RESEARCH ARTICLE

WILEY

Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods

Adamantios Ntakaris[1] | Martin Magris[2] | Juho Kanniainen[2] | Moncef Gabbouj[1] | Alexandros Iosifidis[3]

[1]Laboratory of Signal Processing, Tampere University of Technology, Tampere, Finland

[2]Laboratory of Industrial and Information Management, Tampere University of Technology, Tampere, Finland

[3]Department of Engineering, Electrical and Computer Engineering, Aarhus University, Aarhus, Denmark

Correspondence
Adamantios Ntakaris, Laboratory of Signal Processing, Tampere University of Technology, Korkeakoulunkatu 1, Tampere, Finland.
Email: adamantios.ntakaris@tut.fi

Abstract

Managing the prediction of metrics in high-frequency financial markets is a challenging task. An efficient way is by monitoring the dynamics of a limit order book to identify the information edge. This paper describes the first publicly available benchmark dataset of high-frequency limit order markets for mid-price prediction. We extracted normalized data representations of time series data for five stocks from the Nasdaq Nordic stock market for a time period of 10 consecutive days, leading to a dataset of ~4,000,000 time series samples in total. A day-based anchored cross-validation experimental protocol is also provided that can be used as a benchmark for comparing the performance of state-of-the-art methodologies. Performance of baseline approaches are also provided to facilitate experimental comparisons. We expect that such a large-scale dataset can serve as a testbed for devising novel solutions of expert systems for high-frequency limit order book data analysis.

KEYWORDS
high-frequency trading, limit order book, mid-price, machine learning, ridge regression, single hidden feedforward neural network

1 | INTRODUCTION

Automated trading became a reality when the majority of exchanges adopted it globally. This environment is ideal for high-frequency traders. High-frequency trading (HFT) and a centralized matching engine, referred to as a limit order book (LOB), are the main drivers for generating big data (Seddon & Currie, 2017). In this paper, we describe a new order book dataset consisting of approximately 4 million events for 10 consecutive trading days for five stocks. The data are derived from the ITCH feed provided by Nasdaq OMX Nordic and consists of the time-ordered sequences of messages that track and record all the events occurring in the specific market. It provides a complete market-wide history of 10 trading days. Additionally, we define an experimental protocol to evaluate the performance of research methods in mid-price prediction.[1]

Datasets, like the one presented here, come with challenges, including the selection of appropriate data transformation, normalization, description, and classification. This type of massive dataset requires a very good understanding of the available information that can be extracted

---

[1]Mid-price is the average of the best bid and best ask prices.

for further processing. We follow the information edge, as has been recently presented by Kercheval and Zhang (2015). The authors provide a detailed description of representations that can be used for a mid-price movement prediction metric. In light of this data representation, they apply nonlinear classification based on support vector machines (SVM) in order to predict the movement of this metric. Such a supervised learning model exploits class labels[2] for short- and long-term prediction. However, they train their model based on a very small (when compared to the size of the data that can be available for such applications) dataset of 4,000 samples. This is due to the limitations of many nonlinear kernel-based classification models related to their time and space complexity with respect to the training data size. On the other hand, Sirignano (2016) uses large amounts of data for nonlinear classification based on a feedforward network. The author takes advantage of the local spatial structure[3] of the data for modeling the joint distribution of the LOB's state based on its current state.

Despite the major importance of publicly available datasets for advancing research in the HFT field, there are no detailed public available benchmark datasets for method evaluation purposes. In this paper, we describe the first publicly available dataset[4] for an LOB-based HFT that has been collected in the hope of facilitating future research in the field. Based on Kercheval and Zhang (2015), we provide time series representations of approximately 4,000,000 trading events and annotations for five classification problems. Baseline results of two widely used methods—that is, linear and nonlinear regression models, are also provided. In this way, we introduce this new problem for the expert systems community and provide a testbed for facilitating future research. We hope that attracting the interest of expert systems will lead to the rapid improvement of the performance achieved in the provided dataset, thus leading to much better state-of-the-art solutions to this important problem.

The dataset described in this paper can be useful for financial expert systems in two ways. First, it can be used to identify circumstances under which markets are stable, which is very important for liquidity providers (market makers) to make the spread. Consequently, such an intelligent system would be valuable as a framework that can increase liquidity provision. Secondly, analysis of the data

can be used for model selection by speculative traders, who are trading based on their predictions on market movements. In future research, this paper can be employed to identify order book spoofing—that is, situations where markets are exposed to manipulation by limit orders. In this case, spoofers could aim to move markets in certain directions by limit orders that are canceled before they are filled. Therefore, this research is relevant not only for market makers and traders but also for supervisors and regulators.

Therefore, the present work makes the following contributions: (1) To the best of our knowledge this is the first publicly available LOB-ITCH dataset for machine learning experiments on the prediction of mid-price movements. (2) We provide baselines methods based on ridge regression and a new implementation of an RBF neural network based on k-means algorithm. (3) The paper provides information about the prediction of mid-price movements to market makers, traders, and regulators. This paper does not suggest any trading strategies and is reliant on purely machine learning metrics prediction. Overall, this work is an empirical exploration of the challenges that come with high-frequency trading and machine learning applications.

The data from Nasdanq Helsinki Stock Exchange offers important benefits. In the USA the limit orders for a given asset are spread between several exchanges, causing fragmentation of liquidity. The fragmentation poses a problem for empirical research, because, as Gould, Porter, Williams, McDonald, Fenn, and Howison (2013) point out, the "differences between different trading platforms' matching rules and transaction costs complicate comparisons between different limit order books for the same asset." These issues related to fragmentation are not present with data obtained from less fragmented Nasdaq Nordic markets. Moreover, Helsinki Exchange is a pure limit order market, where the market makers have a limited role.

The rest of the paper is organized as follows. We provide a comprehensive literature review of the field in Section 2. Dataset and experimental protocol descriptions are provided in Section 33. Quantitative and qualitative comparisons of the new dataset, along with related data sources, are provided in Section 4. In Section 5, we describe the engineering of our baselines. Section 6 presents our empirical results and Section 7 concludes.

2 | MACHINE LEARNING FOR HFT AND LOB

The complex nature of HFT and LOB spaces is suitable for interdisciplinary research. In this section, we provide a comprehensive review of recent methods exploiting

---

[2]Labels are extracted from annotations provided by experts and represent the direction of the mid-price. Three different states are defined—that is, upward, downward, and stationary movement.
[3]By local movement, the author means that the conditional movement of the future price (e.g., best ask price movement) depends, locally, on the current LOB state.
[4]The dataset can be downloaded from: https://etsin.avointiede.fi/dataset/urn-nbn-fi-csc-kata20170601153214969115https://etsin.avointiede.fi/dataset/urn-nbn-fi-csc-kata20170601153214969115.

machine learning approaches. Regression models, neural networks, and several other methods have been proposed to make inferences of the stock market. Existing literature ranges from metric prediction to optimal trading strategies identification. The research community has tried to tackle the challenges of prediction and data inference from different angles. Although mid-price prediction can be considered a traditional time series prediction problem, there are several challenges that justify HFT as a unique problem.

2.1 | Regression analysis

Regression models have been widely used for HFT and LOB prediction. Zheng, Moulines, and Abergel (2012) utilize logistic regression in order to predict the inter-trade price jump. Alvim, dos Santos, and Milidiu (2010) use support vector regression (SVR) and partial least squares (PLS) for trading volume forecasting for 10 Bovespa stocks. Pai and Lin (2005) use a hybrid model for stock price prediction. They combine an autoregressive integrated moving average (ARIMA) model and an SVM classifier in order to model nonlinearities of class structure in regression estimation models. Liu and Park (2015) develop a multivariate linear model to explain short-term stock price movement where a bid–ask spread is used for classification purposes. Detollenaere and D'hondt (2017) apply an adaptive least absolute shrinkage and selection operator (LASSO)[5] for variable selection, which best explains the transaction cost of the split order. They apply an adjusted ordinal logistic method for classifying ex ante transaction costs into groups. Cenesizoglu, Dionne, and Zhou (2014) work on a similar problem. They hold that the state of the limit order can be informative for the direction of future prices and try to prove their position by using an autoregressive model.

Panayi, Peters, Danielsson, and Zigrand (2016) use generalized linear models (GLM) and generalized additive models for location, shape, and scale (GAMLSS) models in order to relate the threshold exceedance duration (TED), which measures the length of time required for liquidity replenishment, to the state of the LOB. Yu (2006) tries to extract information from order information and order submission based on the ordered probit model.[6] The author shows, in the case of Shanghai's stock market, that an LOB's information is affected by the trader's strategy, with different impacts on the bid and ask sides. Amaya, Filbien, Okou, and Roch (2015) use panel regression[7] for order imbalances and liquidity costs in LOBs so as to identify resilience in the market. Their findings show that such order imbalances cause liquidity issues that last for up to 10 minutes. Malik and Lon Ng (2014) analyze the asymmetric intra-day patterns of LOBs. They apply regression with a power transformation on the notional volume weighted average price (NVWAP) curves in order to conclude that both sides of the market behave asymmetrically to market conditions.[8] In the same direction, Ranaldo (2004) examines the relationship between trading activity and the order flow dynamics in LOBs, where the empirical investigation is based on a probit model. Cao, Hansch, and Wang (2009) examine the depth of different levels of an order book by using an autoregressive (AR) model of order 5 (the AR(5) framework). They find that levels beyond the best bid and best ask prices provide moderate information regarding the true value of an asset. Finally, Creamer (2012) suggests that the LogitBoost algorithm is ideal for selecting the right combination of technical indicators.[9]

2.2 | Neural networks

HFT is mainly a scalping[10] strategy according to which the chaotic nature of the data creates the proper framework for the application of neural networks. Levendovszky and Kia (2012) propose a multilayer feedforward neural network for predicting the price of a EUR/USD pair, trained by using the backpropagation algorithm. Sirignano (2016) proposes a new method for training deep neural networks that try to model the joint distribution of the bid and ask depth, where a focal point is the spatial nature[11] of LOB levels. Bogoev and Karam (2016) propose the use of a single hidden-layer feedforward neural (SLFN) network for the detection of quote stuffing and momentum ignition. Dixon (2016) uses a recurrent neural network (RNN) for mid-price predictions of T-bond[12] and ES futures[13] based on ultra-high-frequency data. Rehman, Khan, and

---

[5]Adaptive weights are used for penalizing different coefficients in the l_1 penalty term.
[6]The method is the generalization of a linear regression model when the dependent variable is discrete.

[7]Panel regression models provide information on data characteristics individually, but also across both individuals over time.
[8]Market conditions of an industry sector have an impact on sellers and buyers who are related to it. Factors to consider include the number of competitors in the sector. For example, if there is a surplus, new companies may find it difficult to enter the market and remain in business.
[9]Technical indicators are mainly used for short-term price movement predictions. They are formulas based on historical data.
[10]Scalping is a type of trading strategy according to which the trader tries to make a profit for small changes in a stock.
[11]The spatial nature of this type of neural network and its gradient can be evaluated at far fewer grid points. This makes the model less computationally expensive. Furthermore, the suggested architecture can model the entire distribution in the R^d space.
[12]Treasury bond (T-bond) is a long-term fixed interest rate debt security issued by the federal government.
[13]E-mini S&P 500 (ES futures) are electronically traded futures contracts whose value is one-fifth the size of standard S&P futures.

Mahmud (2014) apply recurrent Cartesian genetic programming evolved artificial neural network (RCGPANN) for predicting five currency rates against the Australian dollar. Galeshchuk (2016) suggests that a multilayer perceptron (MLP) architecture, with three hidden layers, is suitable for exchange rate prediction. Majhi, Panda, and Sahoo (2009) use the functional link artificial neural network (FLANN) in order to predict price movements in the DJIA[14] and S&P 500[15] stock indices.

Deep belief networks are employed by Sharang and Rao (2015) to design a medium-frequency portfolio trading strategy. Hallgren and Koski (2016) use continuous-time Bayesian networks (CTBNs) for causality detection. They apply their model on tick-by-tick high-frequency foreign exchange (FX) EUR/USD data using a Skellam process.[16] Sandoval and Hernández (2015) create a profitable trading strategy by combining hierarchical hidden Markov models (HHMM), where they consider wavelet-based LOB information filtering. In their work, they also consider a two-layer feedforward neural network in order to classify the upcoming states. They nevertheless report limitations in the neural network in terms of the volume of the input data.

2.3 | Maximum margin and reinforcement learning

Palguna and Pollak (2016) use nonparametric methods on features derived from LOB, which are incorporated into order execution strategies for mid-price prediction. In the same direction, Kercheval and Zhang (2015) employ a multi-class SVM for mid-price and price spread crossing prediction. Han et al. (2015) base their research on Kercheval and Zhang by using multi-class SVM for mid-price movement prediction. More precisely, they compare multi-class SVM (exploring linear and RBF kernels) to decision trees using bagging for variance reduction.

Kim (2001) uses input/output hidden Markov models (IOHMMs) and reinforcement learning (RL) in order to identify the order flow distribution and market-making strategies, respectively. Yang et al. (2015) apply apprenticeship learning[17] methods, like linear inverse reinforcement learning (LIRL) and Gaussian process IRL (GPIRL), to recognize traders or algorithmic trades based on the observed limit orders. Chan and Shelton (2001) use RL for market-making strategies, where experiments based on a Monte Carlo simulation and a state–action–reward–state–action (SARSA) algorithm test the efficacy of their policy. In the same vein, Kearns and Nevmyvaka (2013) implement RL for trade execution optimization in lit and dark pools. Especially in the case of dark pools, they apply a censored exploration algorithm to the problem of smart order routing (SOR). Yang, Paddrik, Hayes, Todd, Kirilenko, Beling, and Scherer (2012) examine an IRL algorithm for the separation of HFT strategies from other algorithmic trading activities. They also apply the same algorithm to the identification of manipulative HFT strategies (i.e., spoofing). Felker, Mazalov, and Watt (2014) predict changes in the price of quotes from several exchanges. They apply feature-weighted Euclidean distance to the centroid of a training cluster. They calculate this type of distance to the centroid of a training cluster where feature selection is taken into consideration because several exchanges are included in their model.

2.4 | Additional methods for HFT and LOB

HFT and LOB research activity also covers topics like the optimal submission strategies of bid and ask orders, with a focus on the inventory risk that stems from an asset's value uncertainty, as in the work of Avellaneda and Stoikov (2008). Chang (2015) models the dynamics of LOB by using a Bayesian inference of the Markov chain model class, tested on high-frequency data. An and Chan (2017) suggest a new stochastic model that is based on independent compound Poisson processes of the order flow. Talebi, Hoang, and Gavrilova (2014) try to predict trends in the FX market by employing a multivariate Gaussian classifier (MGC) combined with Bayesian voting. Fletcher, Hussain, and Shawe-Taylor (2010) examine trading opportunities for the EUR/USD where the price movement is based on multiple kernel learning (MKL). More specifically, the authors utilize SimpleMKL and the more recent LPBoost-MKL methods for training a multi-class SVM. Christensen and Woodmansey (2013) develop a classification method based on the Gaussian kernel in order to identify iceberg[18] orders for GLOBEX.

Maglaras, Moallemi, and Zheng (2015) consider the LOB as a multi-class queueing system in order to solve the problem placement of limit and market order placements. Mankad, Michailidis, and Kirilenko (2013) apply a static plaid clustering technique to synthetic data in order to

---

[14]The Dow Jones Industrial Average (DJIA) is the price-weighted average of the 30 largest, publicly owned US companies.
[15]S&P 500 is the index that provides a summary of the overall market by tracking some of the 500 top stocks in US stock market.
[16]A Skellam process is defined as $S(t) = N^{(1)}(t) - N^{(2)}(t)$, $t \geqslant 0$, where $N^{(1)}(t)$ and $N^{(2)}(t)$ are two independent homogeneous Poisson processes.
[17]Motivation for apprenticeship learning is to use IRL techniques to learn the reward function and then use this function in order to define a Markov decision problem (MDP).

[18]Iceberg order is the conditional request made to the broker to sell or buy a larger quantity of the stock, but in smaller predefined quantities.

classify the different types of trades. Aramonte, Schindler, and Rosen (2013) show that the information asymmetry in a high-frequency environment is crucial.

Vella and Ng (2016) use higher-order fuzzy systems (i.e., an adaptive neuro-fuzzy inference system) by introducing T2 fuzzy sets, where the goal is to reduce microstructure noise in the HFT sphere. Abernethy and Kale (2013) apply market-maker strategies based on low-regr et algorithms for the stock market. Almgren and Lorenz (2006) explain price momentum by modeling Brownian motion with a drift whose distribution is updated based on Bayesian inference. Næs and Skjeltorp (2006) show that the order book slope measures the elasticity of supplied quantity as a function of asset prices related to volatility, trading activity, and an asset's dispersion beliefs.

3 | THE LOB DATASET

In this section, we describe in detail our dataset collected in order to facilitate future research in LOB-based HFT. We start by providing a detailed description of the data in Section 3.1. Data processing steps are followed in order to extract message books and LOBs, as described in Section 3.2.

3.1 | Data description

Extracting information from the ITCH flow, and without relying on third-party data providers, we analyze stocks from different industry sectors for 10 full days of ultra-high-frequency intra-day data. The data provide information regarding trades against hidden orders. Coherently, the nondisplayable hidden portions of the total volume of a so-called iceberg order are not accessible from the data. Our ITCH feed data is day specific and market wide, which means that we deal with one file per day with data over all the securities. Information (block A in Figure 1) regarding (i) messages for order submissions, (ii) trades, and (iii) cancellations is included. For each order, its type (buy/sell), price, quantity, and exact time stamp on a millisecond basis is available. In addition, (iv) administrative messages (i.e., trading halts or basic security data), (v) event controls (i.e., start and ending of trading days, states of market segments), and (vi) net order imbalance indicators are also included.

The next step is the development and implementation of a C++ converter to extract all the information relevant to a given security. We perform the same process for five stocks traded on the Nasdaq OMX Nordic at the Helsinki
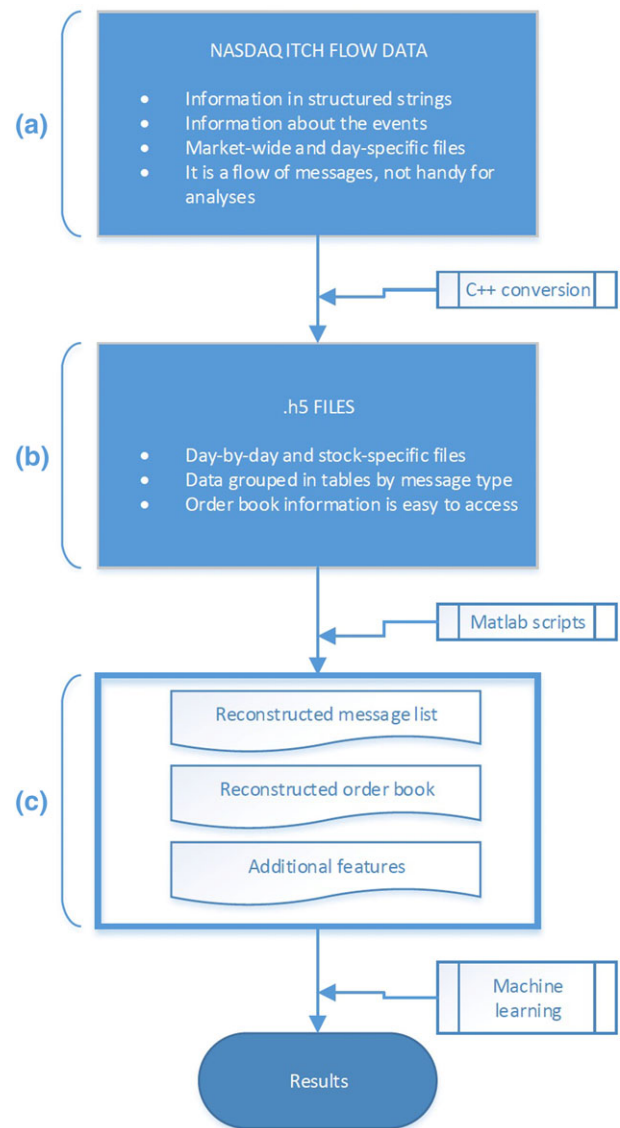
FIGURE 1 Data processing flow [Colour figure can be viewed at wileyonlinelibrary.com]

exchange from June 1, 2010 to June 14, 2010.[19] These data are stored in a Linux cluster. Information related to the five stocks is illustrated in Table 1. The selected stocks[20] are traded in one exchange (Helsinki) only. By choosing only one stock market exchange, the trader has the advantage of avoiding issues associated with fragmented markets. In the case of fragmented markets, the limit orders for

---

[19]There have been about 23,000 active order books, the vast majority of which are very illiquid, show sporadic activity, and correspond to little and noisy data.

[20]The choice is driven by the necessity of having a sufficient amount of data for training (this excludes illiquid stocks) while covering different industry sectors. These five selected stocks (see Table 1), which aggregate input message list and order book data for feature extraction, are about 4 GB; RTRKS was suspended from trading and delisted from the Helsinki exchange on November 20, 2014.

a given asset are spread between several exchanges, posing problems from empirical data analysis (O'Hara & Ye, 2011).

The Helsinki Stock Exchange, operated by Nasdaq Nordic, is a pure electronic limit order market. The ITCH feed keeps a record of all the events, including those that take place outside active trading hours. At the Helsinki exchange, the trading period goes from 10:00 to 18:25 (local time, UTC/GMT +2 hours). However, in the ITCH feed, we observe several records outside those trading hours. In particular, we consider the regulated auction period before 10:00, which is used to set the opening price of the day (the so-called pre-opening period) before trading begins. This is a structurally different mechanism following different rules with respect to the order book flow during trading hours. Similarly, another structural break in the order book's dynamics is due to the different regulations that are in force between 18:25 and 18:30 (the so-called post-opening period). As a result, we retain exclusively the events occurring between 10:30 and 18:00. More information related to the above-mentioned issues can be found in Siikanen, Kanniainen, and Luoma 2017 and (Siikanen, Kanniainen, & Valli, 2017). Here, the order book is expected to have comparable dynamics with no biases or exceptions caused by its proximity to the market opening and closing times.

3.2 | Limit order and message books

Message and LOBs are processed for each of the 10 days for the five stocks. More specifically, there are two types of messages that are particularly relevant here: (i) "add order messages," corresponding to order submissions; and (ii) "modify order messages," corresponding to updates on the status of existing orders through order cancellations and order executions. Example message[21] and limit order[22] books are illustrated in Tables 2 and Table 3, respectively.

LOB is a centralized trading method that is incorporated by the majority of exchanges globally. It aggregates the limit orders of both sides (i.e., the ask and bid sides) of the stock market (e.g., the Nordic stock market). LOB matches every new event type according to several characteristics. Event types and LOB characteristics describe the current state of this matching engine. Event types can be executions, order submissions, and order cancellations. Characteristics of LOB are the resolution parameters (Gould, Porter, Williams, McDonald, Fenn, & Howison, 2013), which are the tick size π (i.e., the smallest permissi-

ble price between different orders), and the lot size σ (i.e., the smallest amount of a stock that can be traded and is defined as $\{k\sigma|k = 1, 2, \dots\}$). Order inflow and resolution parameters will formulate the dynamics of the LOB, whose current state will be identified by the state variable of four elements $(s_t^b, q_t^b, s_t^a, q_t^a), t \geq 0$, where s_t^b (s_t^b) is the best bid (ask) price and q_t^b (q_t^a) is the size of the best bid (ask) level at time t.

In our data, timestamps are expressed in milliseconds based on 1 Jan 1970 format and shifted by three hours with respect to Eastern European Time (in the data, the trading day goes from 7:00 to 15:25). ITHC feed prices are recorded up to 4 decimal places and, in our data, the decimal point is removed by multiplying the price by 10,000, where currency is in euros for the Helsinki exchange. The tick size, defined as the smallest possible gap between the ask and bid prices, is 1 cent. Similarly, order quantities are constrained to integers greater than one.

3.3 | Data availability and distribution

In compliance with Nasdaq OMX agreements, the normalized feature dataset is made available to the research community.[23] The open-access version of our data has been normalized in order to prevent reconstruction of the original Nasdaq data.

3.4 | Experimental protocol

In order to make our dataset a benchmark that can be used for the evaluation of HTF methods based on LOB information, the data are accompanied by the following experimental protocol. We develop a day-based prediction framework following an anchored forward cross-validation format. More specifically, the training set is increased by 1 day in each fold and stops after $n - 1$ days (i.e., after 9 days in our case where $n = 10$). On each fold, the test set corresponds to 1 day of data, which moves in a rolling window format. The experimental setup is illustrated in Figure 2. Performance is measured by calculating the mean accuracy, recall, precision, and F1 score over all folds, as well as the corresponding standard deviation. We measure our results based on these metrics, which are defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (1)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (2)$$

---

[21]A sample from FI0009002422 on June 1, 2010.
[22]A sample from FI0009002422 on June 1, 2010.

[23]We thank Ms. Sonja Salminen at Nasdaq for her support and help.

TABLE 1 Stocks used in the analysis

ID	ISIN code	Company	Sector	Industry
KESBV	FI0009000202	Kesko Oyj	Consumer Defensive	Grocery Stores
OUT1V	FI0009002422	Outokumpu Oyj	Basic Materials	Steel
SAMPO	FI0009003305	Sampo Oyj	Financial Services	Insurance
RTRKS	FI0009003552	Rautaruukki Oyj	Basic Materials	Steel
WRT1V	FI0009000727	Wärtsilä Oyj	Industrials	Diversified Industrials

TABLE 2 Message list example

Timestamp	ID	Price	Quantity	Event	Side
1275386347944	6505727	126200	400	Cancellation	Ask
1275386347981	6505741	126500	300	Submission	Ask
1275386347981	6505741	126500	300	Cancellation	Ask
1275386348070	6511439	126100	17	Execution	Bid
1275386348070	6511439	126100	17	Submission	Bid
1275386348101	6511469	126600	300	Cancellation	Ask

TABLE 3 Order book example

| | | | Level 1 | | | | Level 2 | | | | … |
| | | | Ask | | Bid | | Ask | | Bid | | |
Timestamp	Mid-price	Spread	Price	Quantity	Price	Quantity	Price	Quantity	Price	Quantity	
1275386347944	126200	200	126300	300	126100	17	126400	4765	126000	2800	…
1275386347981	126200	200	126300	300	126100	17	126400	4765	126000	2800	…
1275386347981	126200	200	126300	300	126100	17	126400	4765	126000	2800	…
1275386348070	126050	100	126100	291	126000	2800	126200	300	125900	1120	…
1275386348070	126050	100	126100	291	126000	2800	126200	300	125900	1120	…
1275386348101	126050	100	126100	291	126000	2800	126200	300	125900	1120	…

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{4}$$

where TP and TF represent the true positives and true negatives, respectively, of the mid-price prediction label compared with the ground truth, where FP and FN represents the false positives and false negatives, respectively. From among the above metrics, we focus on the $F1$ score performance. The main reason that we focus on $F1$ score is based on its ability only to be affected in one direction of skew distributions, in the case of unbalanced classes like ours. On the contrary, accuracy cannot differentiate between the number of correct labels (i.e., related to mid-price movement direction prediction) of different classes where the other three metrics can separate the correct labels among different classes, with $F1$ being the harmonic mean of Precision and Recall.

We follow an event-based inflow, as used in Li, et al. (2016). This is due to the fact that events (i.e., orders, executions, and cancellations) do not follow a uniform inflow rate. Time intervals between two consecutive events can vary from milliseconds to several minutes of difference. Event-based data representation avoids issues related to such big differences in data flow. As a result, each of our representations is a vector that contains information for 10 consecutive events. Event-based data description leads to a dataset of approximately half a million representations (i.e., 394,337 representations). We represent these events using the 144-dimensional representation proposed recently by Kercheval and Zhang (2015), formed by three types of features: (a) the raw data of a 10-level limit order containing price and volume values for bid and ask orders; (b) features describing the state of the LOB, exploiting past information; and (c) features describing the information edge in the raw data by taking time into account. Derivations of time, stock price, and volume are calculated for short and long-term projections. More specifically, types in features $u_7, u_8,$ and u_9 are: *trades, orders, cancellations, deletion, execution of a visible limit order*, and *execution of a hidden limit order*. Expressions used for calculating these features are provided in Table 4. One limitation of the adopted features
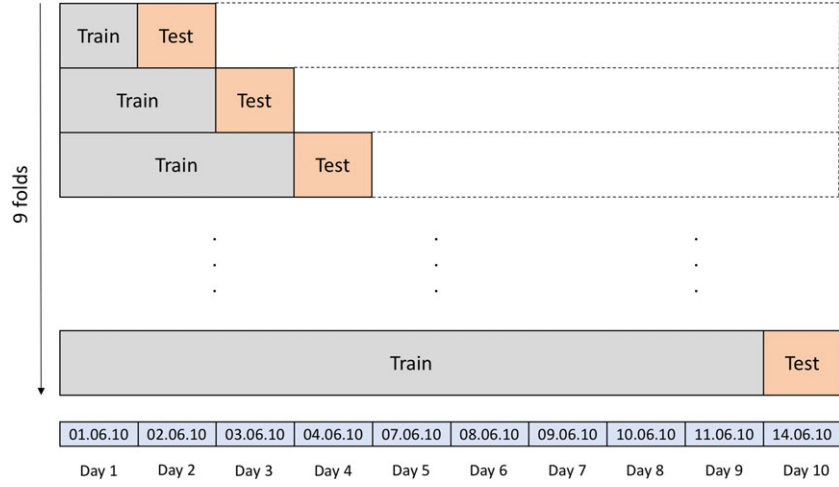
FIGURE 2 Experimental setup framework [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 4 Feature sets

Feature set	Description	Details
Basic	$u_1 = \{P_i^{ask}, V_i^{ask}, P_i^{bid}, V_i^{bid}\}_{i=1}^n$	$10(= n)$-level LOB data
Time-insensitive	$u_2 = \{(P_i^{ask} - P_i^{bid}), (P_i^{ask} + P_i^{bid})/2\}_{i=1}^n$	Spread & Mid-price
	$u_3 = \{P_n^{ask} - P_1^{ask}, P_1^{bid} - P_n^{bid}, \|P_{i+1}^{ask} - P_i^{ask}\|, \|P_{i+1}^{bid} - P_i^{bid}\|\}_{i+1}^n$	Price differences
	$u_4 = \left\{ \frac{1}{n}\sum_{i=1}^n P_i^{ask}, \frac{1}{n}\sum_{i=1}^n P_i^{bid}, \frac{1}{n}\sum_{i=1}^n V_i^{ask}, \frac{1}{n}\sum_{i=1}^n V_i^{bid} \right\}$	Price & Volume means
	$u_5 = \left\{ \sum_{i=1}^n (P_i^{ask} - P_i^{bid}), \sum_{i=1}^n (V_i^{ask} - V_i^{bid}) \right\}$	Accumulated differences
Time-sensitive	$u_6 = \{ dP_i^{ask}/dt, dP_i^{bid}/dt, dV_i^{ask}/dt, dV_i^{bid}/dt \}_{i=1}^n$	Price & Volume derivation
	$u_7 = \{ \lambda_{\Delta t}^1, \lambda_{\Delta t}^2, \lambda_{\Delta t}^3, \lambda_{\Delta t}^4, \lambda_{\Delta t}^5, \lambda_{\Delta t}^6 \}$	Average intensity per type
	$u_8 = \left\{ \mathbf{1}_{\lambda_{\Delta t}^1 > \lambda_{\Delta_T}^1}, \mathbf{1}_{\lambda_{\Delta t}^2 > \lambda_{\Delta_T}^2}, \mathbf{1}_{\lambda_{\Delta t}^3 > \lambda_{\Delta_T}^3}, \mathbf{1}_{\lambda_{\Delta t}^4 > \lambda_{\Delta_T}^4}, \mathbf{1}_{\lambda_{\Delta t}^5 > \lambda_{\Delta_T}^5}, \mathbf{1}_{\lambda_{\Delta t}^6 > \lambda_{\Delta_T}^6} \right\}$	Relative intensity comparison
	$u_9 = \{ d\lambda^1/dt, d\lambda^2/dt, d\lambda^3/dt, d\lambda^4/dt, d\lambda^5/dt, d\lambda^6/dt \}$	Limit activity acceleration

is the lack of information related to order flow (i.e., the sequence of order book messages). However, as can be seen in the Results Section 6, the baselines achieve relatively good performance and therefore we leave the introduction of extra features that can enhance performance to future research.

We provide three sets of data, each created by following a different data normalization strategy—that is, z-score, min–max, and decimal precision normalization—for every i data sample. Z-score, in particular, is the normalization process through which we subtract the mean from our input data for each feature separately and divide by the standard deviation of the given sample:

$$\mathbf{x}_i^{(z\text{-score})} = \frac{\mathbf{x}_i - \frac{1}{N}\sum_{j=1}^N \mathbf{x}_j}{\sqrt{\frac{1}{N}\sum_{j=1}^N (\mathbf{x}_j - \bar{\mathbf{x}})^2}}, \quad (5)$$

where $\bar{\mathbf{x}}$ denotes the mean vector, as appears in Equation 5. On the other hand, min–max scaling, as described by

$$\mathbf{x}_i^{(MM)} = \frac{\mathbf{x}_i - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}}, \quad (6)$$

is the process of subtracting the minimum value from each feature and dividing it by the difference between the maximum and minimum value of that feature sample. The third scaling setup is the decimal precision approach. This normalization method is based on moving the decimal points of each of the feature values. Calculations follow the absolute value of each feature sample:

$$\mathbf{x}_i^{(DP)} = \frac{\mathbf{x}_i}{10^k}, \quad (7)$$

where k is the integer that will give us the maximum value for $\|\mathbf{x}_{DP}\| < 1$.

Having defined the event representations, we use five different projection horizons for our labels. Each of these

TABLE 5 HFT dataset examples

	Dataset	Public available	Unit time	Period	Asset class / No. of stocks	Size	Annotations
1	Dukascopy	✓	ms	Up to date	Various	~20,000 events/day	✕
2	truefx	✓	ms	Up to date	15 FX pairs	~300,000 events/day	✕
3	Nasdaq	AuR	ms	2008-09	Equity / 120	—	✕
4	Nasdaq	AuR	ms	10/07 & 06/08	Equity / 500	~55,000 events/day	✕
5	Nasdaq	✕	ms	—	Equity / 5	2,000 data points	✕
6	Euronext	AuR	—	—	Several products	—	✕
7	Nasdaq	✕	ns	01/14-08/15	Equity / 489	50 TB	✕
8	Our–Nasdaq	✓	ms	01-14/06/10	Equity / 5	4 M samples	✓

horizons portrays a different future projection interval of the mid-price movement (i.e., upward, downward, and stationary mid-price movement). More specifically, we extract labels based on short-term and long-term, event-based, relative changes for the next 1, 2, 3, 5, and 10 events for our representations dataset.

Our labels describe the percentage change of the mid-price, which is calculated as follows:

$$l_i^{(j)} = \frac{\frac{1}{k} \sum_{j=i+1}^{i+k} m_j - m_i}{m_i}, \qquad (8)$$

where m_j is the future mid-price ($k = 1, 2, 3, 5,$ or 10 next events in our representations) and m_i is the current mid-price. The extracted labels are based on a threshold for the percentage change of 0.002. For percentage changes equal to or greater than 0.002, we use label 1. For percentage change that varies from -0.00199 to 0.00199, we use label 2, and, for percentage change smaller or equal to -0.002, we use label 3.

4 | EXISTING DATASETS DESCRIBED IN THE LITERATURE

In this section, we list existing HFT datasets described in the literature and provide qualitative and quantitative comparisons to our dataset. The following works mainly focus on datasets that are related to machine learning methods.

There are mainly three sources of data from which a high-frequency trader can choose. The first option is the use of publicly available data (e.g., (1) Dukascopy and (2) truefx), where no prior agreement is required for data acquisition. The second option is publicly available data upon request for academic purposes, which can be found in (3) Brogaard, Hendershott, and Riordan (2014), (4) Hasbrouck and Saar (2013), (5) De Winne and D'hondt 2007, Detollenaere and D'hondt (2017), and Carrion (2013). Finally, the third and most common option is data through

platforms requiring a subscription fee, like those in (6) Kercheval and Zhang (2015); Li et al. (2016), and (7) Sirignano (2016). Existing data sources and characteristics are listed in Table 5.

In particular, the datasets are at a millisecond resolution, except for number 6 in the table. Access to various asset classes including FX, commodities, indices, and stocks is also provided. To the best of our knowledge, there is no available literature based on this type of dataset for equities. Another source of free tick-by-tick historical data is the truefx.com site, but the site provides data only for the FX market for several pairs of currencies at a millisecond resolution. The data contain information regarding timestamps (in millisecond resolution) and bid and ask prices. Each of these .csv files contains approximately 200,000 events per day. This type of data is used in a mean-reverting jump-diffusion model, as presented in Suwanpetai (2016).

There is a second category of datasets available upon request (AuR), as seen in Hasbrouck and Saar (2013). In this paper, the authors use the Nasdaq OMX ITCH for two periods: October 2007 and June 2008. For that period, they run samples at 10-minute intervals for each day where they set a cutoff mechanism for available messages per period.[24] The main disadvantage of uniformly sampling HFT data is that the trader loses vital information. Events come randomly, with inactive periods varying from a few milliseconds to several minutes or hours. In our work, we overcome this challenge by considering the information based on event inflow, rather than equal time sampling. Another example of data that is available only for academic purposes is Brogaard et al. (2014). The dataset contains information regarding timestamps, price, and buy–sell side prices but no other details related to daily events or feature vectors. Hasbrouck and Saar provide a detailed description of their Nasdaq OMX ITCH data, which is not directly accessible for testing and comparison with their

---

[24]The authors provide a threshold, which is based on 250 events per 10-minute sample interval.

baselines. They use these data to applying low-latency strategies based on measures that capture links between submissions, cancellations, and executions. De Winne and D'hondt (2007) and Detollenaere and D'hondt (2017) use similar datasets from Euronext for LOB construction. They specify that their dataset is available upon request from the provider. What is more, the data provider supplies details regarding the LOB construction by the user. Our work fills that gap since our dataset provides the full LOB depth and it is ready for use and comparison with our baselines.

The last category of dataset has dissemination restrictions. An example is the paper by Kercheval and Zhang (2015), where the authors are trying to predict the mid-price movement by using machine learning (i.e., SVM). They train their model with a very small number of samples (i.e., 4,000 samples). The HFT activity can produce a huge volume of trading events daily, as our database does with 100,000 daily events for only one stock. Moreover, the datasets in Kercheval and Zhang and in Sirignano (2016) are not publicly available, which makes comparison with other methods impossible. In the same direction, we also add works such as Hasbrouck (2009), Kalay, Sade, and Wohl (2004), and Kalay, Wei, and Wohl (2002), which utilize TAQ and Tel Aviv stock exchange datasets (not for machine learning methods), and require subscription.

5 | BASELINES

In order to provide performance baselines for our new dataset of HFT with LOB data, we conducted experiments with two regression models using the data representations described in Section 3.4. Details on the models used are provided in Sections 5.1 and 5.2. The baseline performances are provided in Section 6.

5.1 | Ridge regression (RR)

Ridge regression defines a linear mapping, expressed by the matrix $\mathbf{W} \in \mathbb{R}^{D \times C}$, that optimally maps a set of vectors $\mathbf{x}_i \in \mathbb{R}^D$, $i = 1, \cdots, N$ to another set of vectors (noted as target vectors) $\mathbf{t}_i \in \mathbb{R}^C$, $i = 1, \cdots, N$, by optimizing the following criterion:

$$\mathbf{W}^* = \arg \min_{\mathbf{W}} \sum_{i=1}^N \|\mathbf{W}^T \mathbf{x}_i - \mathbf{t}_i\|_2^2 + \lambda \|\mathbf{W}\|_F^2, \quad (9)$$

or using a matrix notation:

$$\mathbf{W}^* = \arg \min_{\mathbf{W}} \|\mathbf{W}^T \mathbf{X} - \mathbf{T}\|_F^2 + \lambda \|\mathbf{W}\|_F^2. \quad (10)$$

In the above, $\mathbf{X} = [\mathbf{x}_i, \dots, \mathbf{x}_N]$ and $\mathbf{T} = [\mathbf{t}_i, \dots, \mathbf{t}_N]$ are matrices formed by the samples $\mathbf{x}_i$ and $\mathbf{t}_i$ as columns, respectively.

In our case, each sample $\mathbf{x}_i$ corresponds to an event, represented by a vector (with $D = 144$), as described in Section 3.4. For the three-class classification problems in our dataset, the elements of vectors $\mathbf{t}_i \in \mathbb{R}^C$ ($C = 3$ in our case) take values equal to $t_{ik} = 1$, if $\mathbf{x}_i$ belongs to class k, and if $t_{ik} = -1$ otherwise. The solution of Equation 10 is given by

$$\mathbf{W} = \mathbf{X}(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{T}^T, \quad (11)$$

or

$$\mathbf{W} = (\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I})^{-1} \mathbf{X}\mathbf{T}^T, \quad (12)$$

where $\mathbf{I}$ is the identity matrix of appropriate dimensions. Here, we should note that, in our case, where the size of the data is large, $\mathbf{W}$ should be computed using Equation 12, since the calculation of Equation 11 is computationally very expensive.

After the calculation of $\mathbf{W}$, a new (test) sample $\mathbf{x} \in \mathbb{R}^D$ is mapped on its corresponding representation in space $\mathbb{R}^C$—that is, $\mathbf{o} = \mathbf{W}^T \mathbf{x}$—and is classified according to the maximum value of its projection:

$$l_\mathbf{x} = \arg \max_k o_k. \quad (13)$$

5.2 | SLFN network-based nonlinear regression

We also test the performance of a nonlinear regression model. Since the application of kernel-based regression is computationally too intensive for the size of our data, we use an SLFN (Figure 3) network-based regression model. Such a model is formed as follows.

For fast network training, we train our network based on the algorithm proposed in Huang, Zhou, Ding, and Zhang (2012), Zhang, Kwok, and Parvin (2009), and Iosifidis, Tefas, and Pitas (2017). This algorithm is formed by
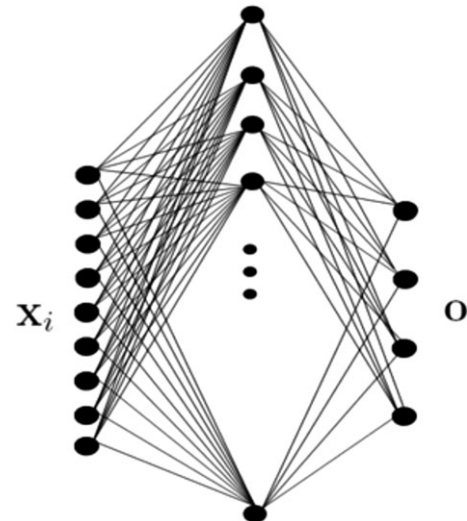

FIGURE 3 SLFN

two processing steps. In the first step, the network's hidden layer weights are determined either randomly (Huang, Zhou, Ding, & Zhang, 2012) or by applying clustering on the training data. We apply *K*-means clustering in order to determine *K* prototype vectors, which are subsequently used as the network's hidden layer weights.

Having determined the network's hidden layer weights $\mathbf{V} \in \mathbb{R}^{D \times K}$, the input data $\mathbf{x}_i, i = 1, \ldots, N$ are nonlinearly mapped to vectors $\mathbf{h}_i \in \mathbb{R}^K$, expressing the data representations in the feature space determined by the network's hidden layer outputs $\mathbb{R}^K$. We use the radial basis function—that is, $\mathbf{h}_i = \varphi_{\mathrm{RBF}}(\mathbf{x}_i)$—calculated in an element-wise manner, as follows:

$$h_{ik} = \exp\left(\frac{\|\mathbf{x}_i - \mathbf{v}_k\|_2^2}{2\sigma^2}\right), \quad k = 1, \cdots, K, \qquad (14)$$

where σ is a hyperparameter denoting the spread of the RBF neuron and $\mathbf{v}_k$ corresponds to the *k*th column of $\mathbf{V}$.

The network's output weights $\mathbf{W} \in \mathbb{R}^{K \times C}$ are subsequently determined by solving for

$$\mathbf{W}^* = \arg \min_{\mathbf{W}} \|\mathbf{W}^T \mathbf{H} - \mathbf{T}\|_F^2 + \lambda \|\mathbf{W}\|_F^2, \qquad (15)$$

where $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_N]$ is a matrix formed by the network's hidden layer outputs for the training data and $\mathbf{T}$ is a matrix formed by the network's target vectors $\mathbf{t}_i, i = 1, \ldots, N$ as defined in Section 5.1. The network's output weights are given by

$$\mathbf{W} = \left(\mathbf{H}\mathbf{H}^T + \lambda \mathbf{I}\right)^{-1} \mathbf{H}\mathbf{T}^T. \qquad (16)$$

After calculation of the network parameters $\mathbf{V}$ and $\mathbf{W}$, a new (test) sample $\mathbf{x} \in \mathbb{R}^D$ is mapped on its corresponding representations in spaces $\mathbb{R}^K$ and $\mathbb{R}^C$; that is, $\mathbf{h} = \varphi_{\mathrm{RBF}}(\mathbf{x})$ and $\mathbf{o} = \mathbf{W}^T \mathbf{h}$, respectively. It is classified according to the maximal network output:

$$l_{\mathbf{x}} = \arg \max_k o_k. \qquad (17)$$

6 ⏐ RESULTS

In our first set of experiments, we have applied two supervised machine learning methods, as described in Sections 5.1 and 5.2, on a dataset that does not include the auction period. Results with the auction period will also be available. Since there is not a widely adopted experimental protocol for these datasets, we provide information for the five different label scenarios under the three normalization setups.

The tables in this section provide details regarding the results of experiments conducted on raw data and three different normalization setups. We present these results, for our baseline models, in order to give insight into the preprocessing step for a dataset like ours, to examine the strength of the predictability of the projected time horizon, and to understand the implications of the suggested methods. Data normalization can significantly improve the metric's performance in combination with the use of the right classifier. More specifically, we measure the predictability power of our models via the performance of the metrics of accuracy, precision, recall, and *F*1 score. For instance, Table 6 presents the results based on raw data (i.e., no data decoding), and in the case of the linear classifier RR and label 5 (i.e., the 5th mid-price event as predicted horizon), we achieve an *F*1 score of 40%, where as in Table 7 (i.e., the *Z*-score data decoding method), Table 8 (i.e., min–max data decoding method), and Table 9 (i.e., the decimal precision decoding method), we achieve 43%, 42%, and 40%, respectively. This shows

TABLE 6 Results based on unfiltered representations

Label	$\mathrm{RR}_{\mathbf{Accuracy}}$	$\mathrm{RR}_{\mathbf{Precision}}$	$\mathrm{RR}_{\mathbf{Recall}}$	RR_{F1}
1	0.637 ± 0.055	0.505 ± 0.145	0.337 ± 0.003	0.268 ± 0.014
2	0.555 ± 0.064	0.504 ± 0.131	0.376 ± 0.023	0.320 ± 0.050
3	0.489 ± 0.061	0.423 ± 0.109	0.397 ± 0.031	0.356 ± 0.070
5	0.429 ± 0.049	0.402 ± 0.113	0.425 ± 0.038	0.400 ± 0.093
10	0.453 ± 0.054	0.400 ± 0.105	0.400 ± 0.030	0.347 ± 0.066
Label	$\mathrm{SLFN}_{\mathbf{Accuracy}}$	$\mathrm{SLFN}_{\mathbf{Precision}}$	$\mathrm{SLFN}_{\mathbf{Recall}}$	SLFN_{F1}
1	0.636 ± 0.055	0.299 ± 0.075	0.335 ± 0.002	0.262 ± 0.015
2	0.536 ± 0.069	0.387 ± 0.132	0.345 ± 0.009	0.260 ± 0.035
3	0.473 ± 0.074	0.334 ± 0.080	0.357 ± 0.005	0.270 ± 0.021
5	0.381 ± 0.038	0.342 ± 0.058	0.370 ± 0.020	0.327 ± 0.043
10	0.401 ± 0.039	0.284 ± 0.102	0.356 ± 0.020	0.290 ± 0.070

TABLE 7 Results based on Z-score normalization

Label	$RR_{Accuracy}$	$RR_{Precision}$	RR_{Recall}	RR_{F1}
1	0.480 ± 0.040	0.418 ± 0.021	0.435 ± 0.029	0.410 ± 0.022
2	0.498 ± 0.052	0.444 ± 0.025	0.443 ± 0.031	0.440 ± 0.031
3	0.463 ± 0.045	0.438 ± 0.027	0.437 ± 0.033	0.433 ± 0.034
5	0.439 ± 0.042	0.436 ± 0.028	0.433 ± 0.028	0.427 ± 0.041
10	0.429 ± 0.046	0.429 ± 0.028	0.429 ± 0.043	0.416 ± 0.044
Label	$SLFN_{Accuracy}$	$SLFN_{Precision}$	$SLFN_{Recall}$	$SLFN_{F1}$
1	0.643 ± 0.056	0.512 ± 0.037	0.366 ± 0.019	0.327 ± 0.046
2	0.556 ± 0.066	0.550 ± 0.029	0.378 ± 0.011	0.327 ± 0.030
3	0.512 ± 0.069	0.497 ± 0.024	0.424 ± 0.047	0.389 ± 0.082
5	0.473 ± 0.036	0.468 ± 0.024	0.464 ± 0.028	0.459 ± 0.031
10	0.477 ± 0.048	0.453 ± 0.056	0.432 ± 0.025	0.410 ± 0.040

TABLE 8 Results Based on min–max normalization

Label	$RR_{Accuracy}$	$RR_{Precision}$	RR_{Recall}	RR_{F1}
1	0.637 ± 0.054	0.499 ± 0.118	0.339 ± 0.005	0.272 ± 0.015
2	0.561 ± 0.063	0.467 ± 0.117	0.400 ± 0.028	0.368 ± 0.060
3	0.492 ± 0.070	0.428 ± 0.111	0.400 ± 0.030	0.357 ± 0.072
5	0.437 ± 0.048	0.419 ± 0.078	0.429 ± 0.043	0.417 ± 0.063
10	0.452 ± 0.054	0.421 ± 0.110	0.399 ± 0.028	0.348 ± 0.066
Label	$SLFN_{Accuracy}$	$SLFN_{Precision}$	$SLFN_{Recall}$	$SLFN_{F1}$
1	0.640 ± 0.055	0.488 ± 0.104	0.348 ± 0.007	0.291 ± 0.022
2	0.558 ± 0.065	0.469 ± 0.066	0.399 ± 0.023	0.367 ± 0.050
3	0.499 ± 0.063	0.447 ± 0.068	0.410 ± 0.032	0.370 ± 0.063
5	0.453 ± 0.038	0.441 ± 0.041	0.444 ± 0.030	0.432 ± 0.050
10	0.450 ± 0.048	0.432 ± 0.070	0.406 ± 0.037	0.377 ± 0.062

TABLE 9 Results based on decimal precision normalization

Label	$RR_{Accuracy}$	$RR_{Precision}$	RR_{Recall}	RR_{F1}
1	0.638 ± 0.054	0.518 ± 0.132	0.341 ± 0.007	0.277 ± 0.018
2	0.551 ± 0.066	0.473 ± 0.118	0.372 ± 0.018	0.315 ± 0.045
3	0.490 ± 0.069	0.432 ± 0.113	0.386 ± 0.023	0.330 ± 0.059
5	0.435 ± 0.051	0.406 ± 0.115	0.430 ± 0.039	0.405 ± 0.095
10	0.451 ± 0.052	0.417 ± 0.108	0.399 ± 0.029	0.349 ± 0.067
Label	$SLFN_{Accuracy}$	$SLFN_{Precision}$	$SLFN_{Recall}$	$SLFN_{F1}$
1	0.641 ± 0.055	0.512 ± 0.027	0.351 ± 0.007	0.297 ± 0.024
2	0.565 ± 0.063	0.505 ± 0.020	0.410 ± 0.026	0.385 ± 0.054
3	0.504 ± 0.061	0.465 ± 0.032	0.421 ± 0.040	0.393 ± 0.073
5	0.457 ± 0.038	0.451 ± 0.029	0.449 ± 0.031	0.438 ± 0.046
10	0.461 ± 0.053	0.453 ± 0.036	0.420 ± 0.035	0.399 ± 0.053

that in the case of the linear classifier the suggested decoding methods did not offer any significant improvements, since the variability of the performance range is approximately 3%. On the other hand, our nonlinear classifier (i.e., SLFN) for the same projected time horizon (i.e., label 5) reacted more efficiently in the decoding process. SLFN achieves 33% for the $F1$ score for nonnormalized data, while the Z-score, min–max and decimal precision methods achieve 46%, 43%, and 43%, respectively. As a result, normalization improves the $F1$ score performance by almost 10%.

Normalization and model selection can also affect the predictability of mid-price movements over the projected time horizon. Very interesting results come to light if we try to compare the $F1$ performance over different time horizons. For instance, we can see that, regardless of the decoding method, the $F1$ score is always better for label 5 than 1, meaning that 'our models'

predictions are better further in the future. This result is significant, especially with unfiltered data and min–max and decimal precision normalizations, when $F1$ score is approximately 27%, in the case of the one-step prediction problem (label 1), and 43% in the case of the five-step problem (label 5).

Another aspect of the experimental results above stems from the pros and cons of linear and nonlinear classifiers. More specifically, the RR linear classifier performed better on the raw dataset and for the Z-score decoding method in terms of $F1$ when compared to the SLFN (i.e., nonlinear classifier). This is not the case for the last decoding methods (i.e., min–max and decimal precision), where our nonlinear classifier presents similar or better results than RR. An explanation for this $F1$ performance discrepancy is due to each of these methods′ engineering has. The RR classifier tends to be very efficient in high-dimensional problems, and these types of problems are linearly separable, in most cases. Another reason that RR can perform better when compared to a nonlinear classifier is that RR can control the complexity by penalizing the bias, via cross-validation, using the ridge parameter. On the other hand, a nonlinear classifier is prone to overfitting, which means that in some cases it offers a better degree of freedom for class separation.

7 | CONCLUSION

This paper described a new benchmark dataset formed by the Nasdaq ITCH feed data for five stocks for 10 consecutive trading days. Data representations that were exploited by order flow features were made available. We formulated five classification tasks based on mid-price movement predictions for 1, 2, 3, 5, and 10 predicted horizons. Baseline performances of two regression models were also provided in order to facilitate future research in the field. Despite the data size, we achieved an average out-of-sample performance ($F1$) of approximately 46% for both methods. These very promising results show that machine learning can effectively predict mid-price movement.

Potential avenues of research that can benefit from exploiting the provided data include: (a) prediction of the stability of the market, which is very important for liquidity providers (market makers) to make the spread, as well as for traders to increase liquidity provision (when markets can be predicted to be stable); (b) prediction on market movements, which is important for expert systems used by speculative traders; (c) identification of order book spoofing—that is, situations where markets are manipulated by limit orders. Although there is no spoofing activity information available for the provided data, the exploitation of such a large corpus of data can be used in order to identify patterns in stock markets that can be further analyzed as normal or abnormal.

ORCID

Adamantios Ntakaris iD http://orcid.org/0000-0001-6949-5337

REFERENCES

Abernethy, J., & Kale, S. (2013). Adaptive market making via online learning, *Advances in Neural Information Processing Systems* (pp. 2058–2066). Cambridge, MA: MIT Press.

Almgren, R., & Lorenz, J. (2006). Bayesian adaptive trading with a daily cycle. *Journal of Trading*, *1*(4), 38–46.

Alvim, L. G., dos Santos, C. N., & Milidiu, R. L. (2010). Daily volume forecasting using high frequency predictors. In *Proceedings of the 10Th IASTED International Conference*, Acta Press, Calgary, Canada, Vol. 674, pp. 248.

Amaya, D., Filbien, J.-Y, Okou, C., & Roch, A. F. (2015). Distilling liquidity costs from limit order books. Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstractid=2660226.

An, Y., & Chan, N. H. (2017). Short-term stock price prediction based on limit order book dynamics. *Journal of Forecasting*, *36*(5), 541–556.

Aramonte, S., Schindler, J. W., & Rosen, S. (2013). Assessing and combining financial conditions indexes. Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstractid=2976840.

Avellaneda, M., & Stoikov, S. (2008). High-frequency trading in a limit order book. *Quantitative Finance*, *8*(3), 217–224.

Bogoev, D., & Karam, A. (2016). An Empirical Detection of High Frequency Trading Strategies. (*Working Paper*). Durham, UK: Durham University.

Brogaard, J., Hendershott, T., & Riordan, R. (2014). High-frequency trading and price discovery. *Review of Financial Studies*, *27*(8), 2267–2306.

Cao, C., Hansch, O., & Wang, X. (2009). The information content of an open limit-order book. *Journal of Futures Markets*, *29*(1), 16–41.

Carrion, A. (2013). Very fast money: High-frequency trading on the NASDAQ. *Journal of Financial Markets*, *16*(4), 680–711.

Cenesizoglu, T., Dionne, G., & Zhou, X. (2014). Effects of the limit order book on price dynamics. Retrieved from https://depot.erudit.org/bitstream/003996dd/1/CIRPEE14-26.pdf.

Chan, N. T., & Shelton, C. (2001). An electronic market-maker. Retrieved from https://dspace.mit.edu/bitstream/handle/1721.1/7220/AIM-2001-005.pdf?sequence=2.

Chang, Y. L. (2015). Inferring Markov chain for modeling order book dynamics in high frequency environment. *International Journal of Machine Learning and Computing*, *5*(3), 247–251.

Christensen, H. L., & Woodmansey, R. (2013). Prediction of hidden liquidity in the limit order book of globex futures. *Journal of Trading*, 8(3), 68–95.

Creamer, G. (2012). Model calibration and automated trading agent for euro futures. *Quantitative Finance*, 12(4), 531–545.

De Winne, R., & D'hondt, C. (2007). Hide-and-seek in the market: placing and detecting hidden orders. *Review of Finance*, 11(4), 663–692.

Detollenaere, B., & D'hondt, C. (2017). Identifying expensive trades by monitoring the limit order book. *Journal of Forecasting*, 36(3), 273–290.

Dixon, M. (2016). High frequency market making with machine learning. Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstractid=2868473.

Felker, T., Mazalov, V., & Watt, S. M. (2014). Distance-based high-frequency trading. *Procedia Computer Science*, 29, 2055–2064.

Fletcher, T., Hussain, Z., & Shawe-Taylor, J. (2010). Multiple kernel learning on the limit order book. In *Proceedings of the First Workshop on Applications of Pattern Analysis*, Vol. 11, pp. 167–174.

Galeshchuk, S. (2016). Neural networks performance in exchange rate prediction. *Neurocomputing*, 172, 446–452.

Gould, M. D., Porter, M. A., Williams, S., McDonald, M., Fenn, D. J., & Howison, S. D. (2013). Limit order books. *Quantitative Finance*, 13(11), 1709–1742.

Hallgren, J., & Koski, T. (2016). Testing for causality in continuous time Bayesian network models of high-frequency data. arXiv preprint retrieved from https://arxiv.org/abs/1601.06651.

Han, J., Hong, J., Sutardja, N., & Wong, S. F. (2015). Machine Learning Techniques for Price Change Forecast Using the Limit Order Book Data. (*Working Paper*). Berkeley, CA: University of California, Berkeley.

Hasbrouck, J. (2009). Trading costs and returns for US equities: Estimating effective costs from daily data. *Journal of Finance*, 64(3), 1445–1477.

Hasbrouck, J., & Saar, G. (2013). Low-latency trading. *Journal of Financial Markets*, 16(4), 646–679.

Huang, G.-B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 42(2), 513–529.

Iosifidis, A., Tefas, A., & Pitas, I. (2017). Approximate kernel extreme learning machine for large scale data classification. *Neurocomputing*, 219, 210–220.

Kalay, A., Sade, O., & Wohl, A. (2004). Measuring stock illiquidity: An investigation of the demand and supply schedules at the TASE. *Journal of Financial Economics*, 74(3), 461–486.

Kalay, A., Wei, L., & Wohl, A. (2002). Continuous trading or call auctions: Revealed preferences of investors at the Tel Aviv stock exchange. *Journal of Finance*, 57(1), 523–542.

Kearns, M., & Nevmyvaka, Y. (2013). Machine Learning for Market Microstructure and High Frequency Trading. In D. Easley, M. López De Prado, & M. O'Hara (Eds.), *High Frequency Trading: New Realities for Traders, Markets and Regulators*. London, UK: Risk Books.

Kercheval, A. N., & Zhang, Y. (2015). Modelling high-frequency limit order book dynamics with support vector machines. *Quantitative Finance*, 15(8), 1315–1329.

Kim, A. J. (2001). Input/Output Hidden Markov Models for Modeling Stock Order Flows. (*Technical Report No. 1370*). Cambridge, MA: MITAI Laboratory.

Levendovszky, J., & Kia, F. (2012). Prediction based-high frequency trading on financial time series. *Periodica Polytechnica: Electrical Engineering and Computer Science*, 56(1), 29–34.

Li, X., Xie, H., Wang, R., Cai, Y., Cao, J., Wang, F., Min, H., & Deng, F. (2016). Empirical analysis: Stock market prediction via extreme learning machine. *Neural Computing and Applications*, 27(1), 67–78.

Liu, J., & Park, S. (2015). Behind stock price movement: Supply and demand in market microstructure and market influence. *Journal of Trading*, 10(3), 13–23.

Maglaras, C., Moallemi, C. C., & Zheng, H. (2015). Optimal execution in a limit order book and an associated microstructure market impact model. Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstractid=2610808.

Majhi, R., Panda, G., & Sahoo, G. (2009). Development and performance evaluation of FLANN based model for forecasting of stock markets. *Expert Systems with Applications*, 36(3), 6800–6808.

Malik, A., & Lon Ng, W. (2014). Intraday liquidity patterns in limit order books. *Studies in Economics and Finance*, 31(1), 46–71.

Mankad, S., Michailidis, G., & Kirilenko, A. (2013). Discovering the ecosystem of an electronic financial market with a dynamic machine-learning method. *Algorithmic Finance*, 2(2), 151–165.

Næs, R., & Skjeltorp, J. A. (2006). Order book characteristics and the volume–volatility relation: Empirical evidence from a limit order market. *Journal of Financial Markets*, 9(4), 408–432.

O'Hara, M., & Ye, M. (2011). Is market fragmentation harming market quality? *Journal of Financial Economics*, 100(3), 459–474.

Pai, P.-F., & Lin, C.-S (2005). A hybrid Arima and support vector machines model in stock price forecasting. *Omega*, 33(6), 497–505.

Palguna, D., & Pollak, I. (2016). Mid-price prediction in a limit order book. *IEEE Journal of Selected Topics in Signal Processing*, 10(6), 1083–1092.

Panayi, E., Peters, G. W., Danielsson, J., & Zigrand, J.-P. (2016). Designating market maker behaviour in limit order book markets. *Econometrics and Statistics*, 5, 20–44.

Ranaldo, A. (2004). Order aggressiveness in limit order book markets. *Journal of Financial Markets*, 7(1), 53–74.

Rehman, M., Khan, G. M., & Mahmud, S. A. (2014). Foreign currency exchange rates prediction using CGP and recurrent neural network. *IERI Procedia*, 10, 239–244.

Sandoval, J., & Hernández, G. (2015). Computational visual analysis of the order book dynamics for creating high-frequency foreign exchange trading strategies. *Procedia Computer Science*, 51, 1593–1602.

Seddon, J. J., & Currie, W. L. (2017). A model for unpacking big data analytics in high-frequency trading. *Journal of Business Research*, 70, 300–307.

Sharang, A., & Rao, C. (2015). Using machine learning for medium frequency derivative portfolio trading. arXiv preprint retrieved from https://arxiv.org/abs/1512.06228

Siikanen, M., Kanniainen, J., & Luoma, A. (2017). What drives the sensitivity of limit order books to company announcement arrivals? *Economics Letters*, 159, 65–68.

Siikanen, M., Kanniainen, J., & Valli, J. (2017). Limit order books and liquidity around scheduled and non-scheduled announcements: Empirical evidence from NASDAQ Nordic. *Finance Research Letters*, 21, 264–271.

Sirignano, J. (2016). Deep learning for limit order books. Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2710331.

Suwanpetai, P. (2016). Estimation of exchange rate models after news announcement. In *AP16Thai Conference 2016: Sixth Asia–Pacific Conference on Global Business, Economics, Finance and Social Sciences*.

Talebi, H., Hoang, W., & Gavrilova, M. L. (2014). Multi-scale foreign exchange rates ensemble for classification of trends in FOREX market. *Procedia Computer Science*, 29, 2065–2075.

Vella, V., & Ng, W. L. (2016). Improving risk-adjusted performance in high frequency trading using interval type-2 fuzzy logic. *Expert Systems with Applications*, 55, 70–86.

Yang, S., Paddrik, M., Hayes, R., Todd, A., Kirilenko, A., Beling, P., & Scherer, W. (2012). Behavior Based Learning in Identifying High Frequency Trading Strategies. In *2012 IEEE Conference on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, IEEE, Piscataway, NJ, pp. 1–8.

Yang, S. Y., Qiao, Q., Beling, P. A., Scherer, W. T., & Kirilenko, A. A. (2015). Gaussian process-based algorithmic trading strategy identification. *Quantitative Finance*, 15(10), 1683–1703.

Yu, Y. (2006). The Limit Order Book Information and the Order Submission Strategy: a Model Explanation. In *2006 International Conference on Service Systems and Service Management*, IEEE, Piscataway, NJ, Vol. 1, pp. 687–691.

Zhang, K., Kwok, J. T., & Parvin, B. (2009). Prototype Vector Machine for Large Scale Semi-Supervised Learning. In *Proceedings of the 26Th Annual International Conference on Machine Learning*, ACM, New York, NY, pp. 1233–1240.

Zheng, B., Moulines, E., & Abergel, F. (2012). Price jump prediction in limit order book. Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2026454.

Adamantios Ntakaris is an ESR within the Marie Curie BigDataFinance project in the Dept. of Signal Processing at Tampere University of Technology. He received a B.Sc. in Mathematics in 2009 from the Aristotle University of Thessaloniki and an M.Sc. in Financial Modelling and Optimization in 2014 from the University of Edinburgh. In 2014 Adamantios completed an industrial placement at Standard Life Investments in Edinburgh. Before commencing his PhD, he worked as an Effective Interest Rate Analyst at CitiGroup investment bank in Edinburgh, and as a Maths Olympiad Coach in Thessaloniki.

Martin Magris is an Early Stage Researcher within the Marie Curie BigDataFinance training network in the Laboratory of Industrial and Information Management at Tampere University of Technology (Finland) since April 2016. He received a B.Sc. in Statistics and Mathematics in 2013 and a M.Sc in Statistical and Actuarial Sciences in 2015 from Universitá degli studi di Trieste, Italy. As a part of his master studies, Martin visited Aarhus university for seven months in 2014. In the years 2015-2016, before commencing his PhD, Martin worked as actuarial analyst for a non-life insurance company, specifically in the car-insurance pricing and in the development, profit-testing and pricing of multiple-peril non-life insurance products.

Juho Kanniainen is a Professor of Financial Engineering at the Tampere University of Technology, Finland. His research agenda is focused on quantitative finance with emphasis on big data problems. Dr. Kanniainen has published in many journals in Finance and Engineering, including Review of Finance, Journal of Banking and Finance, and Digital Signal Processing. He has been coordinating two international EU projects, BigDataFinance (www.bigdatafinance.eu) and HPCFinance (www.hpcfinance.eu).

Moncef Gabbouj is a Professor of Signal Processing at the Department of Signal Processing, Tampere University of Technology, Tampere, Finland. He was Academy of Finland Professor during 2011-2015. He held several visiting professorships at different universities. Dr. Gabbouj is currently the TUT-Site Director of the NSF IUCRC funded Center for Visual and Decision Informatics. His research interests include Big Data analytics, multimedia content-based analysis, indexing and retrieval, artificial intelligence, machine learning, pattern recognition, nonlinear signal and image processing and analysis, voice conversion, and video processing and coding.

Alexandros Iosifidis is currently an Assistant Professor of Machine Learning and Computer Vision in the Department of Engineering, at Aarhus University, Denmark. He has held Postdoctoral Researcher positions in Tampere University of Technology, Finland and Aristotle University of Thessaloniki, Greece. He has participated in many R&D projects financed by EU, Greek, Finnish, and Danish funding agencies and companies. He has co-authored more than 120 papers in international journals and conferences proposing novel Machine Learning techniques and their application in a variety of problems.