ticks away from the price level to ensure that the modelling does not lead to any time-related artefacts where one cannot physically send the order fast enough to be executed on time.

Model evaluation

After the features are designed and extracted, the classification can be performed. As a classifier, I choose the CatBoost estimator. I feel that CatBoost is a good fit for the task since it is resistant to over-fitting, stable in terms of parameter tuning, efficient and one of the best-performing boosting algorithms [48]. Finally, being based on decision trees, it is capable of correctly processing zero-padded feature values when no data at price is available. Other types of estimators might be comparable in one of the aspects and require much more focus in the other ones. For instance, neural networks might offer better performance but are very demanding in terms of architecture and parameter optimisation.

In this study I use precision as the main scoring function (S):

$$S = \frac{TP}{TP + FP}, \tag{4.1}$$

where TP is the number of true positives and FP is the number of false positives. All the statistical tests are run on the precision scores of the samples. This was chosen as the main metric since by design every FP leads to losses, and false negative (FN) means a lost trading opportunity without any explicit loss. To give a more comprehensive view of the model

Table 4.1 Price level feature space component used in the experiments. These features are obtained when the price level is formed. When discussed, features are referred to by the codes in the square brackets at the end of descriptions.

	Equation	Description
Price level (PL) features	$\sum_t^{p=\lvert PL-t\rvert}(V_b+V_a)$	Bid and ask volumes summed across all the ticks for $t \in [0,1,2]$ [PL0]
	$\sum_t^{p=\lvert PL-t\rvert} V_b$	Bid volumes summed across all the ticks for $t \in [0,1,2]$ [PL1]
	$\sum_t^{p=\lvert PL-t\rvert} V_a$	Ask volumes summed across all the ticks for $t \in [0,1,2]$ [PL2]
	$\sum_t^{p=\lvert PL-t\rvert} T_b$	Number of bid trades summed across all the ticks for $t \in [0,1,2]$ [PL3]
	$\sum_t^{p=\lvert PL-t\rvert} T_a$	Number of ask trades summed across all the ticks for $t \in [0,1,2]$ [PL4]
	$M(Tp=\lvert PL-t\rvert)_b$	Maximum bid trade across all the ticks for $t \in [0,1,2]$ [PL5]
	$M(T_{p=\lvert PL-t\rvert})_a$	Maximum ask trade across the ticks for $t \in [0,1,2]$ [PL6]
	$\sum_t^{p=\lvert PL-t\rvert} 1$	Number of ticks at price for $t \in [0,1,2]$ [PL7]
	$\frac{\sum_t^{p=\lvert PL-t\rvert} V_b}{\sum_t^{p=\lvert PL-t\rvert} V_a}$	PL1 divided by PL2, for $t \in [0,1,2]$ [PL8]
	$\frac{\sum_t^{p=\lvert PL-t\rvert} T_b}{\sum_t^{p=\lvert PL-t\rvert} T_a}$	Feature PL3 divided by feature PL4 [PL9]
	$\frac{\sum_t^{p=\lvert PL-t\rvert} M(T)_b}{\sum_t^{p=\lvert PL-t\rvert} M(T)_a}$	Feature PL5 divided by feature PL6 [PL10]
	$\frac{\sum_t^{p=\lvert PL-t\rvert}(V_b+V_a)}{\sum_t^{p=\lvert PL-t\rvert} 1}$	Total volume at price $\lvert PL-t\rvert$ divided by the number of ticks [PL11]
	$\sum_{t=0}^{10}\sum_t^{p=\lvert PL-t\rvert} V_a$	Total Ask Volume [PL12]
	$\sum_{t=0}^{10}\sum_t^{p=\lvert PL-t\rvert} V_b$	Total Bid Volume [PL13]
	$\sum_{t=0}^{10}\sum_t^{p=\lvert PL-t\rvert} T_a$	Total Ask Trades [PL14]
	$\sum_{t=0}^{10}\sum_t^{p=\lvert PL-t\rvert} T_b$	Total Bid Trades [PL15]
	$\sum_{t=0}^{10}\sum_t^{p=\lvert PL-t\rvert}(V_a+V_b)$	Total Volume [PL16]
	-	Peak extremum - minimum or maximum [PL17]
	-	Peak width in ticks described in the Background section [PL18]
	-	Peak prominence - described in the Background section [PL19]
	-	Peak width height - described in the Background section [PL20]
	$\frac{\sum_{t\in[0,1,2]}^{p=\lvert PL-t\rvert} V_b}{\sum_{t\in[3..9]}^{p=\lvert PL-t\rvert} V_b}$	Bid volumes close to extremum divided by ones which are further [PL21]
	$\frac{\sum_{t\in[0,1,2]}^{p=\lvert PL-t\rvert} V_a}{\sum_{t\in[3..9]}^{p=\lvert PL-t\rvert} V_a}$	Ask volumes close to extremum divided by ones which are further [PL22]
	$\frac{\sum_{t\in[0,1,2]}^{p=\lvert PL-t\rvert} V_b}{\sum_{t\in[0,1,2]}^{p=\lvert PL-t\rvert} V_a}$	Sum bid volume close to the price extremum divided by the close ask volume [PL23]
	$\frac{\sum_{t\in[3..9]}^{p=\lvert PL-t\rvert} V_b}{\sum_{t\in[3..9]}^{p=\lvert PL-t\rvert} V_a}$	Sum bid volume far from the price extremum divided by the far ask volume [PL24]

Key					
	OB - order book	T - trades	t - ticks	N - total ticks	p - price
	w - tick window	PL - extremum price	V - volume	b - bid	a - ask
	P_N - price level neighbours until distance		M(X) - Max value in set X		

Table 4.2 Market shift feature space component used in the experiments. These features are obtained right before the already formed price level is approached. When discussed, features are referred to by the codes in the square brackets at the end of descriptions.

	Equation	Description
Market Shift (MS) features	$\frac{\sum_{t,b}^{w=237}(V_b)}{\sum_{t,a}^{w=237}(V_a)}$	Fraction of bid over ask volume for last 237 ticks [MS0]
	$\frac{\sum_{t,b}^{w=237}(T_b)}{\sum_{t,a}^{w=237}(T_a)}$	Fraction of bid over ask trades for last 237 ticks [MS1]
	$\frac{\sum_{t}^{w=237} V_b}{\sum_{t}^{w=237} V_a} - \frac{\sum_{t}^{w=21} V_b}{\sum_{t}^{w=21} V_a}$	Fraction of bid/ask volumes for long minus short periods [MS2]
	$\frac{\sum_{t}^{w=237} T_b}{\sum_{t}^{w=237} T_a} - \frac{\sum_{t}^{w=21} T_b}{\sum_{t}^{w=21} T_a}$	Fraction of bid/ask trades for long minus short periods [MS3]
	$\frac{\sum_{t}^{w=237} M(T)_b}{\sum_{t}^{w=237} M(T)_a} - \frac{\sum_{t}^{w=21} M(T)_b}{\sum_{t}^{w=21} M(T)_a}$	Max bid trade divided by ask for long periods minus short periods [MS4]
	RSI($p \in S$)	Technical indicator RSI with the stated periods p [$MS5_X$]
	MACD($lp \in S$; $sp = lp/2$)	Technical indicator MACD with the stated long & short periods lp, sp [$MS6_X$]
Key	T - trades N - total ticks w - tick window V - volume b - bid a - ask P_N - price level neighbours until distance M(X) - Max value in set X S - ranges [20,40,80,120,160,200]	

performance, I also report F1 scores, PR-AUC (precision-recall area under the curve) and ROC-AUC (receiver-operating characteristic area under the curve) metrics. I report model performances for the two-step feature extraction approach as well as for each of the feature extraction steps separately.

In order to avoid large bias in the base classifier probability, I introduce balanced class weights into the model. The weights are inversely proportional to the number of entries per class. The contracts for training and testing periods are selected sequentially - training is done on the active trading phase of contract N, testing - on $N+1$, for $N \in [0, B-1]$, where B is the number of contracts considered in the study.

I apply a commonly accepted ML community procedure for input feature selection and model parameter tuning [62]. Firstly, I perform the feature selection step using a Recursive Feature Elimination with cross-validation (RFECV) method. In the current study on each RFECV step, we remove 10% of the least important features. Cross-validation allows robust assessment of how the model performance generalizes to the unseen data. Since I operate on the time series, I use time series splits for cross-validation. For the feature selection, the model parameters are left default, the only configuration I adjust is class labels balancing as our data is imbalanced. Secondly, I optimize the parameters of the model in a grid-search fashion. Even though CatBoost has a very wide range of parameters that can be optimised, I choose the parameters common for tree ensemble models for the sake of feasibility of

the optimisation and leaving the possibility of comparing the optimisation behaviour to the other ensemble algorithms. The following parameters are optimised: 1) *Number of iterations*, 2) *Maximum depth of trees*, 3) *has_time parameter set to True or False*, and 4) *L2 regularisation*. For the parameter optimisation, I use a cross-validation dataset as well. I perform training and cross-validation within a single contract and the backtesting of the strategy on the subsequent one to ensure the relevance of the optimised model. The high-level pipeline flow diagram is provided in Figure 4.2

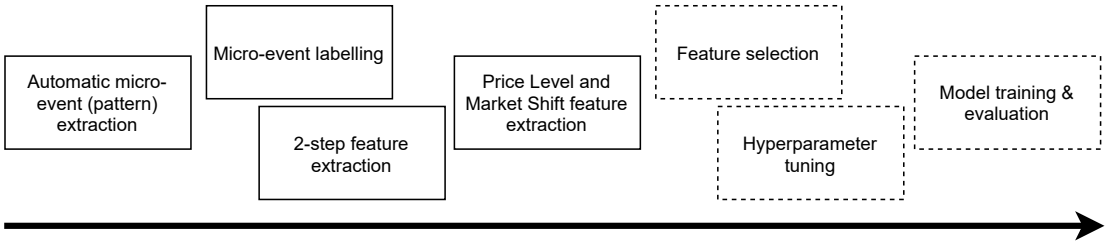


Fig. 4.2 The figure illustrates the flow diagram for the dataset design, data pre-processing and machine learning (ML) model training & evaluation approach is taken in the study. The dashed lines highlight the ML part of the pipeline and solid ones account for pattern extraction and feature engineering. Vertically aligned components may be performed in any order, and the shift indicates their order in the current study.

Statistical evaluation

Here I formalise the research questions by proposing null and alternative hypotheses, suggesting statistical tests for validating them, as well as providing the details on the effect size measure used.

In the current chapter, I report Hedge's g_{av} - an unbiased measure designed for paired data. I correct the confidence intervals for multiple comparisons by applying Bonferroni corrections.

When testing the hypotheses, the samples consist of the test precisions on the considered contracts, leading to equal sample sizes in both groups, and entries are paired as the same underlying data is used. Comparing a small number of paired entries and being unsure about the normality of their distribution, we take a conservative approach and for hypothesis testing use the single-tailed Wilcoxon signed-rank test. This test is a non-parametric paired difference test, which is used as an alternative to a t-test when the data does not fulfil the assumptions required for the parametric statistics. When reporting the test outcomes, I support them with the statistics of the compared groups. Namely, its standard deviations, means and medians.

I set the significance level of the study to $\alpha = .05$. Also, I account for multiple comparisons by applying Bonferroni corrections inside of each experiment family [105]. I consider research questions as separate experiment families.

Model analysis

I perform the model analysis in an exploratory fashion - no research questions and hypotheses are stated in advance. Hence, the outcomes of the analysis might require additional formal statistical assessment. I use SHAP local explanations to understand how models end up with the particular outputs. Through the decision plot visualisations, I aim to find common decision paths across entries as well as informally compare models with small and large numbers of features. The reproducibility package contains the code snippets as well as the trained models, which allow repeating the experiments for all the models and contracts used in the chapter.

Simulated trading

The trading strategy is defined based on our definition of the crossed and rebounded price levels, and schematically illustrated in Fig. 4.3. It is a flat market strategy, where one expects a price reversal from the price level. Backtrader Python package [2] is used for backtesting the strategy. Backtrader does not allow taking bid-ask spreads into account, that is why I am minimising its effects by excluding HFT trading opportunities (by limiting the minimum peak widths) and sticking to the actively traded contracts only. Since ES is a very liquid trading instrument, its bid-ask spreads are usually 1 tick, which however does not always hold during extraordinary market events, scheduled news, session starts and ends. I additionally address the impact of spreads as well as order queues in the Discussion section.

In the performed backtests, I evaluate the performance of the models with different rebound configurations and fixed take-profit parameters, and varying take-profits with a fixed rebound configuration to better understand the impact of both variables on the simulated trading performance.

Since take-profit and stop-loss are fixed within the experiments, it does not make a difference whether using the actual TP&SL values in a regression form or class labels. Trading fees and other market assumptions may vary depending on the volumes, brokerage, country of residence, etc., hence they are not included in the labelling process.

[2] Available at: https://www.backtrader.com/

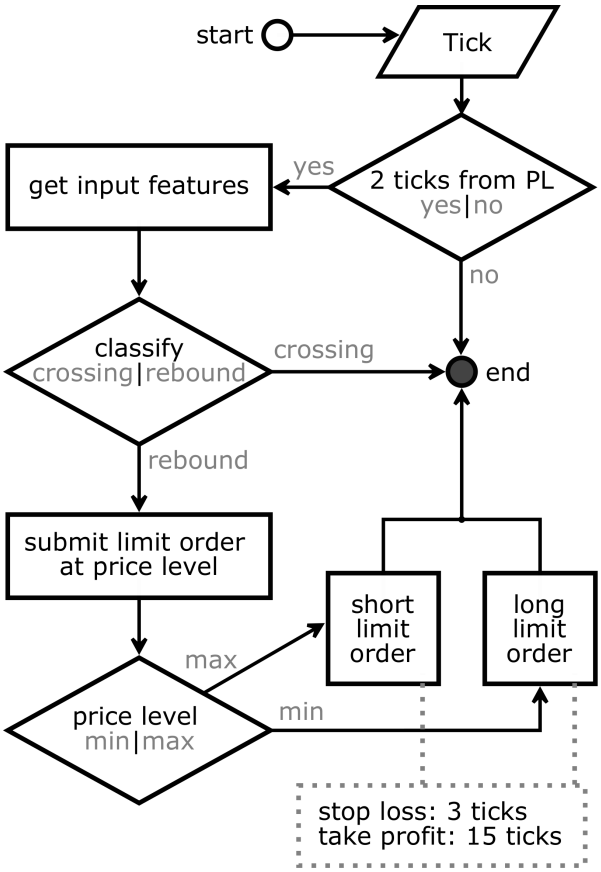


Fig. 4.3 The block diagram illustrates the steps of the trading strategy.

4.1.5 Addressing research questions

In the current subsection, I continue formalising the research questions by proposing the research hypotheses. The hypotheses are aimed to support the findings of the chapter, making them easier to communicate. For both research questions, I run the statistical tests on the precision metric.

RQ1 - CatBoost versus no-information model

In the first research question, I investigate whether it is feasible to improve the baseline performance for the extrema datasets using the chosen feature space and CatBoost classifier. I consider the baseline performance to be the precision of an always positive class output estimator. The statistical test addresses the following hypothesis:

H_{11}: CatBoost estimator allows classification of the extracted extrema with a precision better than the no-information approach.

H_{01}: CatBoost estimator allows classification of the extracted extrema with a worse or equal precision in comparison to the no-information approach.

RQ2 - Two-step versus single-step feature extraction

The second research question assesses if the proposed two-step feature extraction gives any statistically significant positive impact on the extremum classification performance. The statistical test assesses the following hypotheses:

H_{12}: Two-step feature extraction leads to an improved classification precision in comparison to using features extracted from any of the steps on their own.

H_{02}: Two-step feature extraction gives equal or worse classification precision than features extracted from any of the steps on their own.

In the current setting, I compare the target sample (the 2-step approach) to the two control samples (MS and PL components). I am not aiming to formally relate the MS and PL groups, hence only comparisons to the 2-step approach are necessary. In order to reject the null hypothesis, the test outcomes for both MS and PL components should be significant.

4.2 Results

In the current section, I communicate the results of the experiments. Namely: the original dataset and pre-processed data statistics, model performance for all the considered experiment configurations, statistical evaluation of the overall approach and the two-step feature extraction, and, finally, simulated trading and model analysis. An evaluation of these results is presented in Sec. 4.3.

4.2.1 Raw and pre-processed data

The considered data sample and processed datasets are provided in Table 4.3. In the table, I report the number of ticks per contract, the number of positive entries as well as the total number of entries for the considered price reversal (rebound) configurations. I perform the whole study on 3 different rebound configurations: 7, 11 and 15 ticks price movement required for the positive labelling. The contracts are sorted by the expiration date from top to bottom ascendingly. The number of ticks changes non-monotonically - while the overall trend is rising, the maximum number of ticks is observed for the ESZ2018 contract. And the largest change is observed between ESZ2017 & ESH2018. As one can see, the numbers of the extracted extrema do not strictly follow the linear relation with the numbers of ticks per

contract. Considering the numbers of positively labelled entries, the numbers decrease for the larger rebound sizes.

Table 4.3 The table communicates numbers of reconstructed ticks per contract, numbers of positive labels, as well as total numbers of entries per contract. 'Reb.' corresponds to the rebound - the required number of ticks for the positive labelling. Rebound columns show numbers of positively labelled entries.

Contract	No. Ticks	Reb. 7	Reb. 11	Reb. 15	Total
ESH2017	1271810	896	804	703	4911
ESM2017	1407792	951	881	756	4181
ESU2017	1243120	858	812	693	3446
ESZ2017	1137427	689	640	518	12317
ESH2018	2946336	2014	1983	1953	11537
ESM2018	2919757	1965	1936	1868	6534
ESU2018	1825417	1271	1226	1095	16331
ESZ2018	3633969	2677	2620	2565	11718
ESH2019	3066530	1990	1949	1883	9743
ESM2019	2591000	1711	1692	1630	10389
ESU2019	2537197	1761	1735	1704	6191

4.2.2 Price Levels

Automatic extraction

The first step of the pipeline is detecting peaks, which is done automatically, the same way as shown in Fig. 4.1. The peaks are marked with grey circles and the associated peak widths are depicted with solid grey lines. In the considered setting some of the peaks are not automatically discovered as not satisfying conditions of the algorithm by having insufficient widths or being not prominent enough (see Chapter 2, Section 2.6.4 for the definitions of both).

Classification of the extrema

For all the considered models I perform feature selection and parameter tuning. The optimisation results are made available as a part of the reproducibility package. The model precision obtained on a per-contract basis for the 2-step feature extraction, PL and MS feature spaces, and no-information model is provided in Table 4.4. The relative changes in the precision across contracts are preserved across the labelling configurations. There is no evidence that certain configuration shows consistently better performance across contracts.

I report the rest of the metrics for the 2-step feature extraction method, PL and MS features in the supplementary data, in Tables B.2, B.2, B.3, respectively.

Table 4.4 Precisions of CatBoost classifier with the 2-step feature extraction ('2-step') and always-positive output classifier ('Null'). As well as Precisions of the Market Shift (MS) and Price Level (PL) feature spaces. The performance is reported for 3 labelling configurations: 7,11 and 15 ticks rebounds. The implications of the obtained performance values in the context of the financial markets is detailed in the Discussion section.

Contract	Rebound 7				Rebound 11				Rebound 15			
	2-step	Null	MS	PL	2-step	Null	MS	PL	2-step	Null	MS	PL
ESH2017	.20	.19	.20	.19	.20	.18	.19	.17	.15	.15	.15	.15
ESM2017	.22	.21	.24	.21	.23	.19	.20	.19	.19	.17	.16	.18
ESU2017	.25	.20	.25	.23	.15	.19	.23	.17	.18	.15	.15	.19
ESZ2017	.17	.16	.18	.16	.17	.16	.16	.16	.19	.16	.16	.15
ESH2017	.19	.17	.19	.17	.18	.17	.18	.18	.17	.16	.17	.16
ESM2018	.21	.19	.22	.19	.21	.19	.21	.20	.18	.17	.18	.17
ESU2018	.17	.16	.17	.15	.16	.16	.17	.16	.16	.16	.16	.15
ESZ2018	.18	.17	.18	.17	.17	.17	.17	.18	.16	.16	.17	.16
ESH2019	.18	.18	.19	.18	.18	.17	.18	.17	.18	.17	.17	.17
ESM2019	.18	.17	.18	.18	.17	.17	.18	.17	.17	.16	.18	.16

RQ1 - Price Levels, CatBoost versus No-information estimator

Below I report the effect sizes with .95 confidence intervals (CIs) (Table 4.5) associated with the research question. Concretely, I use precision as the measurement variable for comparing the no-information model and the CatBoost classifier. There are no configurations showing significant effect sizes. The largest effect size is observed for the 15 tick rebound labelling. Large CIs are observed partially due to the small sample size.

I test the null hypothesis for rejection for the 3 considered configurations. The original data used in the tests are provided in Table 4.4 - '2-step' and 'Null' columns. I report test outcomes in a form of test statistics and p-values in Table 4.5. Additionally, in the same table, the sample statistics are included. I illustrate the performance of the compared groups in the supplementary materials, in Figure B.2. There is no skew in any of the labelling configurations - medians and means do not differ within the groups. I see around 2 times larger standard deviations for the CatBoost model in comparison to the no-information model. The potential reasons and implications are discussed in sections 4.3.2 and 4.3.5, respectively. There are 3 tests run in this experiment family, hence after applying Bonferroni corrections for multiple comparisons, the corrected significance level is $\alpha = .05/3 = .0167$.

RQ2 - Price Levels, 2-step feature extraction versus its components

I communicate the effect sizes related to the second research question in Table 4.4 - '2-step', 'PL', 'MS' columns. They are reported separately for PL and MS components versus the 2-step. There are no significant effects observed for any of the labelling configurations. Moreover, for the considered sample MS effect sizes are negative for rebounds 7 and 11 (Tab. 4.5). The negative effect size in the considered setting means that the MS compound performs better than the 2-step approach. This effect is insignificant, hence does not generalise to the population.

I perform statistical tests to check if the null hypothesis H_{02} can be rejected. The original data used in the tests are provided in Table 4.4, for the target (2-step) and control groups (MS and PL). I communicate the test outcomes in Table 4.5. In the same table, I report the compared groups' standard deviations, means and medians. The results of the tests are interpreted in section 4.3.3. Finally, to support the reader, I plot the performance of the considered groups in the supplementary data, Figure B.4. There are 6 tests run in this experiment family, hence after applying Bonferroni corrections, the corrected significance level is $\alpha = .05/6 = .0083$.

4.2.3 Model analysis

Below I report the exploratory analysis of the trained models. I choose two models trained on the same contract but with different labelling configurations. Namely, the analysis of the models trained on the ESH2019 contract, with rebounds 7 and 11 is reported. The choice is motivated by very different numbers of features after the feature selection step.

The core analysis is done on the decision plots, provided in Figures 4.4 and 4.5. Since no pattern was observed when plotting the whole sample, I illustrate a random sub-sample of 100 entries from the top 1000 entries sorted by the per-feature contribution. Additionally, Figures B.5 & B.6 provide summary plots with the SHAP values of the models for the whole sample in the supplementary materials. There are 29 features in the model trained on the rebound 11 configuration, and 8 features in the one trained on the rebound 7 labels. 7 features are present in both models: PL23, MS6_80, MS0, PL24, MS2, MS6_20, MS6_200. Contributions from the features in the rebound 7 model are generally larger, also, the overall confidence of the model is higher. One might notice that the most impactful features are coming from the Market Shift features. In Figure 4.4 one can see two general decision patterns: one ending up at around 0.12-0.2 output probability and another one consisting of a number of misclassified entries ending up at around 0.8 output probability. The output probabilities correspond to the positive class (price reversal). In the first decision path, most

Table 4.5 Statistics supporting the outcomes of the Wilcoxon test which assesses whether CatBoost estimator with the 2-step feature extraction ('CB' column) leads to a better classification performance than the always-positive output classifier ('Null' column) and whether CatBoost estimator with the 2-step feature extraction ('2-step' column) leads to a better classification performance than each of the single-step feature extraction ('PL' and 'MS' columns). The result is reported for the rebound labelling configurations of 7, 11 and 15 ticks.

Statistics	Test Groups					
	Rebound 7					
	RQ1		**RQ2**			
			PL		**MS**	
One-tailed Wilcoxon test p-value	$< .001$		.0049		.96	
Test Statistics	55.0		52.0		11.0	
Effect Size (Hedges g_{av})	0.64 ± 1.02		0.57 ± 1.15		-0.1 ± 1.07	
	CB	Null	2-step	PL	2-step	MS
Mean (Precision)	.19	.18	.19	.18	.19	.20
Median (Precision)	.18	.17	.18	.18	.18	.19
Standard Deviation (Precision)	.025	.0151	.025	.018	.025	.028
	Rebound 11					
	RQ1		**RQ2**			
			PL		**MS**	
One-tailed Wilcoxon test p-value	.053		.080		.58	
Test Statistics	44.0		42.0		26.0	
Effect Size (Hedges g_{av})	0.38 ± 0.96		0.6 ± 1.16		-0.02 ± 1.06	
	CB	Null	2-step	PL	2-step	MS
Mean (Precision)	.18	.17	.18	.18	.18	.19
Median (Precision)	.18	.17	.18	.18	.18	.18
Standard Deviation (Precision)	.022	.0112	.022	.012	.022	.017
	Rebound 15					
	RQ1		**RQ2**			
			PL		**MS**	
One-tailed Wilcoxon test p-value	.0049		.052		.35	
Test Statistics	52.0		44.0		32.0	
Effect Size (Hedges g_{av})	1.05 ± 1.16		0.8 ± 1.22		0.2 ± 1.07	
	CB	Null	2-step	PL	2-step	MS
Mean (Precision)	.17	.16	.17	.16	.17	.17
Median (Precision)	.17	.16	.17	.16	.17	.17
Standard Deviation (Precision)	.012	.0055	.012	.006	.012	.010

of the MS features contribute consistently towards the negative class, however, PL23 and PL2 often push the probability in the opposite direction. In the second decision path, PL23

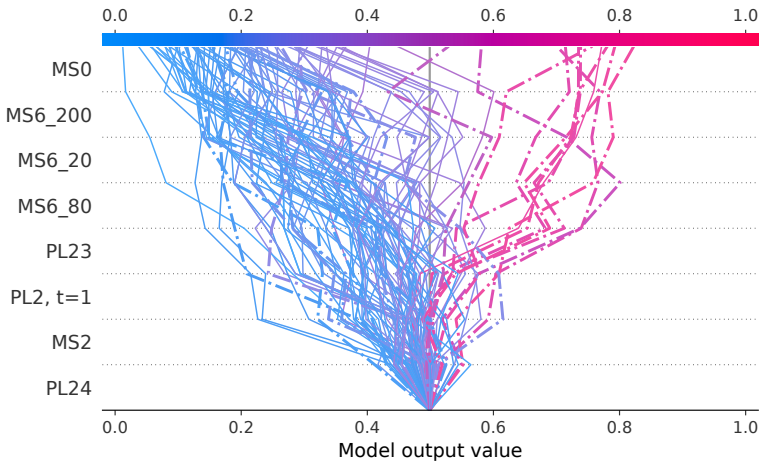


Fig. 4.4 The figure illustrates the feature contributions to the output on a per-entry basis for the CatBoost model, trained on ESH2019, rebound 7 configuration. The X-axis shows the strength of the contribution either towards a positive class (when the change is >0) or towards the negative one. Colours indicate the model output confidence of the positive class - blue corresponds to the negative class (crossing) and red - to the positive (rebound). Features are sorted by importance descending from the top. Misclassified entries are depicted with dashed lines.

has the most persistent effect towards the positive class, which is opposed by MS6_80 in some cases and gets almost no contribution from MS0 and MS6_200 features.

In Figure 4.5 there is a skew in the output probabilities towards the negative class. Contributions from the PL features are less pronounced than for the rebound 7 model - the top 8 features belong to the MS feature extraction step. There is no obvious decision path with misclassified entries. At the same time, I see strong contributions towards negative outputs from MS6_20 and MS6_40. This is especially noticeable for the entries which have output probabilities around 0.5 before MS6 are taken into account.

4.2.4 Simulated trading

In the current section, I report the cumulative profits for the 3 different labelling configurations in Fig. 4.6. In addition to the cumulative profits, I report annualised Sharpe ratios with a 5% risk-free annual profit. These results are reported for 15 ticks take-profit, as shown in Fig. 4.3. There is a descending trend in the Sharpe ratios across all the configurations. However, the behaviour of rebound 15 differs from the rest by performing worse till Oct 2018, then having a profitable time period which is not observed for the other configurations. I also run experiments with altered take-profits for 15 ticks rebound labelling and find out that decreasing take-profit leads to slightly worse cumulative profits and comparable

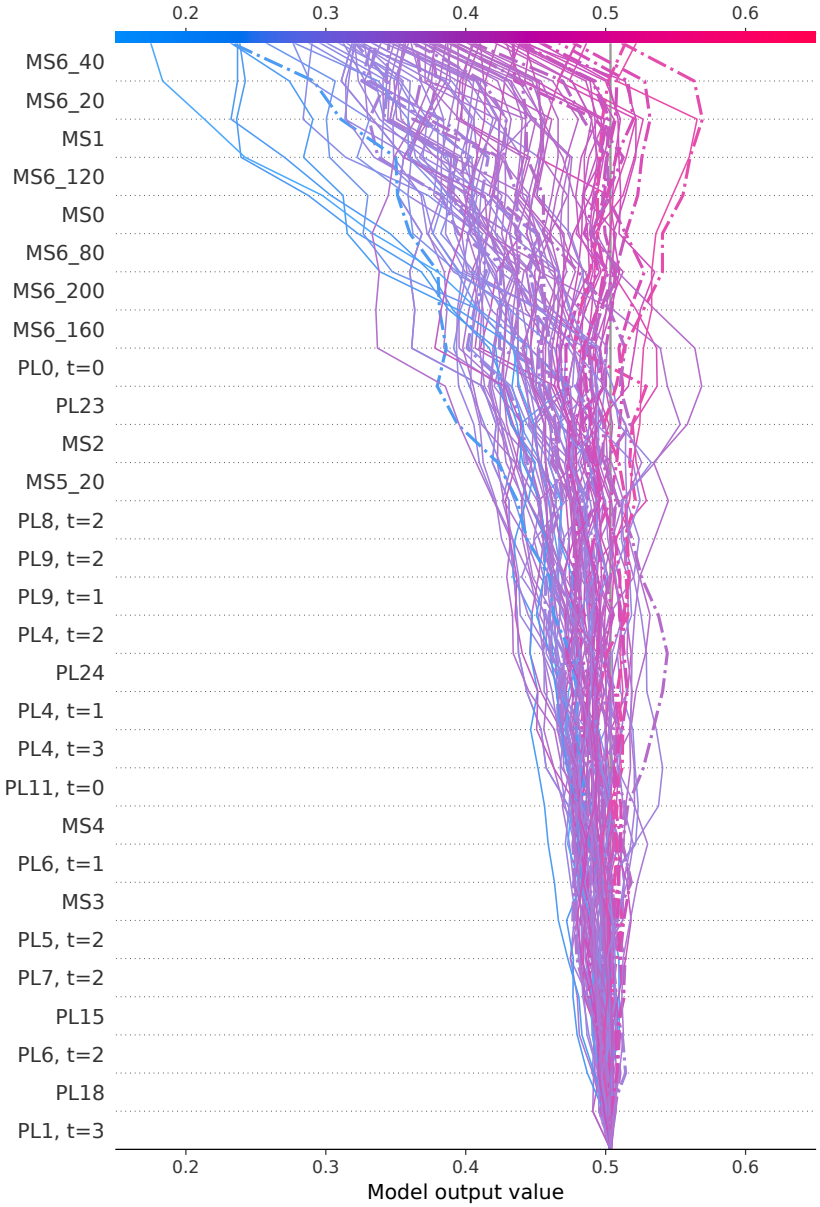


Fig. 4.5 The figure illustrates the feature contributions to the output on a per-entry basis for the CatBoost model, trained on ESH2019, Rebound 11 configuration. Read as above.

Sharpe ratios (Figure B.7). Seems that the profitability period after Oct 2018 depends on the take-profit size and vanishes if take it is reduced.

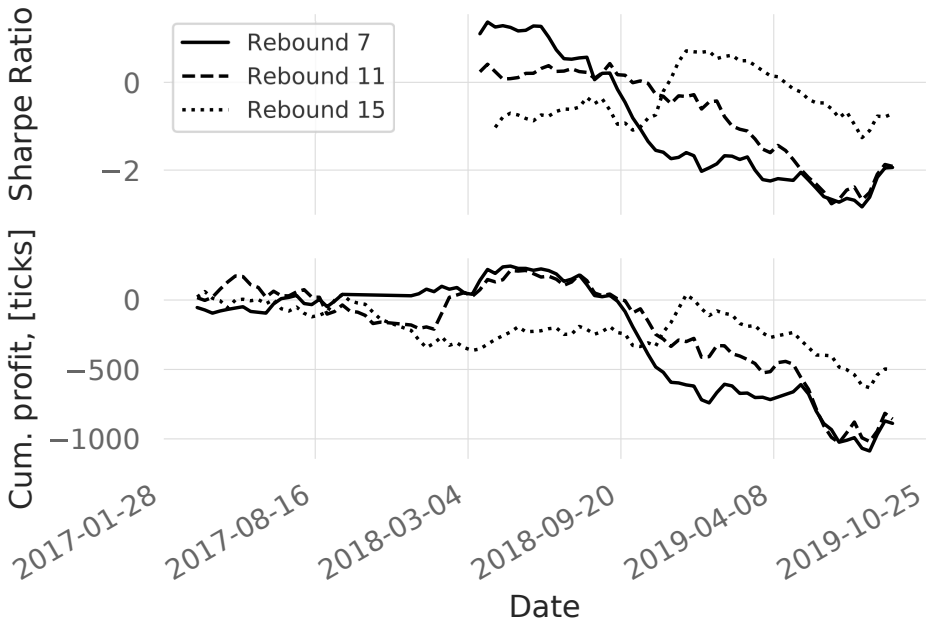


Fig. 4.6 Cumulative profit curves for the best-performing rebound configuration of 15 ticks and take-profits of 7, 11 and 15 ticks for years 2017-2019 with the corresponding annualised rolling Sharpe ratios (computed for 5% risk-free income). The trading fees are already included in the cumulative profits.

When computing the net outcomes of the trades, I add $4.2 per-contract trading costs based on the assessment of broker and clearance fees in Jan 2021. I do not set any slippage in the backtesting engine, since ES liquidity is large. However, I execute the stop-losses and take-profits on the tick following the close-position signal to account for order execution delays and slippages. This allows taking into account uncertainty rooting from large volatilities and gaps happening during the extraordinary market events. The backtesting is done on the tick data, therefore there are no bar-backtesting assumptions made.

4.3 Discussion

This section breaks down and analyses the results presented in the previous section (Section 4.3). Results are discussed in relation to the overall pattern (micro-event) extraction and model performance, then research questions, model analysis and, finally, simulated trading. Additionally, I discuss limitations in regards to our approach and how they can

be addressed. Finally, I present the view on implications for practitioners and intuition of potential advancements and future work in this area.

4.3.1 Pattern extraction

The numbers of price levels and ticks per contract follow the same trend (Table 4.3). Since the number of peaks is proportional to the number of ticks, one can say that the mean peak density is preserved over time to a large extent. In this context, the peak pattern can be considered stationary and appears in various market conditions.

4.3.2 RQ1 - Price Levels, CatBoost versus No-information estimator

The precision improvement for the CatBoost over the no-information estimator varies a lot across contracts (Table 4.4). At the same time, the improvement is largely preserved across the labelling configurations. It might be due to the original feature space, whose effectiveness depends a lot on the market state - for certain market states (and time periods) the utility of the feature space drops and the drop is quite consistent across the labelling configurations. Extensive research of the feature spaces would be necessary to make further claims. The overall performance of the models is weak on an absolute scale, however, it is comparable to the existing body of knowledge in the area of financial markets [169].

Effect sizes in RQ1 Table 4.5 are not significant. The significance here means that the chosen feature space with the model statistically significantly contributes towards the performance improvement with respect to the no-information model. Positive but insignificant effect size means that there is an improvement that is limited to the considered sample and is unlikely to generalise to the unseen data (statistical population).

Assessing the results of the statistical tests, I use the significance level corrected for the multiple comparisons. Tests run on the rebound 7 and 15 configurations result in significant p-values, rebound 11 is insignificant (Table 4.5). Hence, I reject the null hypothesis H_{01} for the labelling configurations of 7 and 15 ticks. This outcome is not supported by the effect sizes. This divergence between the test and effect sizes indicates a need for feature space optimisation before the use of the approach in the live trading setting. The insignificant outcome may be caused by particular market properties, or by unsuitable feature space for this particular labelling configuration. Interestingly, standard deviations differ consistently between the compared groups across the configurations. While no-information model performance depends solely on the fraction of the positively labelled entries, CatBoost performance additionally depends on the suitability of the feature space and model parameters - this likely explains higher standard deviations in the case of the CatBoost model.

4.3.3 RQ2 - Price Levels, 2-step feature extraction versus its components

Computing the effect sizes in Table 4.5 I compare the 2-step feature extraction approach to its components - PL and MS. I do not see any significant effects showing supremacy of any of the approaches. Negative effect sizes in the case of the Market Shift component mean that in the considered sample MS performs better than the 2-step approach. This result does not generalise to the unseen data as its confidence intervals cross the 0-threshold (are not significant). The possible explanation of the result is the much larger feature space of the 2-step approach (consisting of PL and MS features) than the MS compound. In case if PL features are generally less useful than MS, which is empirically supported by our model analysis (Figs B.5 and B.6), they might have a negative impact during the feature selection process by introducing noise.

Assessing the statistical tests, I use the significance level corrected for 6 comparisons. The p-values from Table 4.5 show that there is no significant outcome and the null hypothesis H_{02} cannot be rejected. While the 2-step approach does not bring any improvement to the pipeline, there is no evidence that it significantly harms the performance either. It might be the case that if PL features are designed differently, the method could benefit from them. I withhold from iteratively tweaking the feature space to avoid any loss of statistical power. I find this aspect interesting for future work, however, it would require increasing the sample size to be able to account for the increased number of comparisons.

4.3.4 Simulated trading

In the simulated trading, I observe an interesting result - data labelling configuration has more impact on the profitability than the take-profits (Figures B.7,4.6). I hypothesise that the reason is the simplistic trading strategy which is overused by the trading community in various configurations. In contrast, the labelling configuration is less straightforward and has more impact on profitability. Note that the obtained precision (Table 4.4) cannot be directly related to the modelled profitability trading strategy as there might be multiple price levels extracted within the time interval of a single trade. Consequently, there are extrema that are not traded.

It is hard to expect consistent profitability considering the simplicity of the strategy and lack of optimisation of the feature space, however, even in the current setting one can see profitable episodes (Figure 4.6). The objective of the experiments is not to provide a ready-to-trade strategy, but rather to demonstrate the approach.

4.3.5 Limitations

The proposed experiment design is one of the many ways the financial markets can be studied empirically. Statistical methods often have strong use case conditions and assumptions. When there is no entirely suitable tool available, one chooses the closest matching solution. While it is advised to use Glass's Δ in case of the significantly different standard deviations between groups, this measure does not have corrections for the paired data. Hence, in the proposed experimental design I choose to stick to the Hedge's g_{av}. For the sake of completeness, I verified the results using Glass's Δ - 15 ticks rebound effect size becomes significant in the RQ1.

In the backtesting, I use the last trade price to define ticks and do not take into account bid-ask spreads. In live trading, trades are executed by bid or ask price, depending on the direction of the trade. In reality, it leads to an implicit cost of the size of the bid-ask spread per trade. This is crucial for intraday trading as average per-trade profits often lay within a couple of ticks. Moreover, when modelling order executions, I do not consider per-tick volumes coming from aggressive buyers and sellers (bid and ask). It might be the case that for some ticks only aggressive buyers (sellers) are present, and the strategy executes a long (short) limit order. This leads to uncertainty in opening positions - in reality, some of the profitable orders may have not been filled. At the same time, losing orders would always be executed.

Another limitation is that I do not model order queues, and, consequently cannot guarantee that orders would have been filled if submitted live even if both bid and ask volumes are present in the tick. This is crucial for high-frequency trading (HFT), where thousands of trades are performed daily with tiny take-profits and stop-losses, but has less impact on the trade intervals considered in the experiments. Finally, there is an assumption that the strategy entering the market does not change its state significantly. I believe it is valid to assume so considering the liquidity of S&P E-mini futures. Commenting on the potential profitability of the method, I compare the obtained Sharpe Ratios to the existing body of knowledge. For instance, a paper by Xiong et al. [170] reports the Sharpe Ratio of 1.79 using a deep reinforcement learning approach. The results of the study were obtained for US stocks. Another study, by Yang et al. [171], reports the Sharpe Ratio of 1.3. In the study, the authors consider a subset of 30 US stocks data. The approach suggested in the current study offers worse performance, with the Sharpe Ratios being around 0. However, when directly comparing performance, the backtesting environment assumptions, as well as data granularity, should be taken into account. It is essential to highlight that the purpose of the current study is defined by the research questions, and not aimed at maximisation of the

strategy profitability. In the next chapter, a method of performance superior to the listed studies will be proposed.

4.3.6 Implications for practitioners

I provide a systematic approach to the evaluation of automated trading strategies. While large market participants have internal evaluation procedures, I believe that the conducted research could support various existing pipelines. Considering the state of the matter with the lack of code and data publishing in the field, I am confident that the demonstrated approach can be used towards improving the generalisability and reproducibility of research. Specific methods like extrema classification and 2-step feature extraction are a good baseline for the typical effect sizes observed in the field.

4.3.7 Future work

In the current chapter, I have proposed an approach for extracting extrema from the time series and classifying them. Since the peaks are quite consistently present in the market, their characteristics might be used for assessing the market state in a particular time scale.

The approach can be validated for trading trends - in this case, one would aim to classify price level crossings with high precision. A stricter definition of the crossings in terms of the price movement is necessary for that.

In terms of improving the strategy, there is a couple of things that can be done. For instance: take-profit and stop-loss offsets might be linked to the volatility instead of being constant. Also, flat strategies usually work better at certain times of the day - it would be wise to interrupt trading before USA and EU session starts and ends, as well as scheduled reports, news and impactful speeches. Additionally, all the mentioned parameters I have chosen can be looked into and optimised to the needs of the market participant.

In terms of the chosen model, it would be interesting comparing the CatBoost classifier to DA-RNN [172] model as it makes use of the attention-based architecture designed on the basis of the recent breakthrough in the area of natural language processing [54].

Finally, I see a gap in the available FLOSS (Free/Libre Open Source Software) backtesting tools. To the best of my knowledge, there is no publicly available backtesting engine taking into account bid and ask prices and order queues. While there are solutions with this functionality provided as parts of the proprietary trading platforms, they can only be used as a black box. An open-source engine would contribute to the transparency of the field and has the potential to become the solution for both the research and industry worlds.

4.4 Chapter conclusion

The current chapter showcased an end-to-end approach to perform automated trading using price extrema as micro-events, which are automatically extracted and classified. Whilst extrema have been discussed as potentially high performance means for trading decisions, there has been no work proposing means to automatically extract them from data and design a strategy. The current work demonstrated an automated pipeline using this approach and showed some interesting results.

This chapter has presented every single aspect of data processing, feature extraction, feature evaluation and selection, machine learning estimator optimisation and training, as well as details of the trading strategy. Moreover, I statistically assessed the findings - the null hypothesis was rejected when answering RQ1 - the proposed approach performs statistically better than the baseline. I did not observe any significant effect sizes for RQ2 and could not reject the null hypothesis. Hence, the use of the 2-step feature extraction does not improve the performance of the approach for the proposed feature space and the model. However, there is no evidence that it significantly hampers the performance of the pipeline either. I hope that providing every single step of the ATP, would enable further research in this area and be useful to a varied audience. I conclude by providing samples of the code online [166].

While price extrema are thought to manifest the swings of the beliefs of the market participants, they might be also formed due to a "thin" market (low liquidity). In the next chapter, I look into a more holistic way of extracting the micro-events from the financial time series data. Namely, I use the volumes property of the market together with the price to unambiguously identify the attention of the market participants to a particular price in time.

Chapter 5

Event Detection in Volume Profile Patterns of Financial Time Series

Financial markets are a source of non-stationary multidimensional time series which has been drawing attention for decades. Each financial instrument has specific properties in how it changes over time, making its analysis a complex task. Improvement of understanding and development of methods for financial time series analysis is essential for successful operation on financial markets and a general increase of financial system stability. In this chapter, I propose a volume-based data pre-processing method for making financial time series more suitable for machine learning pipelines. I use a statistical approach for assessing the performance of the method. Namely, I propose a set of research questions, formally state the hypotheses, compute effect sizes with confidence intervals, and reject the null hypotheses. Additionally, I assess the trading performance of the proposed method on historical data and compare it to the approach from the previous chapter (Chapter 4). My analysis shows that the proposed volume-based method allows successful classification of the financial time series events, and also leads to better classification performance than a price action-based method, excelling specifically on more liquid financial instruments. Finally, I propose an approach for obtaining feature interactions explicitly from tree-based models and compare the approach to SHAP local explanations method.

5.1 Material and methods

In this section, I provide details of the aims of the study, all the datasets used, experiments performed, model training and statistical evaluation methodology. Most of the discussion

around the market structures will focus on the currently proposed volume bars, details of price levels approach are discussed in the previous chapter.

5.1.1 Research gap & aims

When studying financial time series, it is natural to do so in the context of the financial market activities, like trading and investing. Trading platforms offer a wide range of time series analysis tools, however since the analysis was done manually in the past, these methods were originally developed for humans. They are often adopted in the automated trading systems off the shelf. This fact creates a knowledge gap in the field, as machine learning methods might have different requirements to the data and limitations in the analysis in comparison to human intelligence. To the best of my knowledge, the way I am addressing it was not studied by the research community. In the current chapter, I focus on financial time series patterns, based on volume profiles, by making them suitable for machine learning pipelines.

I use a state-of-the-art boosting trees algorithm - CatBoost [173]. It is explainable by post hoc explanations i.e. SHAP [19], efficient and easy to tune. To my knowledge, there is no study investigating whether SHAP explanations of CatBoost, which are derived from an approximation of the original model, are applicable to the financial markets. In the current work, I bridge the gap by comparing SHAP feature interactions to the ones extracted explicitly from the CatBoost model.

I state four research questions. The questions are aimed at a detailed assessment of the proposed method performance as well as empirical identification of optimal performance conditions. Additionally, I tackle the interpretability of the used classifier by directly extracting feature interactions from it.

The proposed method allows the extraction of events (or patterns) of certain properties from the financial time series. In order to make use of the extracted patterns, I classify them into scenarios depending on the price action after the pattern is formed. It is done the same way as in chapter 4). I consider two the most general price behaviours which can be traded - price crossing the *target* and price reversing from the *target*. While in the previous chapter the targets were price extrema, here it is POC (Point Of Control - the price with the largest volume in the volume profile). By choosing a particular target, I informally attempt to bring stationarity in the non-stationary financial time series.

The classification depends a lot on the feature space. Hence, answering the first two research questions, I investigate whether the chosen market microstructure-based feature space and model are appropriate for the proposed method and further experiments. Below

I list the research questions, while the hypotheses are formulated further in the text. This way I gradually introduce technical aspects of the study to the reader.

RQ1: *Given our proposed volume-based pattern extraction method baseline performance (as always-positive), can one further increase it with a domain-led feature engineering and ML model?*

RQ2: *Are the proposed volume-based patterns potentially better suitable for trading than price level-based?*

To answer this question generally, I do not limit myself to a particular trading strategy. Instead, I set up a classification task and compare the classification performance of the methods.

For the third research question, I hypothesise that volume-based patterns will perform statistically better on a more liquid market (S&P E-mini in our case). The reasoning is that if there are more large players on the market, the price is less price-action driven and "noisy". I use a volume-based approach for pattern extraction and expect it to be more suitable for liquid markets.

RQ3: *Does the classification of volume-based patterns show better results on a liquid market?*

While explainability of model outputs might be not necessarily critical in textual communications micro-event detection, in finance it is essential to have justifications behind decisions. Explanations of complex ML models are not always straightforward. Since financial time series is very challenging in terms of analysis, the trained models offer performance that would be considered poor in other domains and such models would be disregarded. Hence, it is not clear whether the established explainability methods are suitable in the financial time series domain. Consequently, the final RQ arises from the field of explainable machine learning, seeking to assess to what extent one can apply commonly used explainability approaches in a financial time series setting.

RQ4: *Are feature interactions discovered with SHAP associated with the ones obtained directly from the decision paths?*

SHAP is a game-theoretical model-agnostic framework for interpretations of the model outputs [19]. Among other information, it provides a ranking of features based on contribution to the prediction per instance, and feature interactions. Since SHAPs provide an intuitive understanding and are easy to interpret, they are an ideal candidate to evaluate

diverse real-world models. Nevertheless, since SHAP values are an approximation of the original model, it would be useful to compare SHAPs to an explicit model interpretation method for completeness. To do so, I propose a way of obtaining any order feature interactions explicitly from a tree-based boosting model by leveraging the Monoforest approach [174].

5.1.2 Datasets

In the current study, I use a convenience sample of S&P E-mini (ES) and British Pound (B6) futures instruments data, traded on CME Globex. Namely, I perform the experiments on a time range of 39 months - from March 2017 until June 2020. When discussing the results, the data outside of this time range is considered as a statistical population.

Since order book data is less commonly available and requires extra assumptions for pre-processing, I use tick data with only Time&Sales per-tick statistics. Concretely: I obtain numbers of trades and volumes performed by aggressive sellers and buyers, at the bid and ask, respectively. Additionally, I collect millisecond-resolution time stamps on tick starts and ends as well as the prices of the ticks. While in Chapter 4 I use actively traded contract data samples, here I want to compare results between financial instruments with different expiration dates. Hence, to make the instruments comparable, I use price-adjusted volume-based rollover contracts data. This way the contract transitions become seamless and instruments with different expiration dates can be directly compared. Detailed data pre-processing is explained later on.

5.1.3 Volume-centred range bars (VCRB)

A volume profile is a representation of trading activity over a specified time and price range, its variations are also called market profiles. They are commonly used for market characterisation (when considering daily volume profiles) and trading. Volume profiles are usually built from the temporal or price range bars[1], once every n bars. These settings make the obtained profiles highly non-stationary and applying ML to them has the same drawbacks as feeding raw financial time series into a model. Namely, in such a setting models are hard to fit and require large datasets due to the constantly changing properties of the data.

In the proposed method I aim to increase the stationarity of the volume profiles, as well as obtain as many entries per time interval as possible. I illustrate a high-level diagram of the VCRB extraction pipeline in Figure 5.1.

[1] Please see Table 2.1 for definition

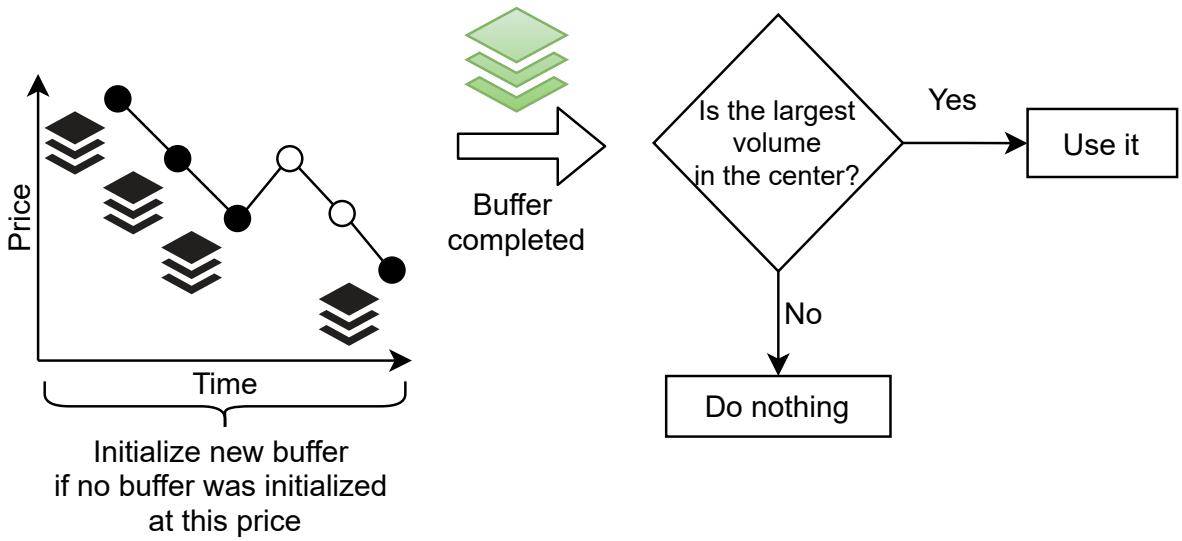


Fig. 5.1 Visualisation of the volume-based pattern extraction approach. Entries filled with black indicate initialisations of new buffers.

The core of the proposed method is a set of tick buffers simultaneously filled on a per-tick basis. A set is formed by new buffers started at a price if there is currently no incomplete buffer that has been initialised at the price in the past. Consequently, there is a buffer initialised at every possible historical price. This way I ensure capturing all the patterns potentially having the desired volume-centred configuration which is described in the next paragraph. When the desired price range is reached for a buffer, it is considered complete and is not filled with the streamed ticks anymore - the same way as it works for conventional range bars. The price range is measured as a minimum tick price subtracted from the maximum price in the buffer.

After the buffer is complete, I build its volume profile and check if the largest volume is in its centre. If so, I proceed with computing its features and labelling it. Otherwise, I ignore it. As I show later in Section 4.2, the tick buffer fulfilling the condition can be visualised as a volume-centred range bar (VCRB). The largest volume in the volume profile of the buffer is called Point of Control (POC). I use the PoC as a target for labelling, where price either reverses from or crosses it. From now on I refer to the price range as *range configuration*.

5.1.4 Experiment Design

In the current subsection, I describe all the components of the experiment design. I ensure that the design is as uniform as feasible across the experiments. When highlighting the differences, I refer to the experiments by the associated research questions - from RQ1 to

RQ4. The methodology proposed in Chapter 3 is reused in the current chapter. Also, its effect size assessment, model analysis, and hypothesis testing aligned with the machine learning components are consistent with the methodology of Chapter 3.

Label Design and Classification Setting

For the sake of comparability, in the current study, I reuse the labelling procedure from Chapter 4. The labelling process starts when the VCRB is formed. After the price reaches the target (PoC or extremum of the pattern for VCRB and price level, respectively), there are two scenarios: it reverses or continues its movement. For the reversal, I require the price to move for at least 15 ticks from the target. For the crossings, I take the 3 ticks beyond the target. Price reversals are labelled as positive examples and crossings as negative ones. Based on the domain knowledge, I consider these numbers suitable for intraday trading scenarios - this is also supported by intraday standard deviations of the price. Optimisation of these numbers is out of the scope of the current study.

I perform binary classification of the extracted VCRB and price levels patterns. Namely: I classify whether the entries are followed by price rebounds (reversals) from the target or target crossings. CatBoost is used as a classifier throughout the chapter as an example of the robust and efficient cutting-edge ML algorithm.

Feature Space and Model Parameter Tuning

I conduct two types of experiments - with and without feature selection and model parameter tuning. I do so to study different aspects of the matter:

- Experiments with feature selection and model tuning. Since the models are free to choose the feature subset and model configurations, these experiments are not limited by a particular feature set, but rather by an overall feature space. I use this setting for RQ1-3.

- Experiments with a fixed feature space and model parameters. These are aimed at studying feature interactions across datasets. Fixed feature space and model parameters ensure that any differences in the feature interactions are caused by the data or the model analysis approach and not by varying model configuration or feature space. These are used specifically in RQ4.

When designing the feature space I follow the approach proposed in Chapter 4 - I extract a set of features from the volume-based pattern or a price level - called Pattern (P) features, and the second one from the most recent ticks before the target is approached (being 2

ticks away from it) - called Market Shift (MS) features. In order to decrease the number of uncontrolled factors and their impact on the experiment design, I limit the feature space to only market microstructure-based features. I list the features and the associated equations in Table 5.1. The provided equations are valid for volume-based patterns. Since the data is available only below or above the extrema in the case of the price level patterns, the features are computed slightly differently. Negative t values are considered as an odd number of ticks distances from extrema and positive - as even numbers. I believe that this alteration is the closest possible to the original while preserving the domain knowledge at the same time. Numbers 237 and 21 in Table 5.1 were taken arbitrarily with three requirements: not round, within the trading time interval range, substantially different. I performed no optimisation of these parameters.

To perform the feature selection, I run a Recursive Feature Elimination with Cross-Validation (RFECV) process with a step of 1 and use the internal CatBoost feature importance measure for the feature ranking. Model parameter tuning is done for the following model parameters: i) a number of iterations; ii) maximum tree depth; iii) whether a temporal component of the data is considered, and iv) L2-regularisation. I considered it infeasible to optimise over a larger number of parameters, as with the current setting the experiments take in total >7k CPU hours. When optimising the feature space and tuning the model parameters, the precision metric is used - I explain the reasons in Chapter 4, Section 4.1.4.

Performance Metrics

For the RQ1 experiments, the performance is evaluated using the precision metric as it directly defines the potential profitability of the model. I elaborate on this in the Discussion section.

While the underlying data is the same for all the RQ1 experiments for null and alternative hypotheses, there is no constraint on the performance metrics used. In RQ2 and RQ3 I compare the classification performance of differently imbalanced datasets, hence I use binary Precision-Recall Area Under Curve (PR-AUC) score. This measure is advised for use in this setting [175].

For the sake of completeness and easier interpretation of the results, I provide the mentioned metrics together with ROC-AUC and f1-score for all the experiments. As there is no notion of performance in the experiments associated with RQ4, I describe its metric later, in Section 5.1.5.

Table 5.1 Features (referred to by code '[code]') used in the study in two stages, Stage 1 - Pattern and Stage 2 - Market Shift (MS)

	Equation	**Description**
Pattern features	$\frac{\sum_{t\in[1;5]}^{p=X+t}(V_b)}{\sum_{t\in[-5;-1]}^{p=X+t}(V_b)}$	Sum of upper (above PoC) bid volumes divided by lower ones [P0]
	$\frac{\sum_{t\in[1;5]}^{p=X+t}(V_a)}{\sum_{t\in[-5;-1]}^{p=X+t}(V_a)}$	Sum of upper ask volumes divided by lower ones [P1]
	$\frac{\sum_{t\in[1;5]}^{p=X+t}(T_a)}{\sum_{t\in[-5;-1]}^{p=X+t}(T_a)}$	Number of upper bid trades divided by lower ones [P2]
	$\frac{\sum_{t\in[1;5]}^{p=X+t}(T_b)}{\sum_{t\in[-5;-1]}^{p=X+t}(T_b)}$	Number of upper ask trades divided by lower ones [P3]
	$\frac{\sum_{t\in[1;5]}^{p=X+t}(V_b)}{\sum_{t\in[1;5]}^{p=X+t}1}$	Average upper bid trade size [P4]
	$\frac{\sum_{t\in[1;5]}^{p=X+t}(V_a)}{\sum_{t\in[1;5]}^{p=X+t}1}$	Average upper ask trade size [P5]
	$\frac{\sum_{t\in[-5;-1]}^{p=X+t}(V_b)}{\sum_{t\in[-5;-1]}^{p=X+t}1}$	Average lower bid trade size [P6]
	$\frac{\sum_{t\in[-5;-1]}^{p=X+t}(V_a)}{\sum_{t\in[-5;-1]}^{p=X+t}1}$,	Average lower ask trade size [P7]
	$\frac{\sum^{p=X}V_b}{\sum_{t\in[1;5]}^{p=X+t}V_b}$	Sum of PoC bid volumes divided by sum of upper bid volumes [P8]
	$\frac{\sum^{p=X}V_a}{\sum_{t\in[1;5]}^{p=X+t}V_a}$	Sum of PoC ask volumes divided by sum of upper ask volumes [P9]
	$\frac{\sum^{p=X}V_b}{\sum_{t\in[-5;-1]}^{p=X+t}V_b}$	Sum of PoC bid volumes divided by sum of lower bid volumes [P10]
	$\frac{\sum^{p=X}V_a}{\sum_{t\in[-5;-1]}^{p=X+t}V_a}$	Sum of PoC ask volumes divided by sum of lower ask volumes [P11]
	$\frac{\sum_t^{p=X+t}V_b}{\sum_t^{p=X+t}V_a}$	Sum of bid volumes divided by sum of ask volumes as price X+t; $t\in[-1;1]$ as different features [P12]
	$\frac{\sum_t^{p=X+t}T_b}{\sum_t^{p=X+t}T_a}$	Number of bid trades divided by number of ask trades as price X+t; $t\in[-1;1]$ as different features [P13]
	-	Side - below or above the price when the pattern is formed [P14]
MS features	$\frac{\sum_{t,b}^{w=237}(V_b)}{\sum_{t,a}^{w=237}(V_a)}$	Fraction of bid over ask volume for last 237 ticks [MS0]
	$\frac{\sum_{t,b}^{w=237}(T_b)}{\sum_{t,a}^{w=237}(T_a)}$	Fraction of bid over ask trades for last 237 ticks [MS1]
	$\frac{\sum_t^{w=237}V_b}{\sum_t^{w=237}V_a}-\frac{\sum_t^{w=21}V_b}{\sum_t^{w=21}V_a}$	Fraction of bid/ask volumes for long minus short periods [MS2]
	$\frac{\sum_t^{w=237}T_b}{\sum_t^{w=237}T_a}-\frac{\sum_t^{w=21}T_b}{\sum_t^{w=21}T_a}$	Fraction of bid/ask trades for long minus short periods [MS3]
Key	t - number of ticks V - volume	a - ask p - price w - tick window P_N - neighbours until distance N b - bid T - trades X - PoC or extremum

Model Evaluation

Price-adjusted volume-based rollover pre-processing of the data allows splitting the data into 3-month chunks without any extra care of contract rollovers. I do so since the length of contracts is different for ES and B6. When training the models, I apply a temporal sliding window approach, using batch N for training, $N+1$ for testing, for $N \in [1, B-1]$, where B is the total number of the 3-month batches available. For the feature selection and model parameter tuning, I use 3-fold time series cross-validation within the training batch with the final re-training on the whole batch. To support the reader, I summarise the experiment design for all the experiments in Table 5.2. In the following subsections, I address the statistical component of the methodology.

Table 5.2 Experiment design across the experiments. F_{sel} & M_{tune} column accounts for feature selection and model tuning. "Comparison H_0/H_1" column indicates the settings used to obtain null and alternative hypotheses data. "Other" indicates whether the test is conducted on both instruments (ES, B6) and whether only VCRB pattern extraction method was used. "Metric" corresponds to the measurement variable. All hypotheses are evaluated using Wilcoxon test.

Experiment	F_{sel} & M_{tune}	Metric	Comparison H_0/H_1	Other
RQ1	✓	Precision	No-info. / CatBoost	VCRB; ES, B6
RQ2	✓	PR-AUC	Price levels / VCRB	ES, B6
RQ3	✓	PR-AUC	B6 / ES	VCRB
RQ4	-	Footrule distance	Bootstrap / SHAP, Monoforest-based	ES, B6

Statistical Evaluation

In the current study, I use effect sizes following the formalism of comparison of treatment and control groups. For the two groups, I perform a pair-matched comparison choosing suitable measurement variables for each case, as shown in Table 5.2. In the current setting, the treatment and control groups are represented by the methods applied to the time series batches.

Before evaluating the hypotheses, I compute Hedge's g_{av} effect sizes for paired data together with the .95 confidence intervals (CIs).

In the choice of the statistical test, I am guided by the same reasons as in Chapter 4 and choose Wilcoxon signed-rank test [176] as the best suitable candidate, applying its single-sided version to validate the hypotheses of the chapter.

I set the significance level for the statistical tests of the study to $\alpha = .05$. Finally, I apply Bonferroni corrections to all the statistical tests within each experiment family [105]. The

experiment families are defined based on the data and objectives - each research question forms a separate experiment family.

Model Interpretation

The challenge of interpreting the modern ML models roots in their complexity. Even considering decision tree-based models, interpretation becomes problematic when the number of features grows. When it comes to interpretations, the number of decision paths of the CatBoost model is usually well beyond the limit of manual analysis. The recently proposed Monoforest [174] approach represents the tree ensemble as a set of polynomials and makes the decision paths uniform. Additionally, its implementation gives access to machine-readable decision paths. This allows to retrieve the following information from each polynomial: the subset of the involved features, feature thresholds, as well as support (w) and contribution (c) to the output of the model.

I assume that interactions between features take place if they are found in the same decision path. Please note that the way SHAP defines feature interactions is principally different, as I elaborate in Discussion section. In order to get interactions for the whole model, I average across the decision paths in the following way:

$$I_{(F1,F2)} = \frac{\sum_{i=0}^{N} c \times w}{N}, \tag{5.1}$$

where N is the total number of decision paths containing features 1 and 2 ($F1$, $F2$), c is the contribution of the decision path to the model output and w is the support, computed as a number of times the path was activated in the training set divided by the total number of training entries.

If there is a single feature in the decision path, I consider it as a main effect. I include main effects into the interaction matrix in the same fashion as it is done for SHAP [19] - as diagonal elements of the matrix. When there are more than two features in the decision path, I treat them as multiple pair-wise interactions to be able to represent them in a single 2d matrix and compare them to SHAP interactions. By considering decision paths with a fixed number of features, one can get interactions of a particular order. Moreover, these interactions are directly comparable across orders, models and datasets.

Backtesting

I perform backtesting of the proposed method in the Python Backtrader platform. I use the same strategy, assumptions and trading fees as used in Chapter 4. I focus on performance comparison based on classification tasks over the backtesting simulations for two reasons: i)

different trading intensities lead to varying impacts of the modelling assumptions; ii) limiting our comparison to a particular strategy significantly decreases the generality of the finding. Elaborating on the first point: influences of order queues, slippages and bid-ask spreads are partially taken into account, however, it is fair to expect these effects to have increasing impacts with an increase of the trading intensity. Quantification of these is out of the scope of this thesis, however, might be very useful. I describe how the modelling assumptions might affect the backtesting results in the Discussion section. As a complementary analysis, in the current study, I report annual rolling Sharpe ratios for all the range configurations, and for the price level method.

5.1.5 Addressing Research Questions

In the current subsection, I list the conducted experiments associated with the research questions, as well as highlight any experiments-specific choices.

RQ1 - VCRB method, CatBoost versus no-information estimator

Firstly I assess the performance of the no-information and CatBoost classifiers and compare them. Prior to evaluating the hypotheses, I compute the effect sizes. Then, I run the statistical test with the following hypotheses:

H_{01}: CatBoost estimator performs equally or worse than the no-information model.

H_{11}: CatBoost estimator performs better than the no-information model.

In the results, I report statistical test outcomes for the configuration with the largest effect sizes, with the other configurations, are reported in the supplementary materials.

RQ2 - VCRB vs price levels approach

To answer the second research question I compare the performance of the two methods - price levels and VCRB. I obtain the PR-AUC classification performance for both methods and both instruments. Then, I compute the effect sizes, and, finally, run the statistical test with the following hypotheses:

H_{02}: Price level patterns are classified with performance equally good or better than volume-based patterns.

H_{12}: Volume-based patterns are classified with statistically better performance.

RQ3 - VCRB method, ES versus B6 datasets

Answering the third research question, I compare CatBoost classification performance on the VCRB-extracted data over the two datasets - B6 and ES, where the latter is far more liquid. After the PR-AUC classification performance is obtained, I compute the effect sizes and run the statistical tests with the following hypotheses:

H_{03}: Both instruments perform comparably or performance on B6 is statistically better.

H_{13}: Volume-based patterns are classified with statistically better performance for the instrument with higher liquidity (ES).

RQ4 - Relatedness of feature interactions from SHAP and decision paths

To answer the last research question I need to get the data representing the null hypothesis - having no relation (potentially in contrast to SHAP and decision paths), and propose a method for assessing the relatedness. As the first step, I obtain feature interactions in a form of a square matrix using SHAP and the Monoforest-based method. To generate the null hypothesis data, I bootstrap (randomly sample with replacement) the interaction matrices' elements separately for both approaches. I choose to generate 500 bootstrapped entries for each method. As a result, I get two sets of matrices with the same value distributions as the original feature interaction matrices (SHAP and Monoforest-based). Since I perform bootstrapping, by definition there is no association between any two matrices from the two sets.

Since both methods output interactions scaled differently, I make the matrices comparable by ranking the interaction strengths within each matrix. After that, I compare the ranks across the two methods by computing their Footrule distances [177]. This measure is designed specifically for ranks data and computed as the following:

$$D = |R_1^A - R_1^B| + |R_2^A - R_2^B| + \ldots + |R_n^A - R_n^B|, \tag{5.2}$$

where R is the position order of the element in the rankings A & B of the length n, and the lower indices correspond to the compared elements. Later the distances are used as a proxy to assess the relatedness of the methods - the larger the distance, the weaker the relationship.

In order to get a reliable null hypothesis distance, I compute the mean of the distances between the bootstrapped matrices. At this point, the relatedness of SHAP and decision paths methods can be compared against the bootstrapped data. To quantify the differences, I obtain the effect sizes on the Footrule distance (instead of the classification performance used in the other RQs). Finally, I run the statistical test with the following hypotheses:

H_{04}: There is no difference between Footrule distances on ranks of SHAP-decision paths and bootstrapped feature interactions matrices or SHAP-decision paths are larger.

H_{14}: There is a difference between Footrule distances on ranks of SHAP-decision paths and bootstrapped feature interactions matrices with SHAP-decision paths being smaller than bootstrapped.

5.2 Results

This section presents the results of the experiments described in the previous section. I provide a detailed analysis of how the results match the stated hypotheses in Section 5.3, and the current one only contains outcomes of the experiments.

5.2.1 Pattern extraction from the datasets

In this subsection, I report statistics on the original datasets as well as numbers of entries obtained from every dataset batch, for both pattern (or micro-event) extraction methods and financial instruments.

In Table 5.3 I show numbers of ticks and total volumes per batch for both instruments.

Table 5.3 Original datasets statistics. Volume columns correspond to the total volume traded per the stated time interval. The Ticks columns show the numbers of ticks per the time interval.

Batch	**ES**		**B6**	
	Volume	Ticks	Volume	Ticks
3/17 to 6/17	86123932	151965	6663094	118098
6/17 to 9/17	82384394	132964	6575559	115576
9/17 to 12/17	73925568	98600	7971963	142548
12/17 to 3/18	88050918	517451	7588515	159565
3/18 to 6/18	96653879	536280	7267680	131215
6/18 to 9/18	71968775	235841	6956995	116275
9/18 to 12/18	109410969	581345	7128825	155895
12/18 to 3/19	95948559	673838	6078533	136554
3/19 to 6/19	92201997	378788	6643282	132441
6/19 to 9/19	93229922	458141	5677468	94248
9/19 to 12/19	75613694	343666	7193235	153081
12/19 to 3/20	101547199	587658	6122643	100381
3/20 to 6/20	126756329	3482845	5593815	270096

One notices in Table 5.4 that in the year 2017 there are more entries for VCRB B6 than for ES, later the situation reverses. Overall, there are around 5-8 times fewer entries for

the price level-based method in comparison to the volume-based. I address the potential consequences of these differences for the hypothesis testing in the Discussion section.

Table 5.4 Numbers of extracted patterns for volume-based (VCRB) range 7, and price level-based (PL) methods, reported for the analysed data sets, both instruments.

Batch	**VCRB**		**PL**	
	ES	B6	ES	B6
3/17 to 6/17	2236	2785	458	278
6/17 to 9/17	1998	2789	360	267
9/17 to 12/17	1416	3440	268	378
12/17 to 3/18	10471	4018	1643	418
3/18 to 6/18	10231	3125	1695	335
6/18 to 9/18	4182	2643	736	287
9/18 to 12/18	11876	3789	1926	370
12/18 to 3/19	13937	3354	2234	348
3/19 to 6/19	6940	3480	1186	358
6/19 to 9/19	9111	2329	1413	235
9/19 to 12/19	6202	3878	1060	410
12/19 to 3/20	12886	2505	1856	227
3/20 to 6/20	88486	8373	12085	738

5.2.2 Volume-centred range bars

I generate VCRBs for range sizes of 5, 7, 9 and 11 ticks. For comparison purposes, I extract price-based patterns using a configuration from Chapter 4. I visualise the VCRBs in Fig. 5.2. As it was explained before (Section 5.1), the volume in the centre is the largest one. The volume distributions differ a lot for the provided entries. If there is a price with zero volume (second pattern from the left in Figure 5.2), I skip that price in the visualisation.

5.2.3 Prediction of the reversals and crossings

The initial experimental stages are feature selection and model parameter tuning. I do not report the optimised feature spaces and model parameters in the chapter making this data available in the publicly available reproducibility package [178, 179].

For the classification task, I report all the stated performance metrics for the configuration range 7, which is chosen based on RQ1 effect sizes, together with the PR-AUC metric for the price levels method in Tables 5.5 and 5.6. The rest of the metrics for price levels are reported in the supplementary materials, in Tables C.1 & C.2. Additionally, in the supple-

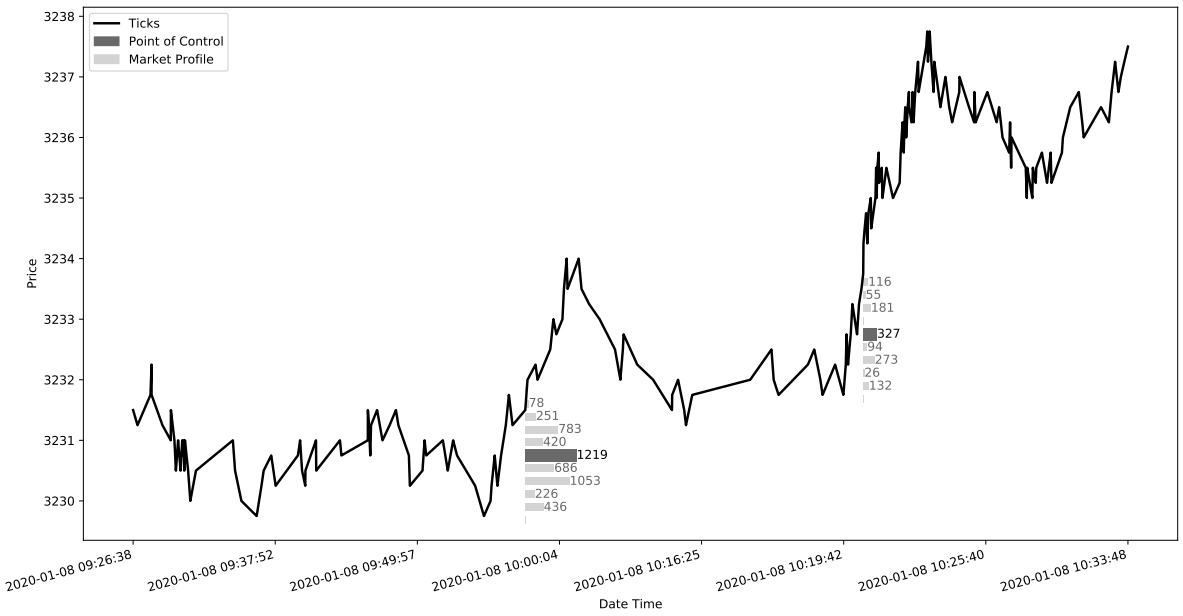


Fig. 5.2 Example of volume-centred range bars generated for ES instrument. Histograms indicate traded volumes within the buffer. Points of control are in the centre of the volume profiles, marked with dark grey. The profiles are formed when the price buffer is complete (9 ticks in this case). Zero-volume entries are not shown.

mentary materials I plot the data representing null and alternative hypotheses throughout the study - in Figures C.1, C.2, C.3 and C.4.

In all the experiment families I compute effect sizes for all the VCRB configurations (ranges 5, 7, 9 and 11). The statistical test results are reported only for the largest effect size configuration from the RQ1 experiment family, on S&P E-mini (ES) instrument. I communicate the rest of the statistical tests in the supplementary materials. When evaluating the statistical tests, I correct for multiple comparisons - the corrected significance levels are provided separately for each experiment group.

RQ1 - VCRB method, CatBoost versus No-information estimator

Here I provide the classification performance comparison between the CatBoost estimator and the no-information estimator. First, I plot the effect sizes with .95 confidence intervals in Fig. 5.3. Then, I provide performance statistics and the statistical test results in Table 5.7. Additionally, to allow easier comprehension of the results, I visualise the precision of the models from Tables 5.5 and 5.6 in supplementary materials, Fig. C.1.

From Figure 5.3 one sees that effect sizes for ES are generally larger. The effect size pattern across configurations is preserved between the two instruments with an exception

Table 5.5 Performance metrics for ES, volume-based pattern extraction configuration range 7. The dates are reported in the form MM/YY. While the results are poor on the absolute scale, it is expected in the financial time series domain, more details are provided in the Discussion section.

Batch	**VCRB**					**Price levels**
	PR-AUC	ROC-AUC	F1-score	Precision	Null_precision	PR-AUC
3/17 to 6/17	0.25	0.54	0.34	0.25	0.23	0.14
6/17 to 9/17	0.23	0.51	0.32	0.22	0.21	0.19
9/17 to 12/17	0.25	0.51	0.30	0.25	0.24	0.15
12/17 to 3/18	0.24	0.52	0.27	0.24	0.23	0.16
3/18 to 6/18	0.25	0.52	0.30	0.24	0.23	0.16
6/18 to 9/18	0.25	0.53	0.34	0.25	0.24	0.16
9/18 to 12/18	0.25	0.52	0.31	0.25	0.24	0.16
12/18 to 3/19	0.24	0.52	0.33	0.23	0.22	0.18
3/19 to 6/19	0.25	0.52	0.29	0.26	0.24	0.16
6/19 to 9/19	0.25	0.53	0.35	0.25	0.24	0.15
9/19 to 12/19	0.24	0.51	0.28	0.25	0.23	0.14
12/19 to 3/20	0.24	0.51	0.30	0.24	0.23	0.17
3/20 to 6/20	0.24	0.52	0.33	0.24	0.23	0.17

Table 5.6 Performance metrics for B6, volume-based pattern extraction configuration range 7. The dates are reported in the form MM/YY. While the results are poor on the absolute scale, it is expected in the financial time series domain, more details are provided in the Discussion section.

	VCRB					**Price levels**
Batch	PR-AUC	ROC-AUC	F1-score	Precision	Null_precision	PR-AUC
3/17 to 6/17	0.24	0.51	0.25	0.25	0.23	0.17
6/17 to 9/17	0.23	0.51	0.27	0.24	0.22	0.16
9/17 to 12/17	0.24	0.51	0.24	0.24	0.23	0.16
12/17 to 3/18	0.23	0.52	0.26	0.23	0.22	0.15
3/18 to 6/18	0.21	0.50	0.25	0.20	0.21	0.12
6/18 to 9/18	0.23	0.51	0.23	0.24	0.23	0.18
9/18 to 12/18	0.23	0.51	0.26	0.23	0.22	0.15
12/18 to 3/19	0.24	0.52	0.26	0.24	0.23	0.19
3/19 to 6/19	0.21	0.50	0.20	0.20	0.21	0.18
6/19 to 9/19	0.22	0.50	0.25	0.23	0.22	0.20
9/19 to 12/19	0.22	0.50	0.24	0.23	0.23	0.14
12/19 to 3/20	0.23	0.50	0.31	0.22	0.22	0.17
3/20 to 6/20	0.24	0.53	0.25	0.24	0.21	0.20

of range 7. For ES the maximum effect size is observed at range 7 and the minimum - at range 9, while for B6 the maximum is at range 5 and the minimum is at range 9. Finally, judging the significance of the effect sizes by the confidence intervals (CIs) crossing the

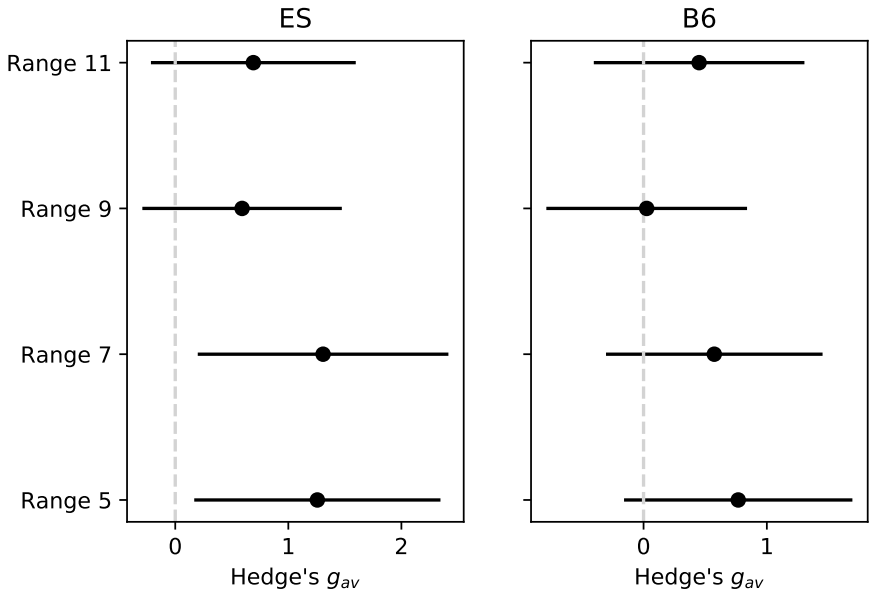


Fig. 5.3 Hedge's g_{av} effect sizes quantifying the improvement of the precision from using the CatBoost over the no-information estimator. The error bars illustrate the .95 confidence intervals, corrected for multiple comparisons. The dashed line corresponds to the significance threshold. Ranges correspond to different configurations of the pattern extraction method.

Table 5.7 Statistics supporting the outcomes of the Wilcoxon test. The test is aimed to check whether, on the VCRB data and the considered feature space, CatBoost performs significantly better than the no-information estimator. The provided result is for the range 7 configuration.

Statistics	**Dataset**			
	ES		**B6**	
One-tailed Wilcoxon test p-value	$< .001$		.024	
Test Statistics	91.0		74.0	
	CatBoost	No-information	CatBoost	No-information
Mean (precision)	0.24	0.23	0.23	0.22
Median (precision)	0.25	0.23	0.23	0.22
Standard Deviation (precision)	0.0092	0.0074	0.0138	0.0067

significance threshold line, one notices that there are no significant effect sizes in B6, while ranges 5 and 7 are significant for ES.

Performance statistics in Table 5.7 indicates no skew in the data between CatBoost and no-information models, at the same time there is a 1.2 and 2 times difference in the sample variance for ES and B6, respectively. The results for other VCRB configurations are provided in the supplementary materials, Table C.3. Since 8 statistical tests are conducted (4 configurations × 2 instruments), I correct the significance level as follows: $\alpha = .05/8 = .00625$.

For the rest of the experiment families, the statistical tests are reported for the range 7 configuration, as it demonstrated the largest effect size for RQ1 experiments.

RQ2 - VCRB versus price levels approach

In the current section, I present the results of the investigation on whether volume-based centred bars lead to a better classification performance than the price level approach. I report Hedge's g_{av} effect sizes on paired data with .95 confidence intervals in Fig. 5.4, and the outcomes of the statistical test with the supporting statistics in Table 5.8. Complementing the results, I visualise PR-AUC values from Tables 5.5 and 5.6 in supplementary materials, Fig. C.2).

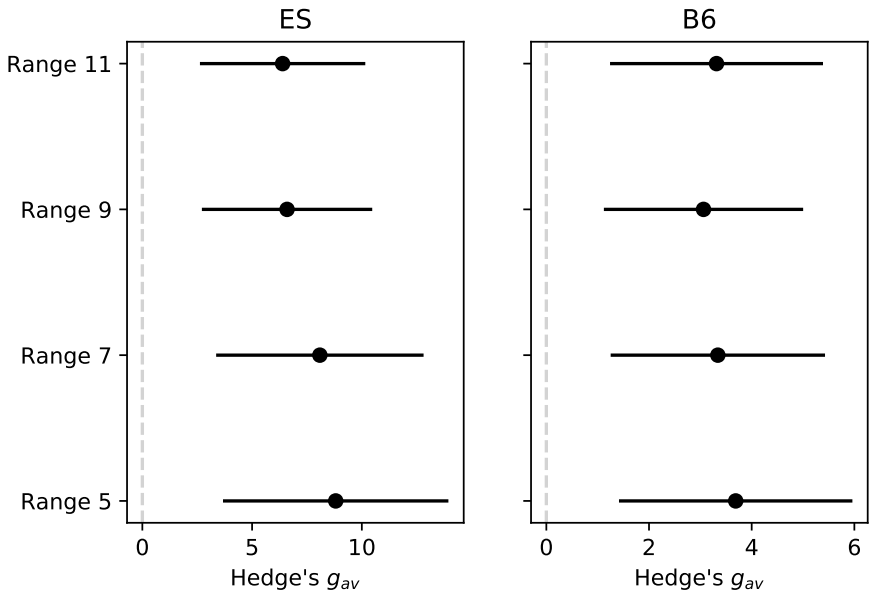


Fig. 5.4 Hedge's g_{av} effect sizes, quantifying the supremacy of the VCRB over the price levels approaches on the basis of the PR-AUC metric. The error bars illustrate the .95 confidence intervals, corrected for multiple comparisons. The dashed line accounts for the significance threshold. Ranges correspond to different configurations of the pattern extraction method.

One sees that all the effect sizes in Fig. 5.4 are significant as the confidence intervals do not overlap with the significance threshold line. Also, confidence intervals for ES are larger in comparison to B6. Lastly, the largest effect size is observed for range 5 configuration across instruments.

Performance statistics in Table 5.8 shows no skew in the data, however, sample variances differ up to 4 times between the two methods. The rest of the VCRB configurations are

Table 5.8 Statistics support the outcomes of the Wilcoxon test which checks whether the Volume-based pattern extraction method leads to better classification performance than the price level pattern extraction. The result is reported for the range 7 configuration.

Statistics	**Dataset**			
	ES		**B6**	
One-tailed Wilcoxon test p-value	< .001		< .001	
Test Statistics	91.0		91.0	
	VCRB	Price levels	VCRB	Price levels
Mean (PR-AUC)	0.24	0.16	0.23	0.17
Median (PR-AUC)	0.25	0.16	0.23	0.17
Standard Deviation (PR-AUC)	0.0066	0.0118	0.0094	0.022

reported in Table C.4. In the current experiment family, I run 8 tests in total, hence the corrected significance level is $\alpha = .05/8 = .00625$.

RQ3 - VCRB method, ES versus B6 datasets

Here I detail the results of comparing the classification performance of the VCRB entries extracted from ES and B6 datasets. I report effect sizes with .95 confidence intervals in Fig. 5.5, and the statistical test with the supporting statistics from range 7 configuration in Table 5.9. The test outcomes for the other configurations are provided in supplementary materials, in Table C.5. Additionally, I visualise PR-AUC performance from Tables 5.5 and 5.6 in supplementary materials, Fig. C.3.

Table 5.9 Statistics support the outcomes of the Wilcoxon test which assesses whether the VCRB pattern extraction method leads to better classification performance on the more liquid market (ES in comparison to B6). The result is reported for the range 7 configuration.

Statistics	**Datasets**	
One-tailed Paired Wilcoxon test p-value	< .001	
Test Statistics	88.0	
	ES	B6
Mean (PR-AUC)	0.24	0.23
Median (PR-AUC)	0.25	0.23
Standard Deviation (PR-AUC)	0.0066	0.0094

In Figure 5.5 larger range of the VCRB leads to a smaller effect size. At the same time, the confidence intervals shrink with the range increase, indicating that the effect for the larger ranges is smaller but more stable.

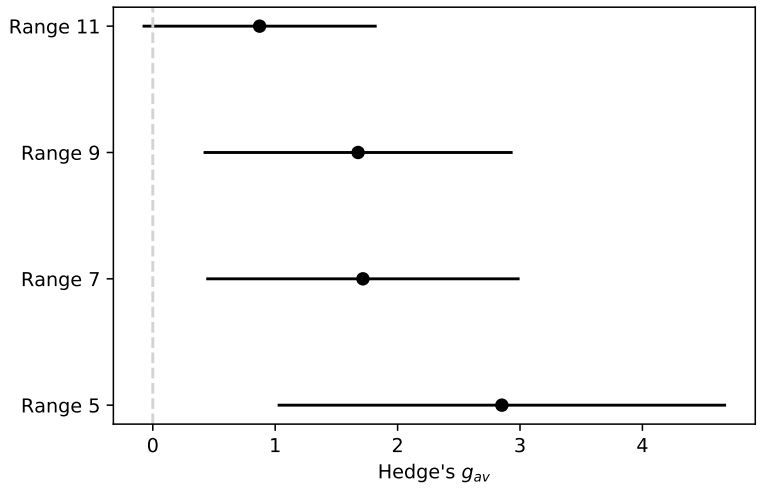


Fig. 5.5 Hedge's g_{av} effect sizes for Volume-based method PR-AUC performance improvement on ES over B6 datasets. Error bars illustrate the .95 confidence intervals corrected for multiple comparisons. Ranges correspond to different configurations of the pattern extraction method.

The statistics on the results in Table 5.9 shows that variances differ by around 50% between the samples and there is no skew in the distributions. In the current experiment family, I run 4 tests in total, hence the corrected significance level is $\alpha = .05/4 = .0125$.

5.2.4 Backtesting

Figures 5.6 and 5.7 show annual rolling Sharpe ratios with a 5% risk-free rate and cumulative profits in ticks for all the configurations of the VCRB and price level methods.

For easier interpretation of the figures, I plot Sharpe ratios averaged over 30-day periods. The lag of Sharpe ratio plots with respect to the cumulative profits is caused by the requirement of the Sharpe ratio to have a year of data available for obtaining the initial value.

5.2.5 RQ4 - Relatedness of feature interactions from SHAP and decision paths

The following results assess the relatedness of the feature interactions data extracted using SHAP and the proposed decision paths methods by comparing their relatedness to the bootstrapped data. Following the format of the previous experiments, I report effect sizes in Figure 5.8, and provide results of the statistical test as well as supporting statistics in Table 5.10. Additionally, I report the statistical test outcomes for the rest of the configurations

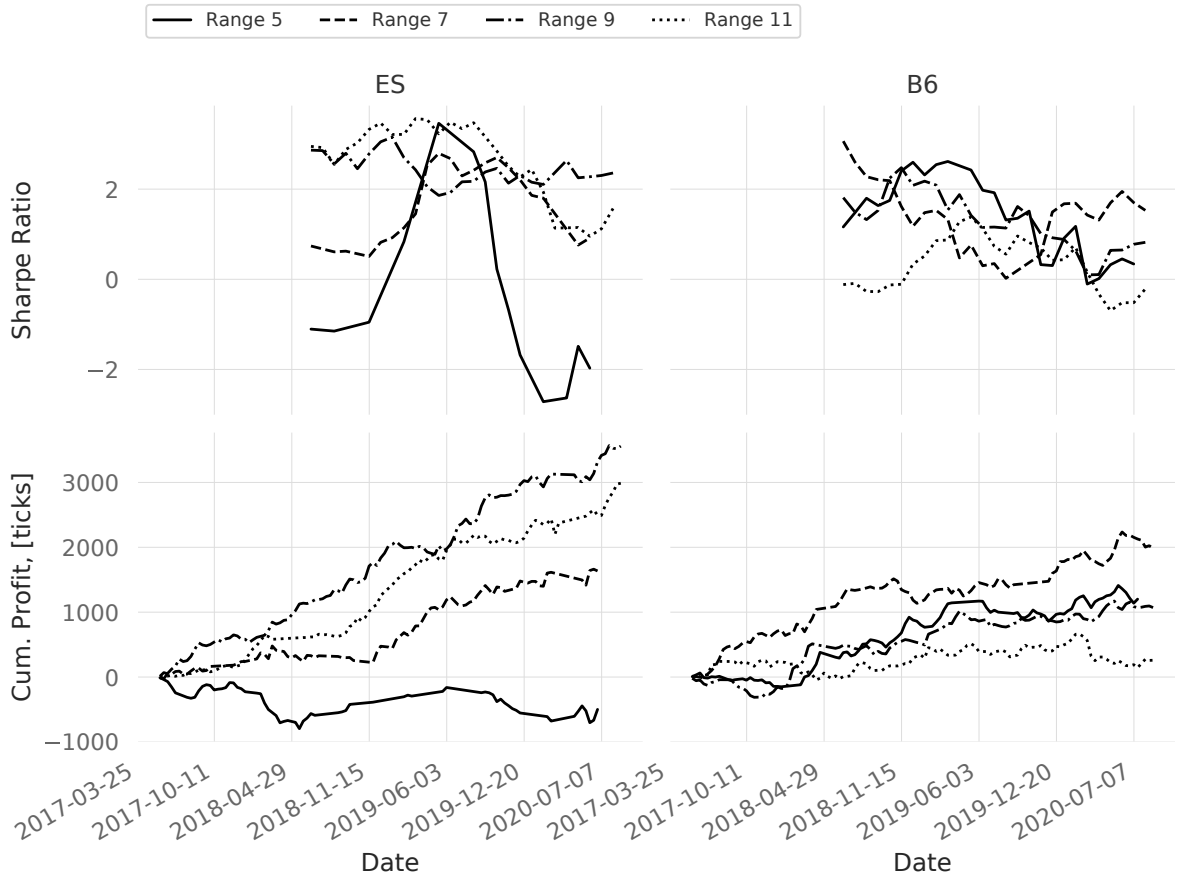


Fig. 5.6 Sharpe ratios and cumulative profits of the volume-based method configurations. Profits are provided in ticks. The simulation does not take bid-ask spreads and order queues into account, hence might be over-optimistic.

in the supplementary materials, Table C.6. Lastly, I plot the differences between data representing both hypotheses in supplementary materials, Figure C.4.

From Figure 5.8 one notices that the smallest effect sizes are observed for configurations 7 and 9 in ES and for configuration 5 in B6. Interestingly, the behaviour across the configurations is flipped for ES and B6. In Table 5.10 one can see that there is no skew in the data as mean and median values are very similar. At the same time, there are significant differences in the data variances, which I address in the Discussion section. The test statistics values are small in comparison to the previous tests - in the current experiment smaller Footrule distances represent the alternative hypothesis, hence the statistical test is computed for the opposite difference sign with respect to the previous cases. In the current experiment family, I run 8 tests in total, hence the corrected significance level is $\alpha = .05/8 = .00625$.

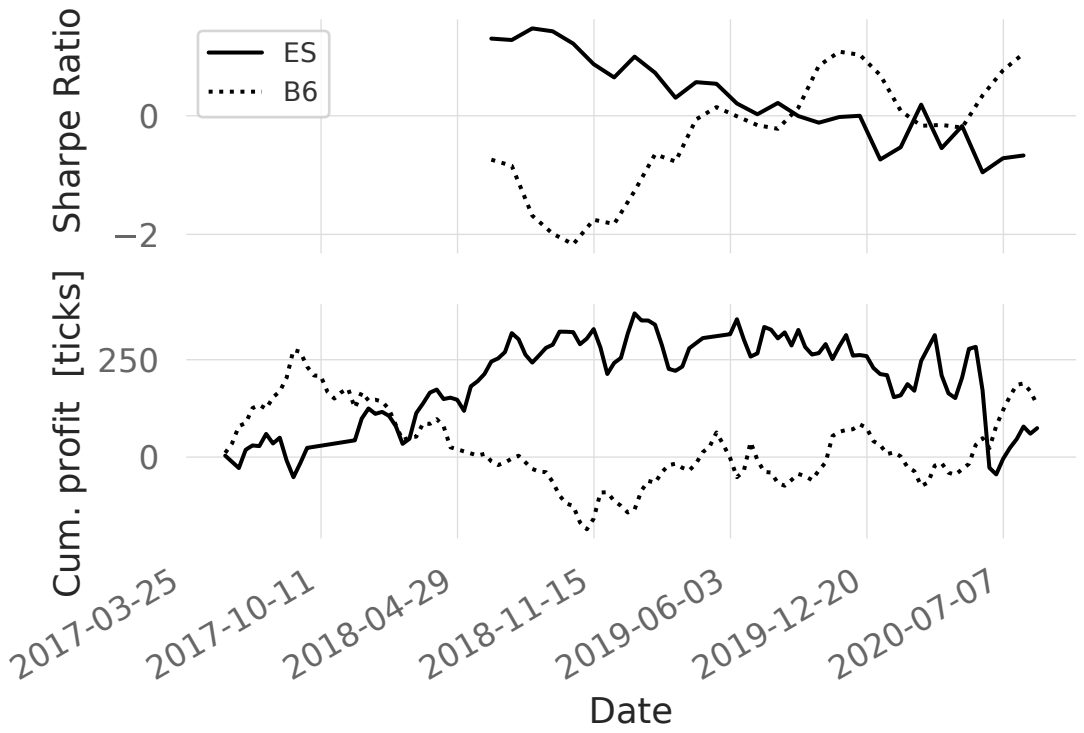


Fig. 5.7 Sharpe ratios and cumulative profits of the price level-based method. Profits are provided in ticks. The simulation does not take bid-ask spreads and order queues into account, hence might be over-optimistic.

Table 5.10 Outcomes of the one-tailed Wilcoxon test which check whether SHAP and decision paths feature interactions extraction methods are related significantly stronger than the bootstrapped data. The statistics of the samples compared by the statistical test are provided in columns "Actual distance" & "Bootstrapped". Mean, Median and Standard Deviation (SD) are produced for footrule distance. Footrule distance is inversely proportional to the relatedness. The result is reported for the range 7 configuration.

Statistics	**Dataset**			
	ES		**B6**	
Wilcoxon	< .001		< .001	
Test Statistics	2.0		0.0	
	Actual distance	Bootstrapped	Actual distance	Bootstrapped
Mean	88281	93281	86552	93280
Median	88104	93279	86296	93281
SD	4208.0	5.3	3134	6.6

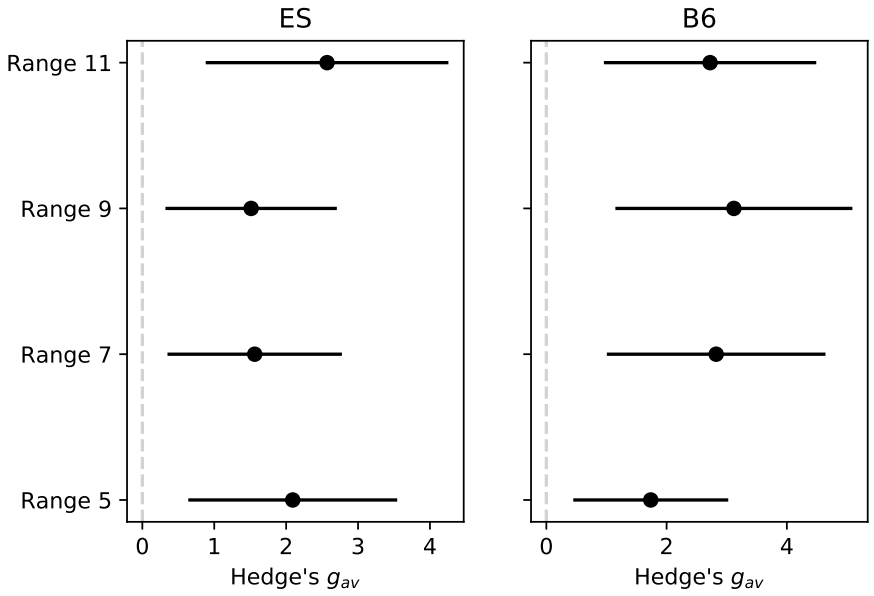


Fig. 5.8 Hedge's g_{av} effect sizes quantifying the relatedness strength of the SHAP and decision paths methods for extracting feature interactions with respect to the relatedness of the bootstrapped data. The relatedness of the feature interactions is assessed through the Footrule distances of the ranked interaction strengths. Error bars illustrate the .95 confidence intervals corrected for multiple comparisons. Ranges correspond to different configurations of the pattern extraction method.

5.3 Discussion

In the current section, I reflect on the obtained results in the same order as the experiments are conducted. Namely, a performance comparison between i) CatBoost and no-information models; ii) volume-based (VCRB) and price levels extraction methods; iii) ES and B6 financial instrument, and iv) relatedness of feature interactions obtained from SHAP and the explicit model decision paths. Furthermore, I discuss the limitations of the study as well as its broader implications and future work.

5.3.1 RQ1 - Classification Performance of VCRB Bars

Larger effect sizes for the ES instrument in Figure 5.3 mean that there is a larger improvement from using the CatBoost model for ES rather than for the B6 instrument. Based on the p-values reported in Tables 5.7 and C.3 and considering the corrected significance level for the current experiment family, the null hypothesis is rejected for configurations 5 and 7, ES instrument. The rejection of the null hypothesis means that the CatBoost estimator performs significantly better than the no-information model. From these results, I conclude

that the feature space and the model work acceptably well in some configurations. I answer positively the first research question (RQ1) for the range 5 and 7 configurations. It is worth a note that the studied setting might have even more potential if considering extensive optimisation of the system parameters.

No-information models, whose precision represents the fraction of the positively labelled patterns, are the highest for range 5 (Table C.3). A potential interpretation is that this configuration is the most suitable for the studied markets. Possible underlying reasons include lack of interest in the implied trading frequency from the larger market participants, whose capitals exceed liquidity offered at this time scale. Alternatively, it might mean that the observed performance supremacy is purely theoretical and in reality, is levelled off by higher risks associated with more frequent market exposure.

Looking at the precision of the models in Figure C.1, there is possibly a descending performance trend for both models and instruments. This can be interpreted as a gradual increase of market efficiency in the aspect of the wider use of market microstructure data for investment and trading decision making. Overall, the CatBoost model with the considered feature space gives a larger performance increase with respect to the no-information model for the ES than for the B6 dataset. This might be due to larger trading volumes of the ES instrument, hence less price action is not supported by volumes.

5.3.2 RQ2 - Comparison of VCRB and Price Level Trading

The statistical test outcomes are reported in Tables 5.8 and C.4. Considering the corrected significance level for the current experiment family, all the statistical tests in the current experiment family have a significant outcome, and the null hypothesis is rejected. Answering the research question, the results mean that VCRB patterns can be classified with significantly better performance than price level-based patterns.

All the effect sizes in Figure 5.4 are significant based on the confidence intervals not overlapping the 0-threshold. In Figure C.2 I see quite a stable gap of around 0.08 in the PR-AUC performance between VCRB and price levels for ES. The gap is smaller (around 0.04) and converges for B6.

Estimators might be worse at learning a reliable classification path from the price levels data for at least two hypothetical reasons: i) higher non-stationarity of the price level patterns in comparison to the VCRBs; ii) smaller price levels dataset sizes (Table 5.4). The latter can be potentially solved by increasing the considered time ranges of the training datasets. However, verification of any of these reasons is out of the scope of the current work and requires separate research for validation. Finally, I see a better backtesting performance for

the volume-based method in comparison to the price level in Figs 5.6, 5.7, which additionally supports the findings.

5.3.3 RQ3 - Impact of Market Liquidity on VCRB

As per my initial hypothesis (H_{13}), VCRB trading performed significantly better in the more liquid market (ES). This is shown by rejecting the null hypothesis, as the VCRB bars allow classification with significantly better performance on the ES dataset for all the considered configurations (Tables 5.9 and C.5). Looking at the model performance for both instruments in Figure C.3, one sees that the method performs generally better for the ES dataset with up to 0.04 differences of PR-AUC, and two cases where B6 has a marginally better performance. This is something expected as with more liquidity comes more impact from the volume-based features. This is also reflected in the number of POCs, with ever-increasing identified points correlating to an increase in liquidity (more recent years).

One of the potential reasons for the better performance for the ES dataset is a much larger number of extracted patterns in the ES market (Table 5.4), hence more training data is available.

Finally, I see a better backtesting performance for the more liquid asset in Figure 5.6, which supports the formal findings. Interestingly, Sharpe ratios have the same character of changes per method across configurations, especially for B6. There is less similarity between the two pattern extraction methods. While it is a known fact that most of the financial instruments are related, making it harder to diversify risks, these relations cause similar impacts on the intraday trading performance. This might suggest that using the same trading approach across instruments might not contribute to risks diversification to the expected extent, and requires careful prior research.

5.3.4 RQ4 - Feature Interaction Associations

In this chapter, I also investigate the relatedness of two different feature interaction methods. SHAP values are a widely used measure that is easy to compute and approximates the predictor. However, due to this approximation, it might be not suitable for every single setting, especially for the cases where the expected performance is far from ideal, like financial markets. Hence, I aimed at checking the relatedness of SHAP and explicitly extracted feature interactions. I designed a null hypothesis dataset using a bootstrapping approach, if the relatedness between the null dataset and the other feature analysis dataset were similar then the relatedness would be insignificant; conversely, if the relatedness was akin between the interaction analysis methods but not the null hypothesis then the null hypothesis (H_{04})

would be rejected. From the test outcomes in Tables 5.10 and C.6 I conclude that for all the configurations and both instruments the null hypothesis is rejected and relatedness between the two methods is significant. In Figure C.4 one sees that mean bootstrapped distances have a negligible variance in comparison to the actual feature interaction methods. The low variance of the null hypothesis data is a sign of the correct choice of the bootstrapped sample size since the baseline is stable.

Also, the distances of all the entries are smaller than the null hypothesis data for B6 and ES with one exception entry being slightly larger than the bootstrapped entry. Distances between the two methods are around 5% smaller on average than the null hypothesis datasets.

SHAP assigns a local explanation based on introducing a numeric measure of credit to each input feature. Then by combining many local explanations, it represents global structure while retaining local faithfulness to the original model [180]. This is in contrast to the used explicit approach which instead computes global values, accounting for the large mean distance. Hence, the varying principles underlying these approaches contribute to the observed results.

5.3.5 Limitations

I would like to highlight that the analysis provided in this chapter is but one of the possible means to empirically answer the stated research questions. Whilst I choose to focus on strict statistical analysis to showcase the significance of the results, other valid approaches could be implemented. A limitation of the conducted work is that even though the analysis is extensive, it focuses on only two examples of trading instruments. Whilst these are chosen to represent different markets that allow observing the performance of the proposed method in vastly different scenarios, one could envision that a more thorough systematic analysis of the same approach across varied instruments could have some benefits; although I argue this may be out of the scope of the current work. One potential limitation of the analysis resides in the relatively small sample size. To reject the null hypotheses I made use of the Wilcoxon test. The same as in Chapter 4, this decision was guided by the small sample size, which in turn made it not feasible to reliably establish whether the data was normally distributed, consequently I could not use parametric tests. Whilst sometimes less precise, this decision is supported by literature to be the optimum choice in these circumstances [181].

When assessing the absolute model performance, I consider the theoretical profitability threshold computed for the simplistic strategy proposed in Chapter 4, Section 4.1.4. Concretely, assuming that the take-profit is 15 ticks, and the stop-loss is 3 ticks, I count 0.5 ticks for the trading fees (which is above a typical trading fee at the beginning of 2021).

Hence, for each entry, I have a maximum possible theoretical profit of 14.5 ticks and a loss of 3.5 ticks. Dividing one by another I get the theoretical profitability threshold at 24.1% precision. Being more conservative, I would want to account for the bid-ask spread, which is 1 tick most of the time for ES and B6 (less stable). The presence of the spread means that after one enters the market, their open P&L (profit & loss) is -1 tick - if the position is opened by bid, it will be liquidated by ask which is 1 tick away and vice versa. Of course, it affects the fraction of the trades closed by the stop loss in a live setting in comparison to the no-spread simulation. There is not enough information available to probabilistically model this, but if I account for the spread by subtracting 1 tick from all trades, I end up with the profitability threshold of 33.3% precision. While the original take profit to stop loss sizes relate as 1 to 5, the actual picture (after taking into account all the mechanics and fees) is very different. In the considered setting, the limitation of the profitability threshold value comes from multiple entries observed within a short time range. With an already open position, the assumed strategy does not make use of the following signals until the current position gets liquidated. Also, there is a limitation caused by order queues hampering the strategy performance in the live market - potentially profitable orders are more likely to not be executed, as one expects the price to reverse. Losing positions will be executed always as the price continues its movement.

I note that the different instruments and datasets will involve differences in volatility and liquidity, which impacts the length of the trades and other factors which may influence backtesting results. Considering all these, I conclude that it is necessary to take the backtesting results with a certain grain of salt.

While in most other machine learning contexts the obtained performance may seem low if not horrendous, in the current domain it is in line with expectations as shown in previous works [169].

It should be noted that by design the hypotheses are tested on market microstructure-based feature space. By rejecting the null hypotheses one cannot claim that these findings hold for an arbitrary feature space, but rather for the proposed one. By running the experiments with the feature selection and model optimisation steps, I make an informal effort to expand the findings to a flexible feature set and a model configuration. Also, even though I have made the best effort to unify the experiment design, the feature spaces slightly vary between the two methods, which may have some impact on the results.

Similar test statistics numbers across Tables 5.7-5.9 result from the small group sizes and similar pairwise measurement variable comparison outcomes.

Performance-wise, the current study's trading performance resulted in Sharpe ratios of around 2 for the best-performing configurations. The obtained performance can be considered at least comparable to the studies mentioned in 4.3.5.

5.3.6 Implications for practitioners

The current chapter proposes an approach to the classification of micro-events in multidimensional non-stationary financial time series. This work advances the body of knowledge in applied financial time series analysis by proposing a pattern extraction method and shaping the contexts in which it can be used. The proposed method can be directly applied as a part of an algorithmic trading pipeline. Moreover, the method might become an alternative way of sampling the market and stand in a row with other types of sampling, like volume and range bars.

There are other research areas dealing with a similar setting, like social networks and forums analysis, topic detection and tracking, fraud detection, etc. I believe that the idea of the proposed method is applicable to some of these fields. Namely, identification of a micro-event in one of the time series dimensions as an "anchor" for the multidimensional pattern extraction. It is likely that for obtaining optimal results, the design of the anchor should involve domain knowledge.

5.3.7 Future work

I see a number of paths for future work. Namely, an extension of the experiments to other markets, more advanced backtesting, fundamental assessment of the stationarity, and extension of the feature space. Other financial markets, like Forex, Crypto and stocks would require certain adjustments of the proposed method since there are multiple marketplaces and volumes are distributed across them. Moreover, prices might also differ between the marketplaces, making it harder to aggregate the data. Overall, each financial instrument has its own characteristic properties and it would be interesting to see how generalisable the proposed method is.

While the used backtesting engine is relatively simple, there are more advanced ones that exist in the field. However, they are usually made available as parts of trading platforms. Since the trading platforms are usually proprietary, the mechanics of the backtesting engines are not always clear and transparent and cannot be replicated outside the platform. Hence, it would be beneficial to develop an open-source package for backtesting which includes bid-ask spreads and models the trading queues.

The current study is based on empirical methods and is considered application-oriented. I hypothesise that the proposed method allows extracting patterns that are more stationary than the market itself. However, I never formally measure the stationarity, to avoid further complications of the study design. A formal assessment of the stationarity would allow choosing the most promising pattern extraction methods which in theory might require fewer training entries for successful classification and trading.

In the current work, I used volumes-based feature space. There is a different approach to feature design - technical indicators, like RSI, MACD, Parabolic SAR, etc. To fully incorporate the indicators into the pipeline, the feature space should be increased significantly. This might be done in parallel with the more fine-grained configuration of the estimator and the development of a more realistic trading strategy. Even though this point is rather implementation-focused, it might lead to very interesting results, which could be further supported by the statistical approach followed in the current chapter.

5.4 Chapter conclusion

In this chapter, I present a new automated trading micro-event extraction method suitable for ML called Volume-Centred Range Bars (VCRB). The work presents a detailed statistical analysis of the presented approach to thoroughly assess its performance.

I firstly assess the volume-based pattern extraction validity by evaluating 1) the significance of the classification performance, 2) the improvement of the proposed feature space and 3) model configurations with respect to the baseline (RQ1). This expands beyond simply trading using VCRBs as I showcase how performance can be improved using a state-of-the-art feature engineering approach and a machine learning estimator. I further investigate the method's effectiveness by comparing it with another successful pattern extraction method based on price levels. The results showcase a net improvement in performance across two different financial instruments (RQ2). By rejecting the null hypothesis H_{03}, I answer positively on research question 3 (RQ3), that liquid markets improve the effectiveness of the proposed approach.

Additionally, contributing to the explainability, I compare two different feature interaction extraction approaches - the popular ML approach SHAP, which approximates the model, and an extension of Monoforest, which uses explicit decision paths from the model. The analysis shows that in the considered setting, both methods are significantly related, hence holding some common findings. I conclude that SHAP can be used for providing explainability in the considered setting, something which had previously not been investigated.

To conclude, the proposed methodology in the current and previous chapters is structured in a way that allows for comparability across studies by providing the effect sizes; and reproducibility, by detailing the method and sharing the reproducibility package. My hope is that this will make it easier for the practitioners to test this same approach and evaluate it against other methods, hopefully helping improve the field for the better. Code to reproduce the conducted analysis and match the results is available online [178, 179].

Chapter 6

Conclusion & final considerations

Reiterating the original objectives:
" i. Define the micro-events in the context of online Q&A communities and financial markets; ii. Design a generalisable methodology allowing to work with weakly manifested micro-events in both domains; iii. Detect and classify the micro-events in forum-like communications data on the example of StackOverflow Q&A platform and financial time series; iv. Investigate the reliability of SHAP model explanation in the context of weakly manifested micro-events; v. Detail the outcomes and limitations of the work."

In the current thesis, I have introduced the concept of micro-events and proposed a generalisable methodology for their detection and classification. The latter allowed me to demonstrate how micro-events can be defined in the domains of textual communications and financial time series. Then, I have demonstrated that SHAP model explanations can be reliably used in the context of the micro-events. Finally, for each domain, I have outlined the outcomes, limitations, and future work of the conducted studies. At this point I conclude that the odjectives of the dissertation were successfully achieved, making a significant contribution to the body of knowledge of micro-event detection and time series analysis [182–184]. The purpose of the current chapter is to discuss the findings of the dissertation in a more general context. This includes the potential impact of the findings on the field, identified fundamental limitations of the conducted studies, as well as potential future work. I finish the chapter with concluding remarks.

The current chapter is structured as follows: I discuss the obtained results in Section 6.1, describe the future work in the area of micro-event detection in Section 6.3, list the limitations of the pursued approach in Section 6.2, and make the concluding remarks in Section 6.4.

6.1 Discussion

In the dissertation, I set up a model experiment aimed at detecting micro-events in textual data with an assumption that these micro-events might impact the online community. I demonstrate that it is feasible to design statistically significant models for detecting the micro-events. The experiment design involves defining the impact time window, engineering the data representation, assessing the feature effect sizes, as well as training analysing linear and nonlinear models. I also propose an approach to financial time series analysis, aimed at extraction and classification of the micro-events. While the proposed approach is one of many ways of looking at the financial time series data, it is supported by financial markets mechanics, therefore, gives a fresh perspective of the analysis. Overall, I propose a way of analysing non-stationary data with a low signal-to-noise ratio in a consistent, comparable and reproducible way.

The quality of the fitted models in the previous chapters is far from the theoretical performance cap. On the one hand, it means that there is still a lot to be done in the field. On the other hand, the considered data is constantly changing which makes the advancement very challenging.

Financial time series are generated to a large extent as a result of multiple parties exploiting the predictable aspects (or inefficiencies) of the market. Being exploited, the predictable traits vanish, however, each market activity might lead to a new inefficiency. As a result, we observe a non-stationary self-regulated system, which can be predicted only to a certain extent. Its change is largely supported by a constantly changing number of interacting parties and their objectives, as well as the advancement of analytical methods. These make financial markets an infinite source of challenging labelled data.

Considering the textual communication data, it also constantly emerges. The changes are caused by the communicating parties, conditioned by the non-linearity and diversity of the biological neural networks - brains. The communicating parties are affected by various external events, which form their lives. These events might either contribute to the general evolvement of the party's personality or be directly related to the communication topic. While there is no direct competition in predicting the change in the data, there are too many factors causing it, and the current body of knowledge is far from accurately identifying and modelling all the impacts.

Being driven by completely different objectives of the parties, the two domains have the same underlying source of non-stationarity - humans. One might hypothesise that the amount of non-stationarity in the society-related data is conditioned by the engagement of the individuals, the amount of freedom they have, a varying amount of interacting agents, as well as the objective of the underlying process. Certain crowd behaviours can be suc-

cessfully modelled and predicted [185, 186]. While the quality of such models is sufficient for many real-world applications, it is quite application-specific and often far from being perfect. Involving a smaller fraction of the community, micro-events are harder to model and analyse. I believe that the current work gives a realistic assessment of the capabilities of the available methods in detecting micro-events. Based on the performance of the models, one might conclude that detecting or predicting micro-events is a principally different task than predicting the behaviour of the crowd.

6.1.1 What I have learnt from the dissertation

- The outcomes of the work allowed us to appreciate the level of complexity of the micro-event detection. I initially hypothesised that this is a higher-risk research topic, and the results of the dissertation clearly support it. One should take into account that the studied communities are fairly open and the communications are completely public. There is no access management based on reputation, as it is usually implemented in dark web communities. Consequently, one would expect the challenge of micro-event detection in dark web forums even more challenging and require a targeted approach. For instance, a work by Pastrana [187] demonstrates how eWhoring can be detected and measured. The measurement relies on assumptions on agent interactions and data quality. Relating the study to the current thesis, one should note that eWhoring often requires a single unit of text for detection, making it stronger-manifested than micro-events considered in the dissertation.

- I have observed the importance of the carefully-designed methodology, allowing to evaluate the success of the study in different dimensions, including the quality of the designed features, the significance of the model fit and feature contributions, the significance of the model performance, and, finally, statistical assessment of the null hypothesis rejection criteria. While shortcuts might be acceptable for stronger-manifested and easier-classifiable events and data, the studied matter requires all the tools to reverse-engineer the experiment design and identify its weaker points.

- While in the current thesis, I aimed to understand the feasibility of a generalisable approach, it naturally limited the final performance of the solution. When developing an applied system for micro-event detection, I would take a targeted approach and evaluate it against the provided generalised solution. This would involve making the best use of the domain knowledge as well as making use of all the available data. For instance, for the textual data, it would mean developing a representation of the community members, their expertise, relationships and contributions, and including

it into the input features. Considering time series data, one could include news, time series of the related markets, as well as order book data into the analysis. It would substantially increase the amount of processed data, the complexity of the final model, and potentially its performance. I conclude that with the currently available analytical methods, the generalised approach for micro-event detection is rather infeasible.

In terms of the broader implications of the current work, I envision that it creates a baseline for micro-event detection not only in terms of model performance but also from the perspective of methodology and rigour. As I stated above, micro-event detection requires a set of specific sensitive tools and methods allowing careful analysis of the experiment design. I hope that the current work will serve as a foundation and a guide for the coming studies on micro-event detection in a wide range of domains.

6.2 Limitations

While the idea of treating the points of interest as events is very broad, it still has certain boundaries. It implicitly focuses on the points of interest, limiting the use of indirect methods, where the points of interest are not clearly defined or even cannot be defined. It is evident when the event detection is compared to topic tracking - the emergence of the new topics is expected at any point of the analysis. Moreover, the process of the topic emergence might be smooth, represented by a gradual transformation or merge of the existing topics, where the particular event point is not present. When one tries to assess this process as an event detection task, the event's temporal boundaries would be largely indefinite. In fact, the change might be inherent in the system's nature. For example, bifurcations in complex systems, where the qualitative change of the system is caused by a gradual change of the certain system parameter values [188]. Relating the matter to the studied systems, the example of a bifurcation might be observed when the market dynamics experiences large changes after a certain number of agents is reached. In terms of textual communications, at some point of the platform popularity, an average weight time before a question is answered might have a sudden drop as more and more people start competing for platform reputation.

Considering the studied domains, in both cases, the interacting parties are represented by highly non-linear agents - humans. Moreover, both domains are known to be successfully studied from the perspective of bifurcation theory [189–191]. While the qualitative change of the system is detectable, its causes are at least difficult to explore by means of the event detection formalism. In terms of predictions - it would require identifying the critical parameters of the system causing the bifurcations, which is at least a very challenging task considering the complexity of the systems. Finally, even when one is not aiming to explicitly

detect or predict bifurcations, they are still present in the systems and might act as a source of noise when detecting or predicting other types of events.

Discussing the financial markets, it is worth mentioning other approaches to define points of interest. Namely, there is a term called alpha, which is defined as a trading signal which potentially adds value to the financial portfolio [192]. The trading signal may be represented as a rule of sampling points of interest from the time series. It can be based on domain knowledge or be completely abstract. The latter can be mined in an automatic way [193]. Of course, one sees parallels between the points of interest sampled by the alpha and the micro-events studied in the thesis - the points of interest might be called events and alphas might be seen as the event extraction methods.

When extracting the events from financial time series, I use domain knowledge. The boundaries introduced by this domain-specific view, limit the flexibility of the method to some extent. In this sense, abstract alpha mining gives a much broader space of variants. I would consider this as a limitation of the methods proposed in the thesis. However, the extraction rules can be merged. This is especially true when the number of sampled entries is large. I expect that running an automatic abstract alpha search on top of the proposed methods would create a foundation for an automatic alpha mining system backed by domain knowledge.

6.3 Future work

While the details of the future work on the individual technical chapters are provided in their corresponding sections (3.3.4, 4.3.7, 5.3.7), the current section covers the future work of the thesis as a whole.

Based on the sections above, I would see at least a couple of aspects, where the current work could be extended. Firstly, it would be very interesting to study the sources of noise of the investigated systems. There is a huge body of knowledge of studying non-linear systems, which suggests potential interdisciplinary research in the field. In particular, modelling of the noise sources might allow better synthetic data generation, as well as assessment of the stability of the newly proposed event extraction methods. Secondly, the extension of the proposed event extraction methods in financial time series might be interesting. While the proposed methods serve as a good baseline, there is definitely space for improvement of the approach. As suggested above, abstract trading signal extraction can be revised from the perspective of integration of the domain knowledge into it through the components suggested in the current work.

When one is considering a wider perspective of the event detection topic, there is currently no method that would allow robust event detection in a domain-agnostic way. However, being aware of the pronounced heterogeneity in the field, one can consider aspects of the analysis that could be generalised. For instance, explainability of the models in the context of event detection. We already know that the effect sizes in the field are medium at most. Hence, models fitted on the considered data might require adjustments of the explainability methods. I have already studied feature interactions reliability for event detection using SHAP. However, due to smaller effect sizes and a large number of contributing factors, there might be a need for more fundamental research of applicability of the surrogate model-based methods like SHAP or LIME to event detection. Finally, even though it is a general challenge, the field would benefit from development of new direct model explainability methods that are not limited by linear models.

In the dissertation, I have covered two domains. If I would have started from scratch, I would have focused on a single domain and after developing the generalisable solution as a baseline, would have aimed to develop a targeted approach. This would have potentially allowed me to propose an end-to-end solution to a real-world problem. Developing the targeted solution, I would focus on the data representation and incorporating the online community structure in the final model. This would apply both for time series analysis and online communications data. Going further, I would model agent interactions in the system, incorporating the body of knowledge of the agent-based modelling research field, and especially agent-based modelling of ML-based agents.

6.4 Concluding remarks

Considering the current development of the event-detection methods and approaches, in the current work I have begun to scratch the surface of the unknown in the area of micro-event detection. This is a broad topic with potential applications in security, well-being support, communication quality enhancement, etc. I believe that with further improvement of the analytical tools, its impact will only grow. The reported statistically-backed findings suggest that there are ways of detecting and classifying the micro-events using the already available tools. However, the performance of the models is yet to be improved. Considering the current machine learning advancement pace, it becomes clear that new findings in the field are not so far away. I hope that the current work will help the research community by providing a sound methodology and baseline performance in the context of micro-event detection and classification.

References

[1] Jiaying Liu, Xiangjie Kong, Feng Xia, Xiaomei Bai, Lei Wang, Qing Qing, and Ivan Lee. Artificial intelligence in the 21st century. *IEEE Access*, 6:34403–34421, 2018.

[2] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2334–2346, 2017.

[3] Liav Sade-Beck. Internet ethnography: Online and offline. *International Journal of Qualitative Methods*, 3(2):45–51, 2004.

[4] Artur Sokolovsky, David Hare, and Jorn Mehnen. Cost-effective vibration analysis through data-backed pipeline optimisation. *Sensors*, 21(19):6678, 2021.

[5] Kai Yu, Lei Jia, Yuqiang Chen, and Wei Xu. Deep learning: yesterday, today, and tomorrow. *Journal of computer Research and Development*, 50(9):1799, 2013.

[6] Marcos Lopez De Prado. *Advances in financial machine learning*. John Wiley & Sons, 2018.

[7] Ruey S Tsay. *Analysis of financial time series*, volume 543. John wiley & sons, 2005.

[8] Andrew Caines, Sergio Pastrana, Alice Hutchings, and Paula J Buttery. Automatically identifying the function and intent of posts in underground forums. *Crime Science*, 7(1):1–14, 2018.

[9] Yla Tausczik and Xiaoyun Huang. Knowledge generation and sharing in online communities: current trends and future directions. *Current opinion in psychology*, 36:60–64, 2020.

[10] Amy X Zhang, Grant Hugh, and Michael S Bernstein. Policykit: building governance in online communities. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 365–378, 2020.

[11] Farzindar Atefeh and Wael Khreich. A Survey of Techniques for Event Detection in Twitter. *Computational Intelligence*, 31(1):132–164, 2 2015.

[12] Nikolaos Panagiotou, Ioannis Katakis, and Dimitrios Gunopulos. Detecting events in online social networks: Definitions, trends and challenges. In *Solving Large Scale Learning Tasks. Challenges and Algorithms*, pages 42–84. Springer, 2016.

[13] Maarten Sukel, Stevan Rudinac, and Marcel Worring. Multimodal classification of urban micro-events. *MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia*, pages 1455–1463, 2019.

[14] Baidyanath Biswas, Arunabha Mukhopadhyay, Sudip Bhattacharjee, Ajay Kumar, and Dursun Delen. A text-mining based cyber-risk assessment and mitigation framework for critical analysis of online hacker forums. *Decision Support Systems*, 152:113651, 2022.

[15] Cambridge International Dictionary of English.

[16] Zafar Saeed, Rabeeh Ayaz Abbasi, Onaiza Maqbool, Abida Sadaf, Imran Razzak, Ali Daud, Naif Radi Aljohani, and Guandong Xu. What's happening around the world? a survey and framework on event detection techniques on twitter. *Journal of Grid Computing*, 17(2):279–312, 2019.

[17] Chengkun He, Jie Shao, and Jiayu Sun. An anomaly-introduced learning method for abnormal event detection. *Multimedia Tools and Applications*, 77(22):29573–29588, 2018.

[18] Quanbiao Shang, Teresa Serra, Philip Garcia, and Mindy Mallory. Looking under the surface: An analysis of iceberg orders in the us agricultural futures markets. *Agricultural Economics*, 52(4):679–699, 2021.

[19] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 4765–4774. Curran Associates, Inc., 2017.

[20] W. Enders. *Applied Econometric Times Series*. Wiley Series in Probability and Statistics. Wiley, 2014.

[21] Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *International Conference on Machine Learning*, pages 8346–8356. PMLR, 2020.

[22] James Allan, Jaime G Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic detection and tracking pilot study final report. 1998.

[23] Thong Hoang, Pei Hua Cher, Philips Kokoh Prasetyo, and Ee-Peng Lim. Crowdsensing and analyzing micro-event tweets for public transportation insights. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 2157–2166. IEEE, 12 2016.

[24] Kasthuri Jayarajah and Archan Misra. Can Instagram posts help characterize urban micro-events? *FUSION 2016 - 19th International Conference on Information Fusion, Proceedings*, pages 130–137, 2016.

[25] Maarten Sukel, Stevan Rudinac, and Marcel Worring. Multimodal classification of urban micro-events. *MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia*, pages 1455–1463, 2019.

[26] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 44(1.2):206–226, 2000.

[27] John R Koza, Forrest H Bennett, David Andre, and Martin A Keane. Automated design of both the topology and sizing of analog electrical circuits using genetic programming. In *Artificial Intelligence in Design'96*, pages 151–170. Springer, 1996.

[28] Peter Harrington. *Machine learning in action*. Simon and Schuster, 2012.

[29] Sheena Angra and Sachin Ahuja. Machine learning and its applications: A review. In *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*, pages 57–60. IEEE, 2017.

[30] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[31] Chris Chatfield. *Time-series forecasting*. Chapman and Hall/CRC, 2000.

[32] S Rassel and P Norwig. Artificial intelligence: a modern approach (aima), 2007.

[33] Frederick Mosteller and John W Tukey. Data analysis, including statistics. *Handbook of social psychology*, 2:80–203, 1968.

[34] Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning, 2018.

[35] Ribana Roscher, Bastian Bohn, Marco F Duarte, and Jochen Garcke. Explainable machine learning for scientific insights and discoveries. *Ieee Access*, 8:42200–42216, 2020.

[36] Norman Matloff. *Statistical regression and classification: from linear models to machine learning*. Chapman and Hall/CRC, 2017.

[37] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[38] David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. *Logistic regression*. Springer, 2002.

[39] Andy Field. *Discovering statistics using R*, volume 50. Sage publications, 2012.

[40] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.

[41] Mohammad Norouzi, Maxwell D Collins, Matthew Johnson, David J Fleet, and Pushmeet Kohli. Efficient non-greedy optimization of decision trees. *arXiv preprint arXiv:1511.04056*, 2015.

[42] Lior Rokach and Oded Maimon. Decision trees. In *Data mining and knowledge discovery handbook*, pages 165–192. Springer, 2005.

[43] Martin Sewell. Ensemble learning. *RN*, 11(02):1–34, 2008.

[44] Gérard Biau and Erwan Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016.

[45] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[46] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[47] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, pages 3146–3154, 2017.

[48] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Advances in neural information processing systems*, pages 6638–6648, 2018.

[49] Oliver Nelles. *Neural Networks*, pages 279–345. Springer International Publishing, Cham, 2020.

[50] Charles J Geyer. Practical markov chain monte carlo. *Statistical science*, pages 473–483, 1992.

[51] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455. PMLR, 2009.

[52] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[53] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning. *35th International Conference on Machine Learning, ICML 2018*, 5:3376–3391, 2018.

[54] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[55] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, 2003.

[56] Martin Gerlach, Tiago P. Peixoto, and Eduardo G. Altmann. A network approach to topic models. *Science Advances*, 4(7):eaaq1360, 7 2018.

[57] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. *Lecture Notes in Computer Science*, 3512:758–770, 2005.

[58] Huan Liu, Hiroshi Motoda, Rudy Setiono, and Zheng Zhao. Feature selection: An ever evolving frontier in data mining. In *Feature selection in data mining*, pages 4–13. PMLR, 2010.

[59] Naoual El Aboudi and Laila Benhlima. Review on wrapper feature selection approaches. In *2016 International Conference on Engineering & MIS (ICEMIS)*, pages 1–5. IEEE, 2016.

[60] Haoyue Liu, MengChu Zhou, and Qing Liu. An embedded feature selection method for imbalanced data classification. *IEEE/CAA Journal of Automatica Sinica*, 6(3):703–715, 2019.

[61] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[62] Max Kuhn and Kjell Johnson. *Feature engineering and selection: A practical approach for predictive models*. CRC Press, 2019.

[63] Stefan Falkner, Aaron Klein, and Frank Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. In *International Conference on Machine Learning*, pages 1437–1446. PMLR, 2018.

[64] Selmer C Larson. The shrinkage of the coefficient of multiple correlation. *Journal of Educational Psychology*, 22(1):45, 1931.

[65] Damjan Krstajic, Ljubomir J Buturovic, David E Leahy, and Simon Thomas. Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of cheminformatics*, 6(1):1–15, 2014.

[66] Christoph Bergmeir and José M Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, 2012.

[67] Maranke Wieringa. What to account for when accounting for algorithms: A systematic literature review on algorithmic accountability. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 1–18, 2020.

[68] Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. Metrics for explainable ai: Challenges and prospects. *arXiv preprint arXiv:1812.04608*, 2018.

[69] Hani Hagras. Toward human-understandable, explainable ai. *Computer*, 51(9):28–36, 2018.

[70] Derek Doran, Sarah Schulz, and Tarek R Besold. What does explainable ai really mean? a new conceptualization of perspectives. *arXiv preprint arXiv:1710.00794*, 2017.

[71] Andreas Holzinger, Chris Biemann, Constantinos S Pattichis, and Douglas B Kell. What do we need to build explainable ai systems for the medical domain? *arXiv preprint arXiv:1712.09923*, 2017.

[72] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

[73] Leo Breiman et al. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.

[74] Hugo Jair Escalante, Sergio Escalera, Isabelle Guyon, Xavier Baró, Yağmur Güçlütürk, Umut Güçlü, Marcel Van Gerven, and Rob van Lier. *Explainable and interpretable models in computer vision and machine learning*. Springer, 2018.

[75] Alexander Zien, Nicole Krämer, Sören Sonnenburg, and Gunnar Rätsch. The feature importance ranking measure. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 694–709. Springer, 2009.

[76] Yanjun Qi. Random forest for bioinformatics. In *Ensemble machine learning*, pages 307–323. Springer, 2012.

[77] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56–67, 2020.

[78] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[79] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

[80] Ken Kelley and Kristopher J Preacher. On effect size. *Psychological methods*, 17(2):137, 2012.

[81] Roger Stern. *Good statistical practice for natural resources research*. CABI, 2004.

[82] D Hawkins, E Gallacher, M Gammell, et al. Statistical power, effect size and animal welfare: recommendations for good practice. *Animal Welfare*, 22(3):339–344, 2013.

[83] Leland Wilkinson. Statistical methods in psychology journals: Guidelines and explanations. *American psychologist*, 54(8):594, 1999.

[84] Wilhelm Kirch. Pearson's correlation coefficient. *Encyclopedia of Public Health*, pages 1090–1091, 2008.

[85] RG Carpenter. Principles and procedures of statistics, with special reference to the biological sciences. *The Eugenics Review*, 52(3):172, 1960.

[86] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[87] Larry V Hedges. Distribution theory for glass's estimator of effect size and related estimators. *journal of Educational Statistics*, 6(2):107–128, 1981.

[88] Geoff Cumming. *Understanding the new statistics: Effect sizes, confidence intervals, and meta-analysis*. Routledge, 2013.

[89] Daniël Lakens. Calculating and reporting effect sizes to facilitate cumulative science: A practical primer for t-tests and ANOVAs. *Frontiers in Psychology*, 4(NOV), 2013.

[90] Cynthia Fraser. Association between two categorical variables: Contingency analysis with chi square. In *Business Statistics for Competitive Advantage with Excel 2019 and JMP*, pages 341–377. Springer, 2019.

[91] Norman Cliff. Dominance statistics: Ordinal analyses to answer ordinal questions. *Psychological Bulletin*, 114(3):494–509, 1993.

[92] Ronald A Fisher. *Statistical methods and scientific inference*. Hafner Publishing Co., 1956.

[93] Joseph A. Durlak. How to select, calculate, and interpret effect sizes. *Journal of Pediatric Psychology*, 34(9):917–928, 2009.

[94] Barnet Woolf. The log likelihood ratio test (the g-test). *Annals of human genetics*, 21(4):397–409, 1957.

[95] Michael R Veall and Klaus F Zimmermann. Pseudo-r2 measures for some common limited dependent variable models. *Journal of Economic surveys*, 10(3):241–259, 1996.

[96] Robert M O'brien. A caution regarding rules of thumb for variance inflation factors. *Quality & quantity*, 41(5):673–690, 2007.

[97] Michael E Miller, Siu L Hui, and William M Tierney. Validation techniques for logistic regression models. *Statistics in medicine*, 10(8):1213–1226, 1991.

[98] R Dennis Cook and Sanford Weisberg. *Applied regression including computing and graphics*, volume 488. John Wiley & Sons, 2009.

[99] R Dennis Cook and Sanford Weisberg. *Residuals and influence in regression*. New York: Chapman and Hall, 1982.

[100] Keith A McNeil, Isadore Newman, and Francis J Kelly. *Testing research hypotheses with the general linear model*. SIU Press, 1996.

[101] Student. The probable error of a mean. *Biometrika*, pages 1–25, 1908.

[102] Ellen R Girden. *ANOVA: Repeated measures*. Number 84. Sage, 1992.

[103] Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer, 1992.

[104] M Usman. On consistency and limitation of independent t-test kolmogorov smirnov test and mann whitney u test. *IOSR Journal of Mathematics*, 12(4):22–27, 2016.

[105] Carlo Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62, 1936.

[106] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[107] Zbyněk Šidák. Rectangular confidence regions for the means of multivariate normal distributions. *Journal of the American Statistical Association*, 62(318):626–633, 1967.

[108] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: Understanding microblogging usage and communities. *Joint Ninth WebKDD and First SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, pages 56–65, 2007.

[109] T Soni Madhulatha. An overview on clustering methods. *arXiv preprint arXiv:1205.1117*, 2012.

[110] Zafar Saeed, Rabeeh Ayaz Abbasi, Imran Razzak, Onaiza Maqbool, Abida Sadaf, and Guandong Xu. Enhanced heartbeat graph for emerging event detection on twitter using time series networks. *Expert Systems with Applications*, 136:115–132, 2019.

[111] Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden, and Robert C. Miller. TwitInfo: Aggregating and visualizing microblogs for event exploration. In *Conference on Human Factors in Computing Systems - Proceedings*, pages 227–236, New York, New York, USA, 2011. ACM Press.

[112] Aron Culotta. Towards detecting influenza epidemics by analyzing Twitter messages. *SOMA 2010 - Proceedings of the 1st Workshop on Social Media Analytics*, pages 115–122, 2010.

[113] Alex Lamb, Michael J. Paul, and Mark Dredze. Separating fact from fear: Tracking flu infections on twitter. *NAACL HLT 2013 - 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Main Conference*, pages 789–795, 2013.

[114] Eiji Aramaki, Sachiko Maskawa, and Mizuki Morita. Twitter catches the flu: Detecting influenza epidemics using Twitter. *EMNLP 2011 - Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1568–1576, 2011.

[115] Long Chen, Huaizhi Zhang, Joemon M Jose, Haitao Yu, Yashar Moshfeghi, and Peter Triantafillou. Topic detection and tracking on heterogeneous information. *Journal of Intelligent Information Systems*, 51(1):115–137, 2018.

[116] Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353, 2018.

[117] Paul Viola, John C. Platt, and Cha Zhang. Multiple Instance boosting for object detection. *Advances in Neural Information Processing Systems*, pages 1417–1424, 2005.

[118] Brian McFee, Justin Salamon, and Juan Pablo Bello. Adaptive Pooling Operators for Weakly Labeled Sound Event Detection. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 26(11):2180–2193, 2018.

[119] Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. *Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference*, 2009.

[120] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning. *35th International Conference on Machine Learning, ICML 2018*, 5:3376–3391, 2018.

[121] Yan Xu, Tao Mo, Qiwei Feng, Peilin Zhong, Maode Lai, and Eric I.Chao Chang. Deep learning of feature representation with multiple instance learning for medical image analysis. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 1626–1630, 2014.

[122] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. *Advances in Neural Information Processing Systems*, 2003.

[123] Rui Wang, Deyu Zhou, and Yulan He. ATM: Adversarial-neural Topic Model. *Information Processing & Management*, 56(6):102098, 11 2019.

[124] Anton Barua, Stephen W. Thomas, and Ahmed E. Hassan. What are developers talking about? An analysis of topics and trends in Stack Overflow. *Empirical Software Engineering*, 19(3):619–654, 6 2014.

[125] Xin-Li Yang, David Lo, Xin Xia, Zhi-Yuan Wan, and Jian-Ling Sun. What Security Questions Do Developers Ask? A Large-Scale Study of Stack Overflow Posts. *Journal of Computer Science and Technology*, 31(5):910–924, 9 2016.

[126] Ahmad Abdellatif, Diego Costa, Khaled Badran, Rabe Abdalkareem, and Emad Shihab. Challenges in chatbot development: A study of stack overflow posts. In *Proceedings of the 17th International Conference on Mining Software Repositories*, pages 174–185, 2020.

[127] Stylianos I Vagropoulos, GI Chouliaras, Evaggelos G Kardakos, Christos K Simoglou, and Anastasios G Bakirtzis. Comparison of sarimax, sarima, modified sarima and ann-based models for short-term pv generation forecasting. In *2016 IEEE International Energy Conference (ENERGYCON)*, pages 1–6. IEEE, 2016.

[128] Prapanna Mondal, Labani Shit, and Saptarsi Goswami. Study of effectiveness of time series modeling (arima) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*, 4(2):13, 2014.

[129] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90:106181, 2020.

[130] Michael C Münnix, Takashi Shimada, Rudi Schäfer, Francois Leyvraz, Thomas H Seligman, Thomas Guhr, and H Eugene Stanley. Identifying states of a financial market. *Scientific reports*, 2:644, 2012.

[131] Ash Booth, Enrico Gerding, and Frank Mcgroarty. Automated trading with performance weighted random forests and seasonality. *Expert Systems with Applications*, 41(8):3651–3661, 2014.

[132] Andreas Mühlbacher and Thomas Guhr. Credit risk meets random matrices: Coping with non-stationary asset correlations. *Risks*, 6(2):42, 2018.

[133] Desislava Chetalova. *Dependencies and non-stationarity in financial time series*. PhD thesis, Duisburg, Essen, 2015.

[134] Pan Du, Warren A Kibbe, and Simon M Lin. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics*, 22(17):2059–2065, 2006.

[135] Mohsen Bahmani-Oskooee and Scott W Hegerty. Exchange rate volatility and trade flows: a review article. *Journal of Economic studies*, 2007.

[136] Carey Caginalp and Gunduz Caginalp. Asset price volatility and price extrema. *arXiv preprint arXiv:1802.04774*, 2018.

[137] Nikolay Miller, Yiming Yang, Bruce Sun, and Guoyi Zhang. Identification of technical analysis patterns with smoothing splines for bitcoin prices. *Journal of Applied Statistics*, 46(12):2289–2297, 2019.

[138] Robert L Kissell. *The science of algorithmic trading and portfolio management*. Academic Press, 2013.

[139] Luc Bauwens, Pierre Giot, Joachim Grammig, and David Veredas. A comparison of financial duration models via density forecasts. *International Journal of Forecasting*, 20(4):589–609, 2004.

[140] Joachim Grammig and Marc Wellner. Modeling the interdependence of volatility and inter-transaction duration processes. *Journal of Econometrics*, 106(2):369–400, 2002.

[141] Simone Manganelli. Duration, volume and volatility impact of trades. *Journal of Financial markets*, 8(4):377–399, 2005.

[142] Alfonso Dufour and Robert F Engle. Time and the price impact of a trade. *The Journal of Finance*, 55(6):2467–2498, 2000.

[143] Michael Aitken and Alex Frino. The determinants of market bid ask spreads on the australian stock exchange: Cross-sectional analysis. *Accounting & Finance*, 36(1):51–63, 1996.

[144] David Easley, Marcos Lopez de Prado, and Maureen O'Hara. Discerning information from trade data. *Journal of Financial Economics*, 120(2):269–285, 2016.

[145] Michael AH Dempster and Vasco Leemans. An automated fx trading system using adaptive reinforcement learning. *Expert Systems with Applications*, 30(3):543–552, 2006.

[146] Andrew W. Lo. The Statistics of Sharpe Ratios. *Financial Analysts Journal*, 58(4):36–52, 7 2002.

[147] Anonymous. Machine learning-based detection of FLOSS version release events from Stack Overflow message data, October 2020.

[148] Artur Sokolovsky, Jaume Bacardit, and Thomas Gross. Machine learning-based detection of FLOSS version release events from Stack Overflow message data, 6 2020.

[149] Jeremy Katz. Libraries.io Open Source Repository and Dependency Metadata, 12 2018.

[150] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc., 2009.

[151] C. J. Hutto and Eric Gilbert. VADER: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*, pages 216–225, 2014.

[152] Radim Rehurek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, 2010.

[153] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 2018-Decem:6638–6648, 2018.

[154] David C. Hoaglin, Boris Iglewicz, and John W. Tukey. Performance of Some Resistant Rules for Outlier Labeling. *Journal of the American Statistical Association*, 81(396):991, 1986.

[155] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[156] Daniel McFadden. Quantitative Methods for Analyzing Travel Behaviour of Individuals: Some Recent Developments. *Behavioural Travel Modelling*, pages 279–318, 1978.

[157] Robbert Jongeling, Proshanta Sarkar, Subhajit Datta, and Alexander Serebrenik. On negative results when using sentiment analysis tools for software engineering research. *Empirical Software Engineering*, 22(5):2543–2584, 2017.

[158] Bin Lin, Fiorella Zampetti, Massimiliano Di Penta, Rocco Oliveto, Gabriele Bavota, and Michele Lanza. Sentiment Analysis for Sooware Engineering: How Far Can We Go? *Icse*, 2018.

[159] Yuan Tian and David Lo. A comparative study on the effectiveness of part-of-speech tagging techniques on bug reports. *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering, SANER 2015 - Proceedings*, pages 570–574, 2015.

[160] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[161] Fabio Calefato, Filippo Lanubile, Federico Maiorano, and Nicole Novielli. Sentiment polarity detection for software development. *Empirical Software Engineering*, 23(3):1352–1382, 2018.

[162] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. *arXiv preprint arXiv:1207.4169*, 2012.

[163] Xinli Yang, David Lo, Li Li, Xin Xia, Tegawendé F. Bissyandé, and Jacques Klein. Characterizing malicious Android apps by mining topic-specific data flow signatures. *Information and Software Technology*, 90:27–39, 2017.

[164] Kai Tian, Meghan Revelle, and Denys Poshyvanyk. Using latent dirichlet allocation for automatic categorization of software. *Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories, MSR 2009*, pages 163–166, 2009.

[165] Wenpeng Yin and Hinrich Schütze. Multichannel variable-size convolution for sentence classification. *CoNLL 2015 - 19th Conference on Computational Natural Language Learning, Proceedings*, pages 204–214, 2015.

[166] Anonymous. Machine learning classification of price extrema based on market microstructure features. a case study of s&p 500 e-mini futures., 2020.

[167] Artur Sokolovsky and Luca Arnaboldi. Machine learning classification of price extrema based on market microstructure and price action features. a case study of s&p500 e-mini futures. reproducibility package, May 2021.

[168] Giulia Iori and Carl Chiarella. A Simulation Analysis of the Microstructure of Double Auction Markets. *Quantitative Finance*, 2:346–353, 2002.

[169] Matthew Dixon, Diego Klabjan, and Jin Hoon Bang. Classification-based financial markets prediction using deep neural networks. *Algorithmic Finance*, 6(3-4):67–77, 12 2017.

[170] Zhuoran Xiong, Xiao-Yang Liu, Shan Zhong, Hongyang Yang, and Anwar Walid. Practical deep reinforcement learning approach for stock trading. *arXiv preprint arXiv:1811.07522*, 2018.

[171] Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. Deep reinforcement learning for automated stock trading: An ensemble strategy. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–8, 2020.

[172] Yao Qin, Dongjin Song, Haifeng Cheng, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. *IJCAI International Joint Conference on Artificial Intelligence*, 0:2627–2633, 2017.

[173] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 2018-Decem:6638–6648, 2018.

[174] Igor Kuralenok, Vasilii Ershov, and Igor Labutin. Monoforest framework for tree ensemble analysis. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 13780–13789. Curran Associates, Inc., 2019.

[175] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*, 10(3), 2015.

[176] Frank Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80, 1945.

[177] C. Spearman. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, 15(1):72, 1904.

[178] Artur Sokolovsky, Luca Arnaboldi, Jaume Bacardit, and Thomas Gross. Explainable ML-driven Strategy for Automated Trading Pattern Extraction - Reproducibility Package, March 2021.

[179] Artur Sokolovsky, Luca Arnaboldi, Jaume Bacardit, and Thomas Gross. Explainable ML-driven Strategy for Automated Trading Pattern Extraction - Reproducibility Package, March 2021.

[180] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020.

[181] Eva Skovlund and Grete U Fenstad. Should we always choose a non parametric test when comparing two apparently non normal distributions? *Journal of clinical epidemiology*, 54(1):86–92, 2001.

[182] Artur Sokolovsky, Thomas Gross, and Jaume Bacardit. Is it feasible to detect floss version release events from textual messages? a case study on stack overflow. *PloS one*, 16(2):e0246464, 2021.

[183] Artur Sokolovsky, Luca Arnaboldi, Jaume Bacardit, and Thomas Gross. Explainable machine learning-driven strategy for automated trading pattern extraction, 2021.

[184] Artur Sokolovsky and Luca Arnaboldi. Machine learning classification of price extrema based on market microstructure and price action features. a case study of s&p500 e-mini futures, 2021.

[185] M Sami Zitouni, Harish Bhaskar, J Dias, and Mohammed E Al-Mualla. Advances and trends in visual crowd analysis: A systematic survey and evaluation of crowd modelling techniques. *Neurocomputing*, 186:139–159, 2016.

[186] HY Swathi, G Shivakumar, and HS Mohana. Crowd behavior analysis: A survey. In *2017 international conference on recent advances in electronics and communication technology (ICRAECT)*, pages 169–178. IEEE, 2017.

[187] Sergio Pastrana, Alice Hutchings, Daniel Thomas, and Juan Tapiador. Measuring ewhoring. In *Proceedings of the Internet Measurement Conference*, pages 463–477, 2019.

[188] P Blanchard, RL Devaney, and GR Hall. Differential equations. london: Thompson. Technical report, ISBN 0-495-01265-3, 2006.

[189] Carl Chiarella, Xue-Zhong He, Duo Wang, and Min Zheng. The stochastic bifurcation behaviour of speculative financial markets. *Physica A: Statistical Mechanics and its Applications*, 387(15):3837–3846, 2008.

[190] Andreas C Tsoumanis, Constantinos I Siettos, George V Bafas, and Ioannis G Kevrekidis. Equation-free multiscale computations in social networks: from agent-based modeling to coarse-grained stability and bifurcation analysis. *International Journal of Bifurcation and Chaos*, 20(11):3673–3688, 2010.

[191] Darren Pais, Carlos H Caicedo-Nunez, and Naomi E Leonard. Hopf bifurcations and limit cycles in evolutionary network dynamics. *SIAM Journal on Applied Dynamical Systems*, 11(4):1754–1784, 2012.

[192] Igor Tulchinsky. *Finding Alphas: A quantitative approach to building trading strategies*. John Wiley & Sons, 2019.

[193] Tianping Zhang, Yuanqi Li, Yifei Jin, and Jian Li. Autoalpha: an efficient hierarchical evolutionary algorithm for mining alpha factors in quantitative investment, 2020.

[194] Paul Jaccard coefficient: Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37, 1901.

[195] Ke Wang and Xiaojun Wan. Sentigan: Generating sentimental texts via mixture adversarial networks. *IJCAI International Joint Conference on Artificial Intelligence*, 2018-July:4446–4452, 2018.

[196] Fernando Pereira, Naftali Tishby, and Lillian Lee. Distributional clustering of english words. *arXiv preprint cmp-lg/9408011*, 1994.

[197] Andrew McCallum and Kamal Nigam. Employing EM and Pool-Based Active Learning for Text Classification. *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 350–358, 1998.

[198] Markus Becker and Miles Osborne. A two-stage method for active learning of statistical grammars. *IJCAI International Joint Conference on Artificial Intelligence*, pages 991–996, 2005.

Appendix A

Supplementary materials for Chapter 3

A.1 The goodness of fit measures of Logistic Regression models

In the current appendix, I provide detailed information on the goodness of fit for the Logistic Regression models (Tables A.1 and A.2). Tjur R^2 and Adjusted McFadden (Adj. MF) R^2 are used to choose the best-fitted model whose detailed analysis is described in the paper. Event-based multiple packages datasets are not considered due to violation of the independence condition of the Logistic Regression model. Based on the goodness of fit, I choose the Selenium package, minor updates, event-based time steps dataset. Not all the statistical tools are meant to be used with the numbers of features considered in the Table A.2 - Tjur R^2 does not fully adjust to the larger numbers of features, leading to erroneous results, hence adjusted McFadden R^2 is a more appropriate measure for the hSBM feature space interpretation.

A.2 Performance of estimators

The appendix provides detailed results of the Chapter 3 experiments using various performance metrics and the permutation test (Tables A.3,A.4). Due to the class imbalance, I report mean values over two cases - events treated as a positive and a negative class. The exception is the ROC-AUC metric which is computed with events as a positive class - its mean is always .5 by definition.

Table A.1 Logistic Regression models, LDA: goodness of fit. "c.w.-based" values correspond to the calendar week-based time step datasets. The Number of Features column corresponds to the number of features in the model after the RFECV feature selection step. Significant model fits based on the Log-likelihood Ratio test are marked with a star(*).

	AIC	LLR Test	Tjur R^2	Adj. McFadden R^2	Number of Features
Multiple major event-based	117.27	.17	.20	-0.16	18
Multiple minor event-based	306.04	<.001*	.40	.27	10
Multiple patch event-based	423.54	<.001*	.23	.18	1
Django minor event-based	311.03	<.001*	.07	.05	4
Django patch event-based	130.02	.10	.02	-0.03	2
Selenium minor event-based	320.96	<.001*	.16	.07	12
Selenium patch event-based	135.06	.17	.00	-0.03	1
Multiple major c.w.-based	165.83	.62	.06	-0.17	18
Multiple minor c.w.-based	457.04	<.001*	.13	.04	15
Multiple patch c.w.-based	480.45	<.001*	.13	.03	17
Django minor c.w.-based	200.83	.01	.05	.01	4
Django patch c.w.-based	95.57	<.001*	.13	.05	5
Selenium minor c.w.-based	306.02	<.001*	.10	.05	8
Selenium patch c.w.-based	151.27	.36	.04	-0.12	14

Table A.2 Logistic Regression models, hSBM: goodness of fit. "c.w.-based" values correspond to the calendar week-based time step datasets. The Number of Features column corresponds to the number of features in the model after the RFECV feature selection step. Significant model fits based on the Log-likelihood Ratio test are marked with a star(*).

	AIC	LLR Test	Tjur R^2	Adj. McFadden R^2	Number of Features
Multiple major event-based	246.00	.90	1.00	-1.41	675
Multiple minor event-based	642.09	.004	.99	-0.52	407
Multiple patch event-based	484.17	<.001*	.09	.06	2
Django minor event-based	270.00	<.001*	1.00	.21	139
Django patch event-based	636.00	1.00	1.00	-3.81	608
Selenium minor event-based	345.82	.05	.01	-0.01	2
Selenium patch event-based	131.00	.01	.02	-0.00	2
Multiple major c.w.-based	153.46	.02	.05	-0.00	5
Multiple minor c.w.-based	294.09	<.001*	.72	.39	72
Multiple patch c.w.-based	500.97	.01	.03	.00	5
Django minor c.w.-based	728.00	1.00	1.00	-2.44	474
Django patch c.w.-based	85.25	<.001*	.20	.22	5
Selenium minor c.w.-based	248.35	<.001*	.68	.23	72
Selenium patch c.w.-based	710.00	1.00	1.00	-4.00	407

A.3 Synthetic data generator optimisation

To ensure maximum similarity between the synthetic and real-world datasets, the softmax temperature and top_k parameters of the fine-tuned GPT-2 generator are optimized. These parameters affect the properties of the generator output distribution.

The following two metrics are sequentially used to assess the similarity between message corpora:

1. Pairwise Jaccard similarity [194] between the posts, which is used in Natural Language Processing in different variations [195];

2. Kullback-Leibler divergence is used for quantitative comparison of the distance distributions, as a well-established method for distributions comparison [196–198].

Since it is not feasible to compute the pairwise distances between all individual messages, I take random samples of 500 posts from the data and repeat this sampling 30 times to get the standard deviations of the result. I also make sure that further increase of the random sample size does not change the optimization outcome.

To my knowledge, there is no unified framework for assessing the quality of the synthetic text. However, it is common to consider such measures as Fluency, Novelty, Diversity and Intelligibility of the generated entries [195]. Generating the data for automated processing purposes, I have taken two properties - Novelty and Diversity, which are obtained for the synthetic and real-world datasets.

Based on the equations in [195], Novelty defines the distance between the synthetically generated and the real-world datasets. Diversity of the dataset can be measured as novelty between two samples of the same dataset. Fluency and Intelligibility are more subtle measures and cannot be directly measured from the defined metrics. Aiming to make the approach simple and universal, I limit the synthetic data assessment to the 2 measures.

With this understanding, I apply the measures to the considered setting:

- Novelty: a sample of the synthetic data is assessed against the sample of the real-world data. The metrics value should be as small as possible. Ideally, it should be equal to the Diversity of the real-world dataset.

- Diversity: a data sample is assessed against a different sample within the same dataset. The metrics values should be as similar as possible for the real world and synthetic data.

The optimization process is performed separately for the event-related and background messages. Then, a single set of parameters is chosen to generate all the messages in order to

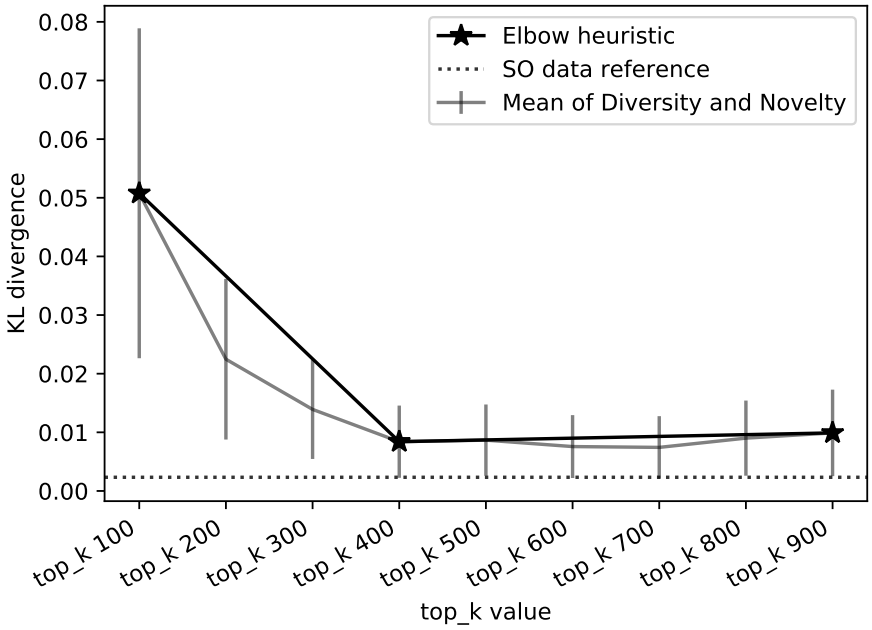


Fig. A.1 Synthetic data generator optimisation. The mean values of SO and synthetic data are computed across background and event messages. The error bars represent the standard deviations over 30 random samples.

preserve the consistency of the dataset and avoid the process of differentiating between the positive and negative entries becoming trivial.

The optimal configuration is chosen based on the mean value of Novelty and Diversity of the event-related and background messages (Fig A.1). The consensus set of parameters is expected to have the least differences between the synthetic and real-world data. The metric I take is the mean value of both properties. Moreover, I use the Elbow heuristic to avoid the divergence of the vocabularies between the SO and synthetic data. I note that a .1 decrease of the softmax temperature leads to a huge divergence between the properties of synthetic and real-world data, hence I do not decrease it. As the result of the optimisation, I set the softmax temperature to 1.0 and the top_k to 400.

Table A.3 Model performance, LDA feature space. In the table I report the number of features after the feature selection step together with ROC-AUC, PR-AUC, F1-score and permutation test p-value measures for LDA feature space.

			CatBoost		
	Number of features	ROC-AUC	PR-AUC	F1-score	P.test p-val
Multiple major event-based	1	.67	.56	.58	.07
Multiple minor event-based	2	.44	.51	.45	<.001
Multiple patch event-based	1	.59	.51	.46	<.001
Django minor event-based	5	.42	.51	.46	.50
Django patch event-based	4	.49	.54	.55	.15
Selenium minor event-based	4	.30	.51	.45	1.00
Selenium patch event-based	14	.59	.50	.46	.81
Multiple major c.w.-based	9	.47	.53	.54	.18
Multiple minor c.w.-based	2	.57	.51	.37	<.001
Multiple patch c.w.-based	2	.51	.50	.52	.40
Django minor c.w.-based	17	.52	.50	.44	1.00
Django patch c.w.-based	2	.44	.50	.53	.21
Selenium minor c.w.-based	7	.43	.50	.45	1.00
Selenium patch c.w.-based	2	.54	.51	.46	1.00
			Random Forest		
	Number of features	ROC-AUC	PR-AUC	F1-score	P.test p-val
Multiple major event-based	1	.66	.57	.58	.06
Multiple minor event-based	1	.52	.59	.52	.19
Multiple patch event-based	1	.50	.70	.51	.01
Django minor event-based	14	.48	.53	.44	.02
Django patch event-based	1	.56	.50	.51	.30
Selenium minor event-based	6	.35	.51	.47	1.00
Selenium patch event-based	14	.49	.50	.46	.03
Multiple major c.w.-based	9	.46	.50	.50	.23
Multiple minor c.w.-based	7	.60	.51	.50	.22
Multiple patch c.w.-based	2	.45	.51	.46	.49
Django minor c.w.-based	10	.57	.50	.44	1.00
Django patch c.w.-based	11	.37	.51	.44	1.00
Selenium minor c.w.-based	18	.31	.51	.48	1.00
Selenium patch c.w.-based	5	.45	.51	.46	1.00
			Logistic Regression		
	Number of features	ROC-AUC	PR-AUC	F1-score	P.test p-val
Multiple major event-based	18	.32	.40	.49	.88
Multiple minor event-based	10	.38	.45	.45	.93
Multiple patch event-based	1	.52	.50	.46	.98
Django minor event-based	4	.53	.51	.43	.36
Django patch event-based	2	.37	.47	.47	.44
Selenium minor event-based	12	.37	.46	.47	.98
Selenium patch event-based	1	.54	.52	.45	.46
Multiple major c.w.-based	18	.42	.48	.48	.80
Multiple minor c.w.-based	15	.56	.54	.52	.19
Multiple patch c.w.-based	17	.46	.46	.32	.89
Django minor c.w.-based	4	.48	.50	.44	.52
Django patch c.w.-based	5	.42	.48	.47	.86
Selenium minor c.w.-based	8	.41	.48	.48	.72
Selenium patch c.w.-based	14	.60	.52	.47	.21

Table A.4 Model performance, hSBM feature space. In the table I report the number of features after the feature selection step together with ROC-AUC, PR-AUC, F1-score and permutation test p-value measures for hSBM feature space.

		CatBoost			
	Number of features	ROC-AUC	PR-AUC	F1-score	P.test p-val
Multiple major event-based	674	.26	.54	.16	1.00
Multiple minor event-based	540	.64	.50	.51	.01
Multiple patch event-based	1	.56	.50	.35	.81
Django minor event-based	1	.50	.50	.50	.26
Django patch event-based	473	.55	.52	.46	1.00
Selenium minor event-based	473	.59	.52	.46	1.00
Selenium patch event-based	674	.52	.50	.44	1.00
Multiple major c.w.-based	71	.55	.50	.46	1.00
Multiple minor c.w.-based	4	.47	.50	.48	.03
Multiple patch c.w.-based	607	.52	.51	.51	.14
Django minor c.w.-based	138	.47	.51	.46	.04
Django patch c.w.-based	272	.59	.50	.48	1.00
Selenium minor c.w.-based	205	.62	.51	.47	1.00
Selenium patch c.w.-based	4	.59	.52	.45	.44
		Random Forest			
	Number of features	ROC-AUC	PR-AUC	F1-score	P.test p-val
Multiple major event-based	138	.71	.52	.16	1.00
Multiple minor event-based	1	.50	.75	.49	<.001
Multiple patch event-based	1	.59	.50	.41	.88
Django minor event-based	1	.57	.55	.47	<.001
Django patch event-based	1	.64	.54	.47	1.00
Selenium minor event-based	607	.48	.50	.46	1.00
Selenium patch event-based	1	.43	.51	.44	1.00
Multiple major c.w.-based	138	.48	.50	.46	1.00
Multiple minor c.w.-based	406	.56	.51	.45	.29
Multiple patch c.w.-based	473	.45	.51	.40	<.001
Django minor c.w.-based	1	.52	.51	.45	1.00
Django patch c.w.-based	607	.48	.50	.48	1.00
Selenium minor c.w.-based	607	.60	.56	.35	.07
Selenium patch c.w.-based	607	.69	.52	.47	1.00
		Logistic Regression			
	Number of features	ROC-AUC	PR-AUC	F1-score	P.test p-val
Multiple major event-based	674	.29	.52	.20	.76
Multiple minor event-based	406	.60	.50	.36	.90
Multiple patch event-based	1	.37	.51	.49	<.001
Django minor event-based	138	.37	.51	.41	.01
Django patch event-based	607	.54	.51	.42	.15
Selenium minor event-based	1	.52	.50	.46	1.00
Selenium patch event-based	1	.52	.50	.44	1.00
Multiple major c.w.-based	4	.57	.51	.47	1.00
Multiple minor c.w.-based	71	.57	.51	.53	.01
Multiple patch c.w.-based	4	.44	.51	.46	.02
Django minor c.w.-based	473	.39	.50	.45	.54
Django patch c.w.-based	4	.55	.50	.48	1.00
Selenium minor c.w.-based	71	.57	.51	.47	.10
Selenium patch c.w.-based	406	.55	.50	.54	.08

Appendix B

Supplementary materials for Chapter 4

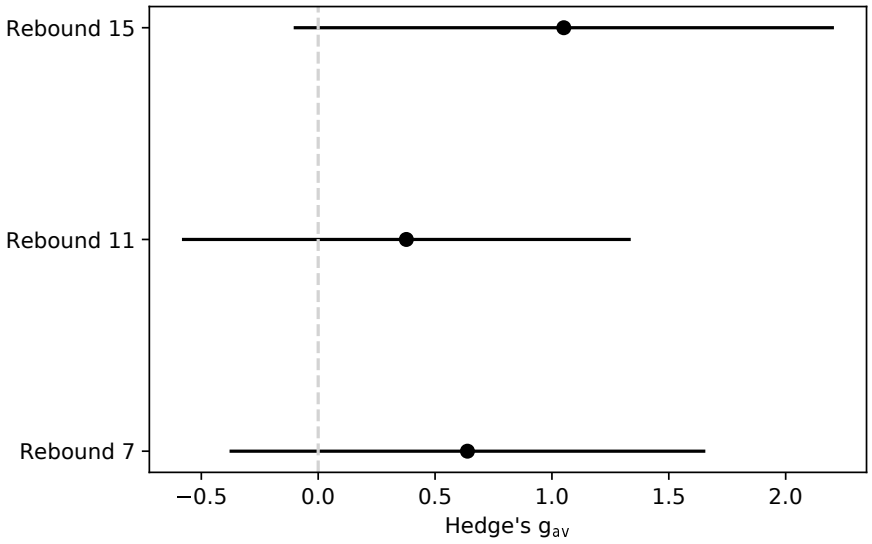


Fig. B.1 Hedge's g_{av} effect sizes quantifying the improvement of the precision from using the CatBoost over the no-information estimator. The error bars illustrate the .95 confidence intervals, corrected for multiple comparisons (3 in this case). The dashed line corresponds to the significance threshold. Rebounds accord to different labelling configurations, where 7, 11 and 15 are ticks required for the positive labelling of an entry.

Table B.1 Model performance measures reported for the 2-step feature extraction, 3 different labelling configurations: 7, 11 and 15 ticks rebounds. Null-Precision corresponds to the performance of an always-positive classifier.

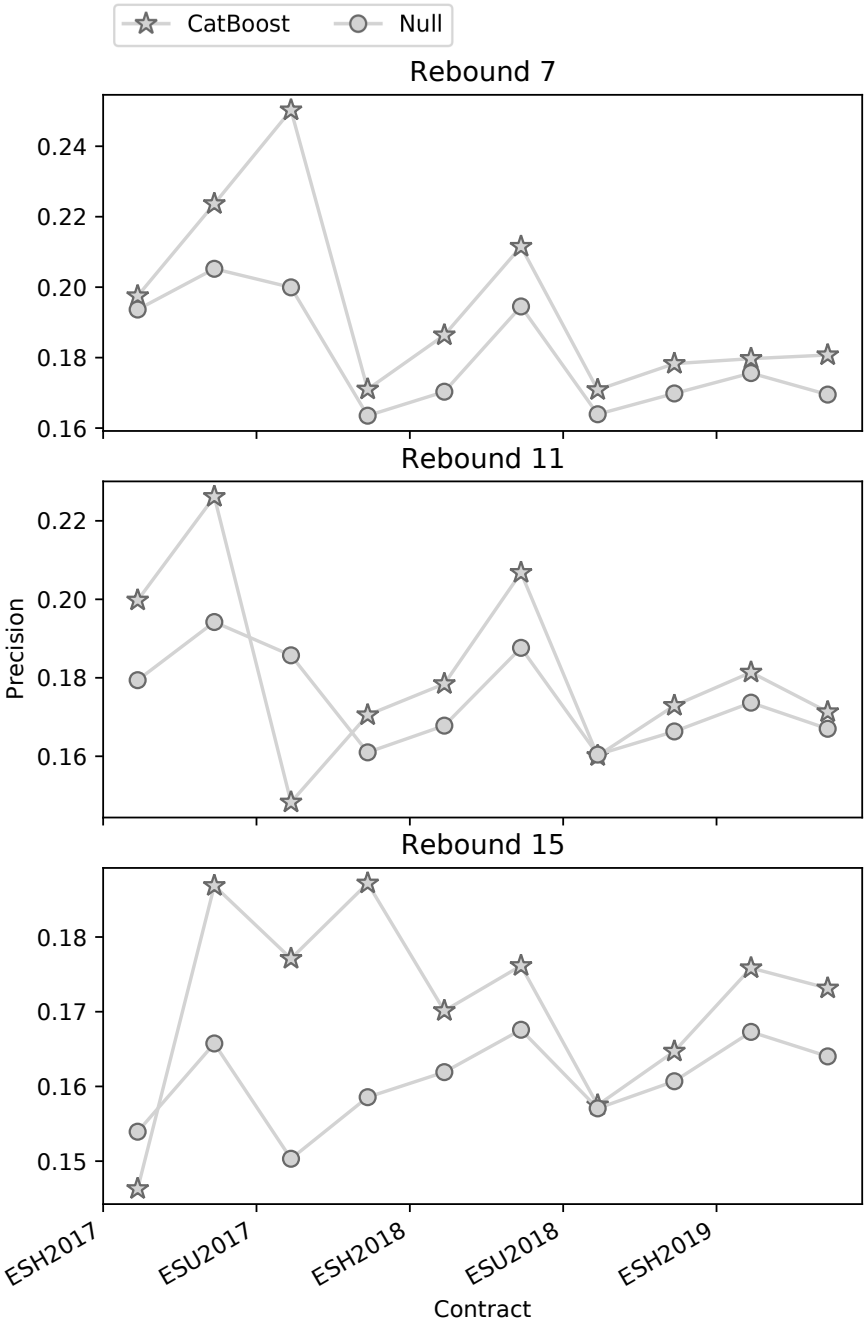
Contract	PR-AUC	F1-score	Precision	ROC-AUC	Null-Precision
Rebound 7					
ESH2017	.19	.26	.20	.51	.19
ESM2017	.25	.28	.22	.54	.21
ESU2017	.25	.29	.25	.56	.20
ESZ2017	.18	.16	.17	.52	.16
ESH2018	.19	.28	.19	.54	.17
ESM2018	.22	.32	.21	.55	.19
ESU2018	.17	.18	.17	.52	.16
ESZ2018	.18	.28	.18	.53	.17
ESH2019	.18	.24	.18	.51	.18
ESM2019	.17	.24	.18	.52	.17
Rebound 11					
ESH2017	.19	.30	.20	.53	.18
ESM2017	.21	.14	.23	.54	.19
ESU2017	.18	.14	.15	.50	.19
ESZ2017	.17	.23	.17	.52	.16
ESH2018	.19	.27	.18	.53	.17
ESM2018	.21	.31	.21	.55	.19
ESU2018	.16	.17	.16	.50	.16
ESZ2018	.19	.27	.17	.52	.17
ESH2019	.18	.24	.18	.52	.17
ESM2019	.17	.25	.17	.52	.17
Rebound 15					
ESH2017	.15	.20	.15	.51	.15
ESM2017	.18	.21	.19	.53	.17
ESU2017	.17	.17	.18	.54	.15
ESZ2017	.17	.11	.19	.51	.16
ESH2018	.17	.26	.17	.52	.16
ESM2018	.18	.28	.18	.53	.17
ESU2018	.16	.22	.16	.51	.16
ESZ2018	.17	.26	.16	.52	.16
ESH2019	.17	.26	.18	.51	.17
ESM2019	.17	.26	.17	.52	.16

Table B.2 Model performance measures reported for the Price Level feature extraction component, 3 different labelling configurations: 7, 11 and 15 ticks rebounds. Null-Precision corresponds to the performance of an always-positive classifier.

Contract	PR-AUC	F1-score	Precision	ROC-AUC	Null-Precision
Rebound 7					
ESH2017	.19	.24	.19	.50	.19
ESM2017	.21	.28	.21	.50	.21
ESU2017	.22	.25	.23	.53	.20
ESZ2017	.16	.19	.16	.49	.16
ESH2018	.18	.23	.17	.50	.17
ESM2018	.19	.25	.19	.49	.19
ESU2018	.16	.15	.15	.50	.16
ESZ2018	.18	.24	.17	.51	.17
ESH2019	.18	.26	.18	.51	.18
ESM2019	.18	.21	.18	.50	.17
Rebound 11					
ESH2017	.17	.19	.17	.49	.18
ESM2017	.20	.18	.19	.51	.19
ESU2017	.18	.18	.17	.50	.19
ESZ2017	.16	.19	.16	.50	.16
ESH2018	.17	.23	.18	.51	.17
ESM2018	.19	.17	.20	.50	.19
ESU2018	.16	.22	.16	.50	.16
ESZ2018	.18	.20	.18	.52	.17
ESH2019	.18	.25	.17	.51	.17
ESM2019	.17	.22	.17	.50	.17
Rebound 15					
ESH2017	.15	.20	.15	.50	.15
ESM2017	.17	.24	.18	.51	.17
ESU2017	.17	.16	.19	.53	.15
ESZ2017	.16	.13	.15	.50	.16
ESH2018	.16	.25	.16	.50	.16
ESM2018	.18	.21	.17	.51	.17
ESU2018	.15	.16	.15	.49	.16
ESZ2018	.16	.19	.16	.51	.16
ESH2019	.17	.24	.17	.50	.17
ESM2019	.16	.18	.16	.49	.16

Table B.3 Model performance measures reported for the Market Shift feature extraction component, 3 different labelling configurations: 7, 11 and 15 ticks rebounds. Null-Precision corresponds to the performance of an always-positive classifier.

Contract	PR-AUC	F1-score	Precision	ROC-AUC	Null-Precision
Rebound 7					
ESH2017	.20	.26	.20	.51	.19
ESM2017	.25	.28	.24	.55	.21
ESU2017	.25	.31	.25	.56	.20
ESZ2017	.17	.26	.18	.52	.16
ESH2018	.20	.29	.19	.54	.17
ESM2018	.21	.33	.22	.55	.19
ESU2018	.17	.23	.17	.51	.16
ESZ2018	.18	.28	.18	.52	.17
ESH2019	.19	.28	.19	.53	.18
ESM2019	.18	.27	.18	.53	.17
Rebound 11					
ESH2017	.18	.24	.19	.51	.18
ESM2017	.21	.24	.20	.50	.19
ESU2017	.21	.31	.23	.56	.19
ESZ2017	.16	.12	.16	.50	.16
ESH2018	.20	.28	.18	.54	.17
ESM2018	.20	.32	.21	.55	.19
ESU2018	.17	.22	.17	.51	.16
ESZ2018	.17	.27	.17	.52	.17
ESH2019	.18	.26	.18	.52	.17
ESM2019	.18	.27	.18	.53	.17
Rebound 15					
ESH2017	.15	.22	.15	.48	.15
ESM2017	.17	.20	.16	.50	.17
ESU2017	.16	.17	.15	.51	.15
ESZ2017	.16	.11	.16	.50	.16
ESH2018	.17	.27	.17	.53	.16
ESM2018	.18	.28	.18	.54	.17
ESU2018	.16	.22	.16	.51	.16
ESZ2018	.17	.27	.17	.52	.16
ESH2019	.17	.27	.17	.52	.17
ESM2019	.18	.27	.18	.53	.16



The p

Fig. B.2 Precision of the CatBoost model and the always-positive estimator. The plot shows the labelling configurations of 7, 11 and 15 ticks rebounds.

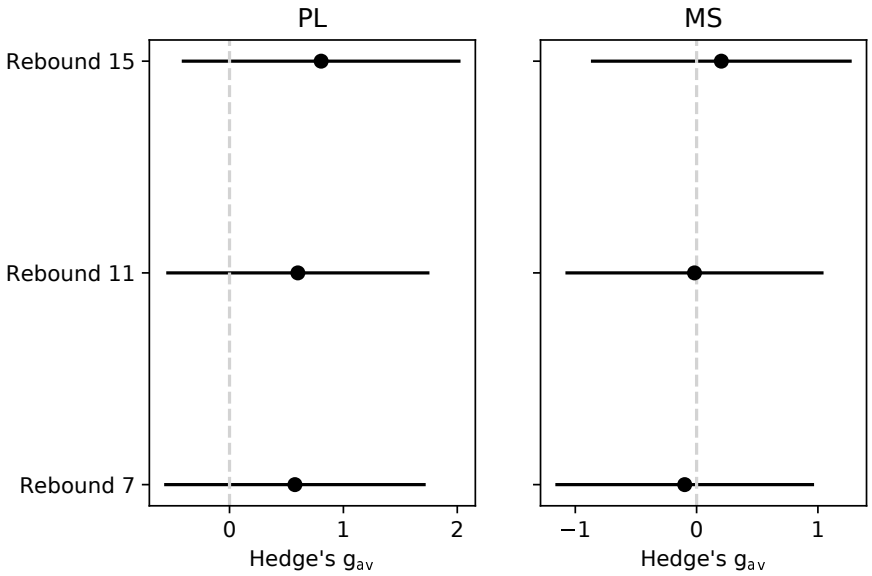


Fig. B.3 Hedge's g_{av} effect sizes quantify the improvement of the precision from using the 2-step feature extraction over each of the components (PL and MS). The error bars illustrate the .95 confidence intervals, corrected for multiple comparisons (6 in this case). The dashed line corresponds to the significance threshold. Rebounds accord to different labelling configurations, where 7, 11 and 15 are ticks required for the positive labelling of an entry.

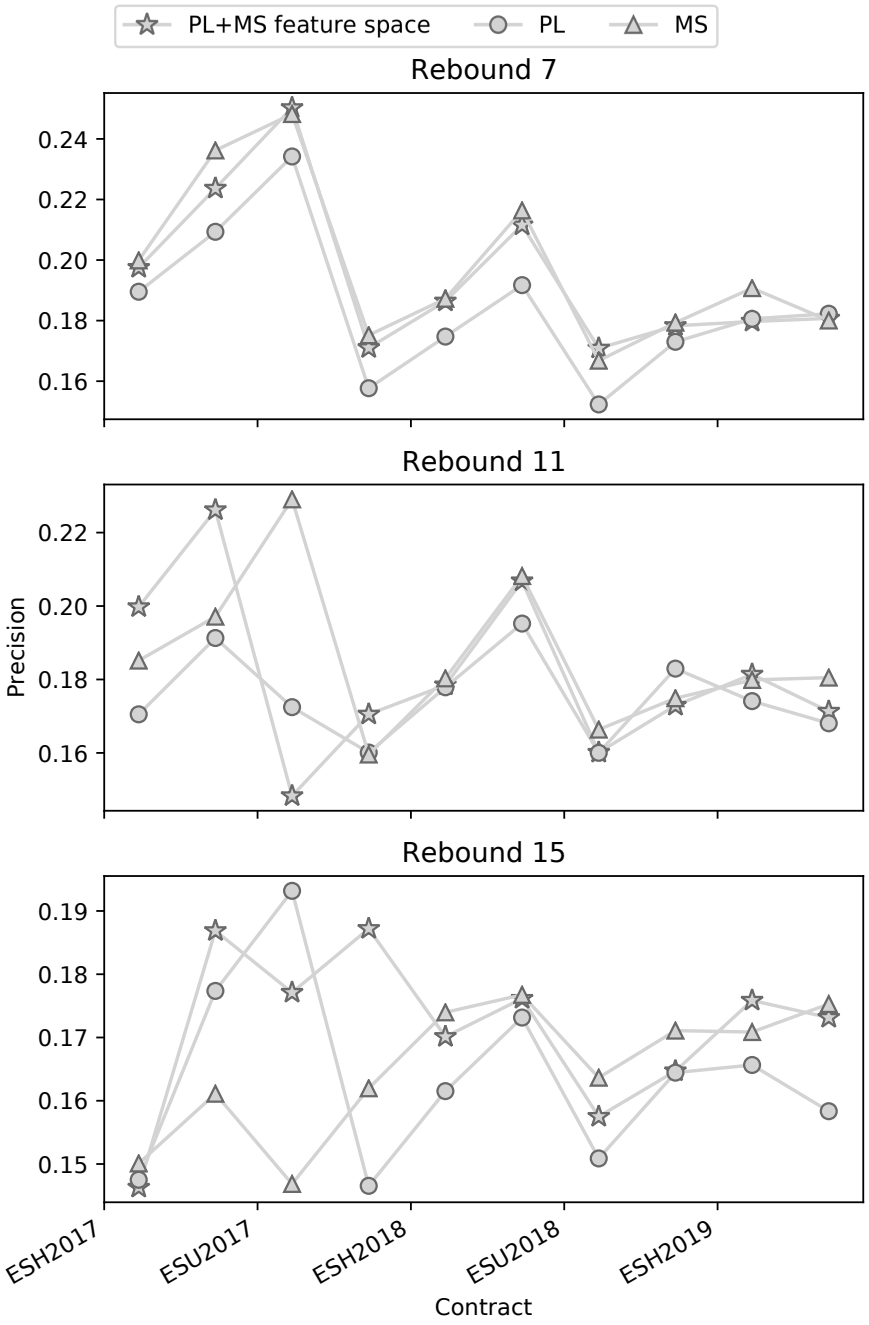


Fig. B.4 The precision of the model which uses the 2-step feature extraction (MS+PL) versus the performance of the models using the single-step feature extraction (MS, PL). The plot shows the labelling configurations of 7, 11 and 15 ticks rebounds.

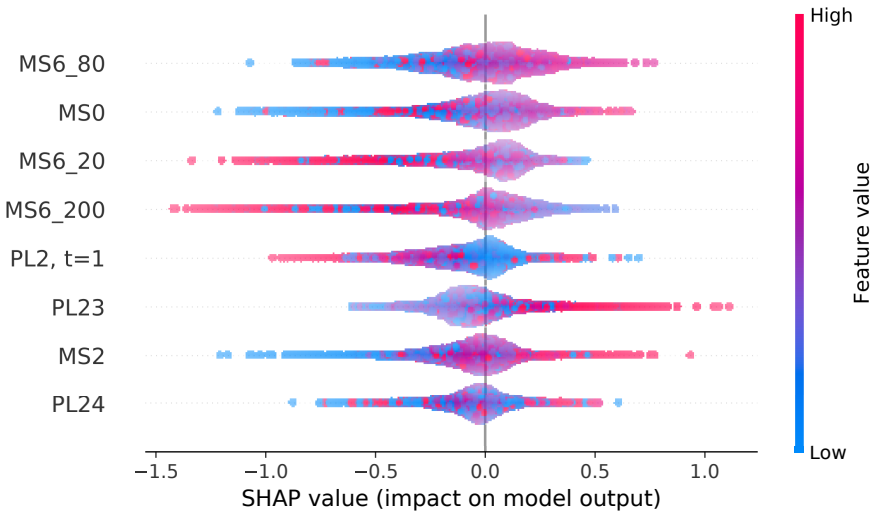


Fig. B.5 SHAP summary plot of the model trained on ESH2019 contract, rebound 7 configuration. Each marker is a classified entry. X-axis quantifies the contribution of the entries towards the positive or negative class output.

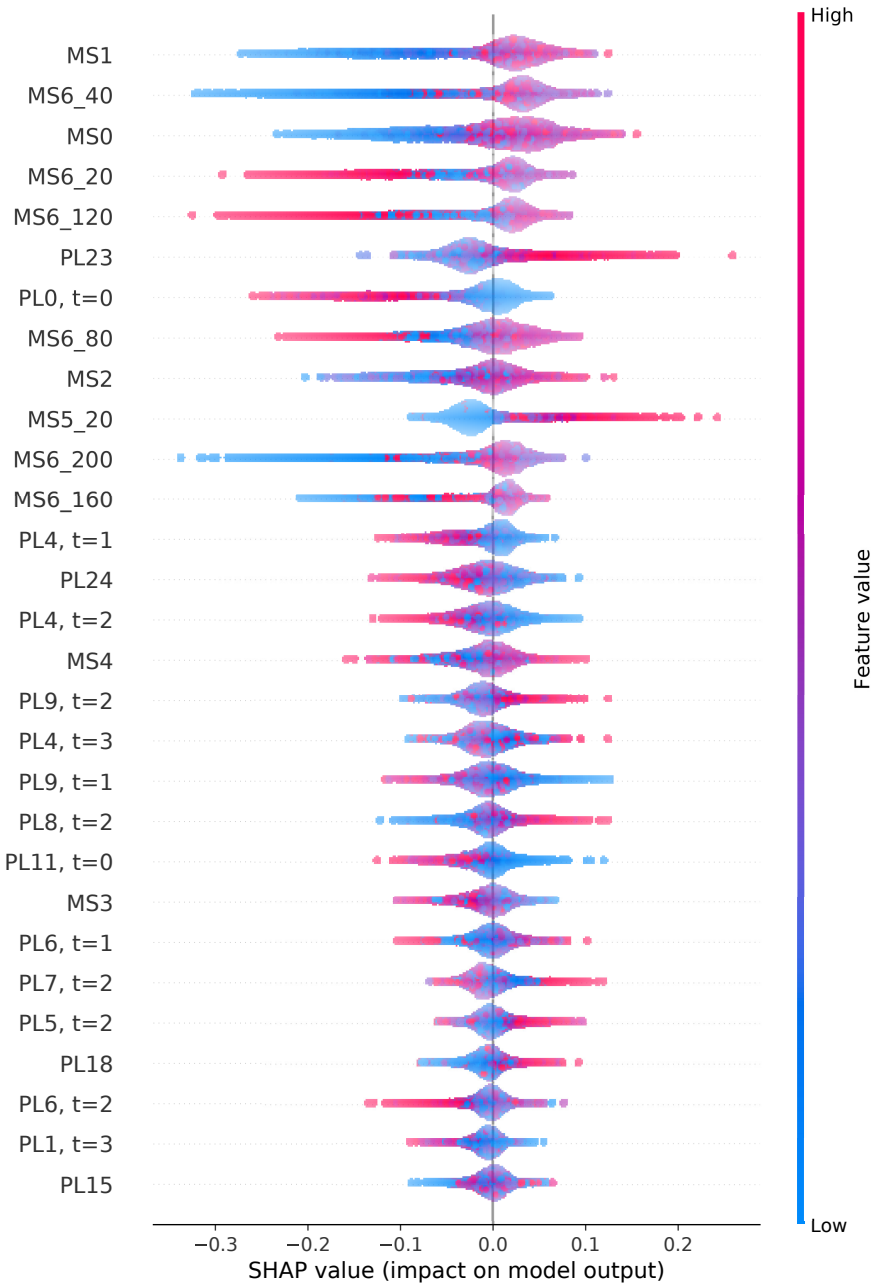


Fig. B.6 SHAP summary plot of the model trained on ESH2019 contract, rebound 11 configuration. Each marker is a classified entry. X-axis quantifies the contribution of the entries towards the positive or negative class output.

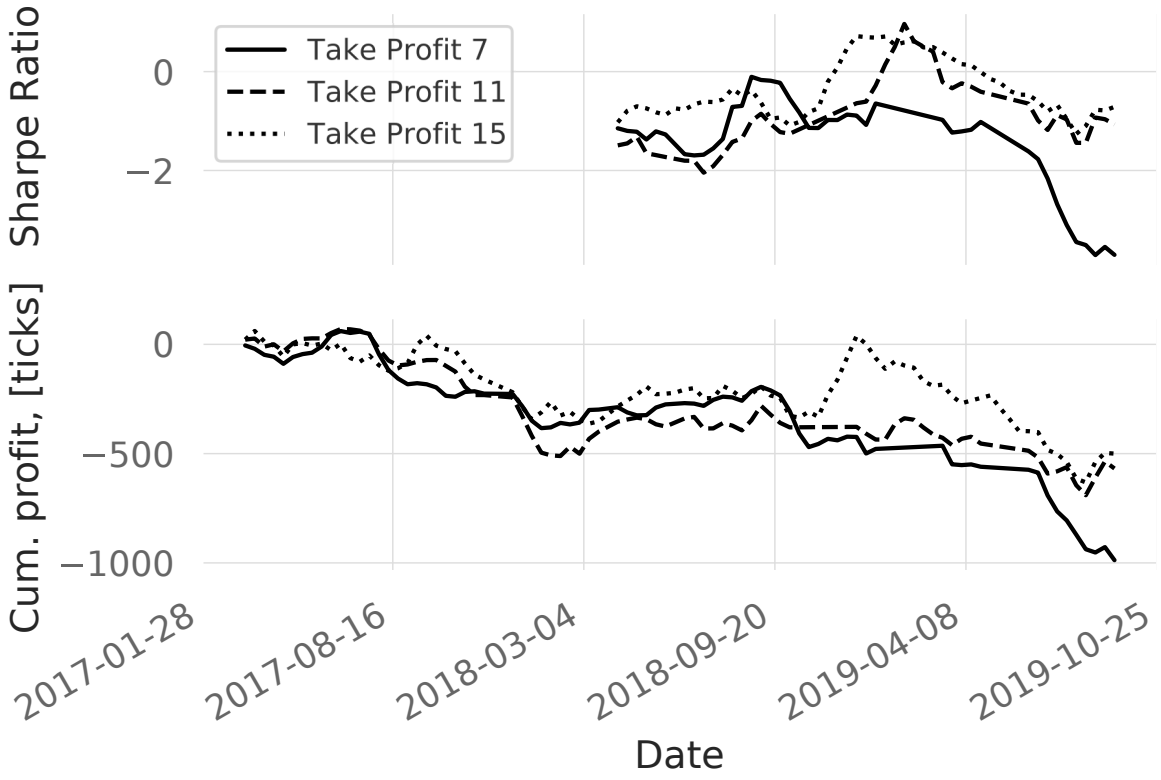


Fig. B.7 Cumulative profit curves for all the rebound configurations and fixed take profit of 15 ticks for years 2017-2019 with the corresponding annualized rolling Sharpe ratios (computed for 5% risk-free income). The trading fees are already included in the cumulative profits.

Appendix C

Supplementary materials for Chapter 5

Table C.1 Performance metrics for ES, price levels pattern extraction method. The dates are reported in the form MM/YY. "Null_precision" corresponds to the no-information model precision.

	PR-AUC	ROC-AUC	F1-score	Precision	Null_precision
3/17 to 6/17	0.14	0.44	0.11	0.10	0.15
6/17 to 9/17	0.19	0.53	0.14	0.14	0.18
9/17 to 12/17	0.15	0.49	0.20	0.16	0.16
12/17 to 3/18	0.16	0.50	0.17	0.15	0.16
3/18 to 6/18	0.16	0.48	0.07	0.12	0.17
6/18 to 9/18	0.16	0.51	0.25	0.17	0.16
9/18 to 12/18	0.16	0.51	0.10	0.16	0.15
12/18 to 3/19	0.18	0.53	0.13	0.18	0.17
3/19 to 6/19	0.16	0.50	0.18	0.17	0.17
6/19 to 9/19	0.15	0.50	0.14	0.14	0.15
9/19 to 12/19	0.14	0.50	0.14	0.13	0.15
12/19 to 3/20	0.17	0.50	0.15	0.18	0.17
3/20 to 6/20	0.17	0.51	0.21	0.17	0.16

Table C.2 Performance metrics for B6, price levels pattern extraction method. "Null_precision" corresponds to the no-information model precision.

	PR-AUC	ROC-AUC	F1-score	Precision	Null_precision
3/17 to 6/17	0.17	0.47	0.15	0.15	0.17
6/17 to 9/17	0.16	0.50	0.15	0.18	0.16
9/17 to 12/17	0.16	0.49	0.24	0.16	0.16
12/17 to 3/18	0.15	0.48	0.15	0.17	0.16
3/18 to 6/18	0.12	0.51	0.13	0.11	0.13
6/18 to 9/18	0.18	0.54	0.27	0.20	0.16
9/18 to 12/18	0.15	0.50	0.18	0.13	0.13
12/18 to 3/19	0.19	0.51	0.22	0.14	0.15
3/19 to 6/19	0.18	0.53	0.11	0.18	0.16
6/19 to 9/19	0.20	0.51	0.21	0.22	0.19
9/19 to 12/19	0.14	0.45	0.13	0.15	0.16
12/19 to 3/20	0.17	0.50	0.20	0.22	0.15
3/20 to 6/20	0.20	0.52	0.12	0.14	0.16

Table C.3 Statistics supporting the outcomes of the Wilcoxon test. The test is aimed to validate whether on the VCRB data and the considered feature space, CatBoost performs significantly better than the no-information estimator. The result is reported for the range configurations of 5, 9 and 11.

Statistics	Dataset			
	Range 5			
	ES		B6	
One-tailed Wilcoxon test p-value	< .001		.024	
Test Statistics	90.0		74.0	
	CatBoost	No-information	CatBoost	No-information
Mean (precision)	0.25	0.24	0.23	0.23
Median (precision)	0.25	0.24	0.23	0.23
Standard Deviation (precision)	0.0084	0.0049	0.0081	0.0039
	Range 9			
	ES		B6	
One-tailed Wilcoxon test p-value	.0199		.47	
Test Statistics	75.0		47.0	
	CatBoost	No-information	CatBoost	No-information
Mean (precision)	0.24	0.23	0.22	0.22
Median (precision)	0.24	0.23	0.23	0.22
Standard Deviation (precision)	0.0109	0.0043	0.0112	0.0068
	Range 11			
	ES		B6	
One-tailed Wilcoxon test p-value	.0164		.170	
Test Statistics	76.0		60.0	
	CatBoost	No-information	CatBoost	No-information
Mean (precision)	0.24	0.24	0.23	0.23
Median (precision)	0.24	0.23	0.23	0.23
Standard Deviation (precision)	0.0118	0.0057	0.0201	0.0082

Table C.4 Statistics supporting the outcomes of the Wilcoxon test which validates whether Volume-based pattern extraction method leads to better classification performance than the price level pattern extraction. The result is reported for the range configurations of 5, 9 and 11.

Statistics	Dataset			
	Range 5			
	ES		B6	
One-tailed Wilcoxon test p-value	< .001		< .001	
Test Statistics	91.0		91.0	
	VCRB	Price levels	VCRB	Price levels
Mean (PR-AUC)	0.25	0.16	0.23	0.17
Median (PR-AUC)	0.25	0.16	0.23	0.17
Standard Deviation (PR-AUC)	0.0066	0.0118	0.0060	0.022
	Range 9			
	ES		B6	
One-tailed Wilcoxon test p-value	< .001		< .001	
Test Statistics	91.0		91.0	
	VCRB	Price levels	VCRB	Price levels
Mean (PR-AUC)	0.24	0.16	0.22	0.17
Median (PR-AUC)	0.24	0.16	0.22	0.17
Standard Deviation (PR-AUC)	0.0106	0.0118	0.0085	0.022
	Range 11			
	ES		B6	
One-tailed Wilcoxon test p-value	< .001		< .001	
Test Statistics	91.0		91.0	
	VCRB	Price levels	VCRB	Price levels
Mean (PR-AUC)	0.24	0.16	0.23	0.17
Median (PR-AUC)	0.24	0.16	0.23	0.17
Standard Deviation (PR-AUC)	0.0126	0.0118	0.0131	0.022

Table C.5 Statistics supporting the outcomes of the Wilcoxon test which assesses whether VCRB pattern extraction method leads to better classification performance on the more liquid market (ES in comparison to B6). The result is reported for the range configurations of 5, 9 and 11.

Statistics	Datasets	
	Range 5	
One-tailed Wilcoxon test p-value	< .001	
Test Statistics	91.0	
	ES	B6
Mean (PR-AUC)	0.25	0.23
Median (PR-AUC)	0.25	0.23
Standard Deviation (PR-AUC)	0.0066	0.0060
	Range 9	
One-tailed Wilcoxon test p-value	< .001	
Test Statistics	88.0	
	ES	B6
Mean (PR-AUC)	0.24	0.22
Median (PR-AUC)	0.24	0.22
Standard Deviation (PR-AUC)	0.0106	0.0085
	Range 11	
One-tailed Wilcoxon test p-value	.040	
Test Statistics	71.0	
	ES	B6
Mean (PR-AUC)	0.24	0.23
Median (PR-AUC)	0.24	0.23
Standard Deviation (PR-AUC)	0.0126	0.0131

Table C.6 Statistics supporting the outcomes of the Wilcoxon test which assesses whether SHAP and decision paths feature interaction extraction methods are related significantly stronger than the bootstrapped data. Footstep distance is inversely proportional to the relatedness. The result is reported for the range configurations of 5, 9 and 11.

Statistics	Dataset			
	Range 5			
	ES		B6	
One-tailed Wilcoxon test p-value	< .001		< .001	
Test Statistics	3.0		2.0	
	Actual distance	Bootstrapped	Actual distance	Bootstrapp
Mean (footstep distance)	88247	93280	87891	93279
Median (footstep distance)	87390	93281	88408	93278
Standard Deviation (footstep distance)	3166.0	5.0	4083.0	3.4
	Range 9			
	ES		B6	
One-tailed Wilcoxon test p-value	.00171		< .001	
Test Statistics	6.0		0.0	
	Actual distance	Bootstrapped	Actual distance	Bootstrapp
Mean (footstep distance)	88188	93279	86172	93278
Median (footstep distance)	88728	93279	86132	93279
Standard Deviation (footstep distance)	4429.0	6.1	2998.0	4.6
	Range 11			
	ES		B6	
One-tailed Wilcoxon test p-value	< .001		< .001	
Test Statistics	1.0		0.0	
	Actual distance	Bootstrapped	Actual distance	Bootstrapp
Mean (footstep distance)	87266	93281	87228	93280
Median (footstep distance)	86994	93281	86198	93279
Standard Deviation (footstep distance)	3081.0	5.5	2925.0	3.4

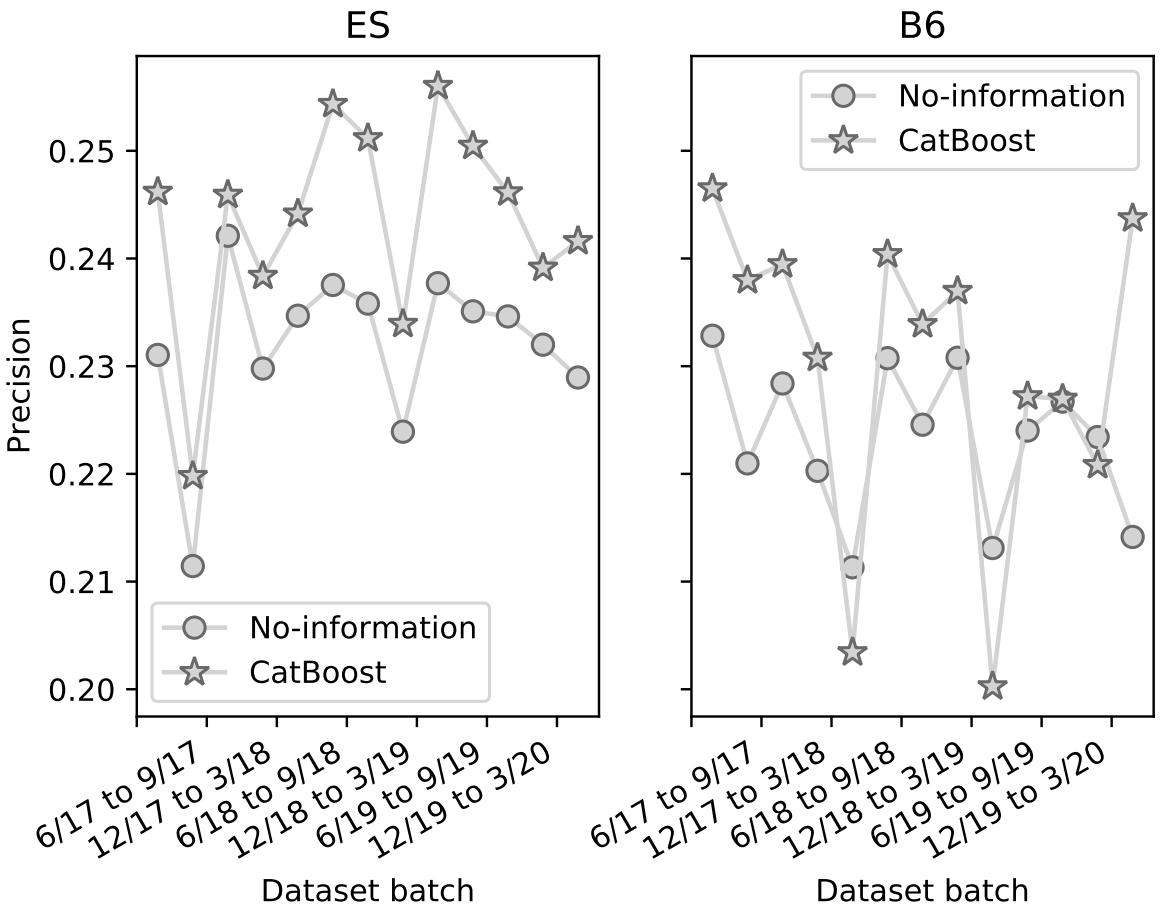


Fig. C.1 Precision performance metric for the no-information model and CatBoost plotted for both instruments - ES and B6. The Y-axis is mutual for both subplots.

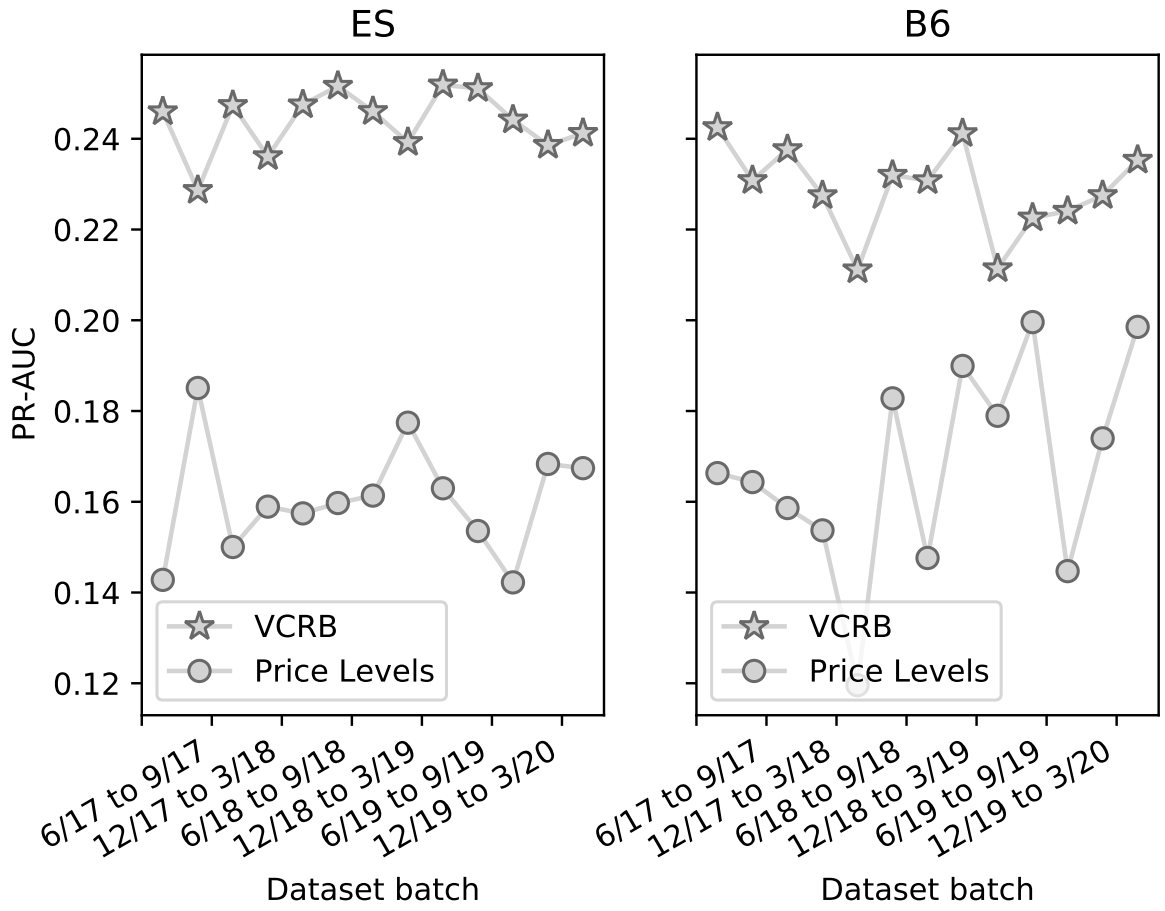


Fig. C.2 We plot PR-AUC for volume-based pattern extraction method and price level-based. The metric is reported for ES and B6 instruments. Volume-based method is reported for range 7 configuration.

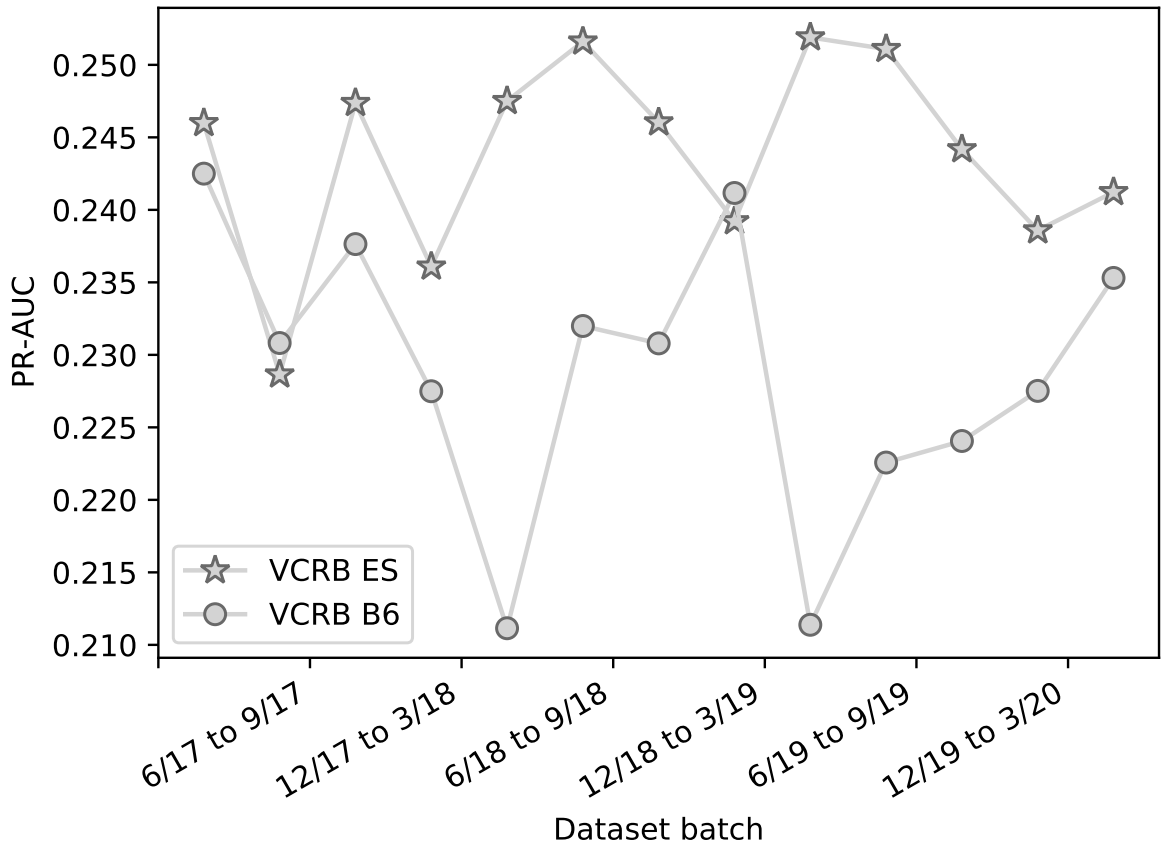


Fig. C.3 PR-AUC of CatBoost models obtained for ES and B6 futures instruments. Reported for range 7 configuration.

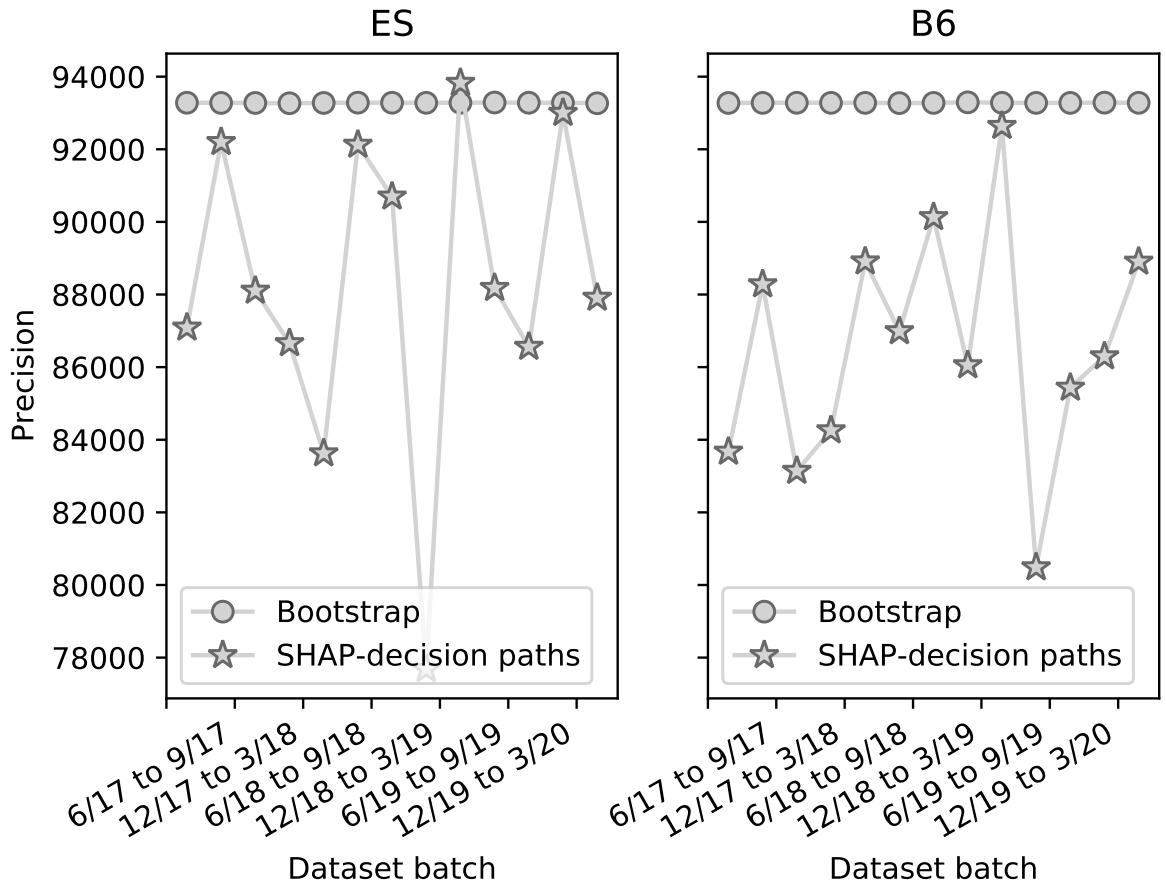


Fig. C.4 Footrule distances between ranks of the feature interactions for SHAP and decision path-based methods for S&P E-mini and British Pound futures instruments.