Applied Soft Computing Journal 158 (2024) 111555

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Modeling of limit order book data with ordered fuzzy numbers

Adam Marszałek [a,*], Tadeusz Burczyński [a,b]

[a] Faculty of Computer Science and Telecommunications, Cracow University of Technology, 31-155, Cracow, Poland
[b] Institute of Fundamental Technological Research, Polish Academy of Sciences, 02-106, Warsaw, Poland

ARTICLE INFO

Dataset link: https://dx.doi.org/10.17632/3g4mhdp899.1

Keywords:
Limit order book
Ordered fuzzy number
High-frequency forecasting
Mid-price
Data perturbation

ABSTRACT

This paper presents a novel approach to representing the Limit Order Book data at a given timestamp using the Ordered Fuzzy Numbers concept. The limit order book contains all buy and sell orders placed by investors, updated in real-time, for the most liquid securities, even several hundred times a minute. Due to its irregular nature (different and dynamic changes in the number of buy and sell orders), direct calculations on the order book data are not feasible without transforming it into feature vectors. Currently, most studies use a price level-based data representation scheme when applying deep learning models on limit order book data. However, this scheme has limitations, particularly its sensitivity to subtle perturbations that can negatively impact model performance. On the other hand, the ordered fuzzy number is a mathematical object (a pair of two functions) used to process imprecise and uncertain data. Ordered Fuzzy Numbers possess well-defined arithmetic properties. Converting the limit order book data to ordered fuzzy numbers allows the creation of a time series of ordered fuzzy numbers (order books) and use them for further calculations, e.g., to represent input data for deep learning models or employing the concept of fuzzy time series in various domains, such as defining liquidity measures based on limit order book data. In this paper, the proposed approach is tested using one-year market data from the Polish Stock Exchange for the five biggest companies. The DeepLOB model is employed to predict mid-price movement using different input data representations. The proposed representation of Limit Order Book data demonstrated remarkably stable out-of-sample prediction accuracy, even when subjected to data perturbation.

1. Introduction

Financial market research constantly evolves by integrating innovative computational tools and novel mathematical concepts. The scope and form of data used in these studies are also evolving, from simple closing prices to candlestick chart data to complete order information from the limit order book. Today, more than half of the stock markets are order-driven markets that use electronic limit order books [1,2]. Therefore, order limit book data is increasingly used as input to a wide range of computational models.

The Limit Order Book (LOB) is the central element of order-driven markets that records the intentions of buyers and sellers. It serves as a real-time representation of supply and demand for a particular financial instrument, displaying the changing price dynamics at a detailed, micro-structural level. LOBs, with their layers of price points and corresponding volumes, are inherently multi-dimensional, revealing complex spatial and time frames that explain price fluctuations. The structure of a LOB is complex, with multi-dimensional aspects resulting from multiple price points and order volumes spread over different levels for buy and sell orders. Fig. 1 provides a snapshot of this dynamic system, illustrating the intricate interplay of buy and sell orders at different price levels.

To better understand the meaning of LOBs, it is necessary to highlight the wealth of data they generate. With the advent of algorithmic trading and the digitization of exchanges, a deluge of LOB data has spurred a renewed focus on data-driven approaches in financial markets. As a result, the intersection of machine learning, in particular deep learning and quantitative finance, has flourished [3,4]. However, representing LOB data in machine learning models comes with its own set of challenges. While plenty of raw LOB data is available, translating this data into actionable insights requires sophisticated feature engineering or representation learning techniques. Interestingly, these advanced models are based on the initial representation of the LOB data. Currently, most studies use a price level-based data representation scheme. However, this representation has its limitations. Namely, it is susceptible to data perturbations. For example, traders are placing and canceling orders at deeper LOB levels, significantly amplifying the data's noise.

* Corresponding author.
E-mail address: adam.marszalek@pk.edu.pl (A. Marszałek).

https://doi.org/10.1016/j.asoc.2024.111555
Received 24 October 2023; Received in revised form 19 March 2024; Accepted 24 March 2024
Available online 2 April 2024

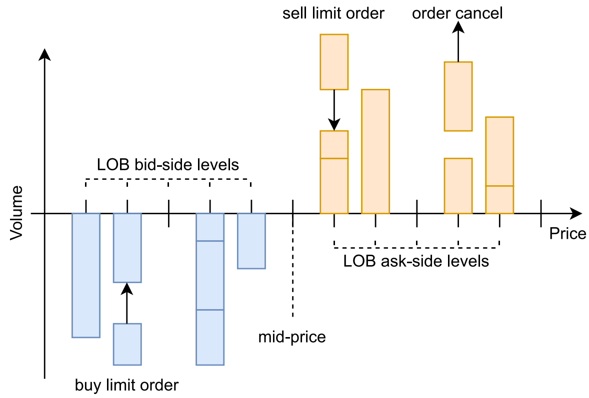

Fig. 1. A snapshot of the limit order book.

This paper introduces a novel approach to modeling LOB data for machine learning applications, challenging the conventional price-level-based representation. It proposes using Ordered Fuzzy Numbers (OFNs) to encapsulate the dynamic behavior of LOBs, enhancing robustness against data perturbations such as adding or removing small, insignificant orders. This innovative representation provides a more stable foundation for predictive models, potentially improving their performance forecasting market movements. This study makes several significant contributions to the field of financial market analysis. Firstly, by adopting ordered fuzzy numbers for LOB data representation, our approach provides a more nuanced understanding of market dynamics, enabling machine learning models to account for uncertainties and temporal variations in a way that traditional representations cannot. This innovation represents a substantial departure from existing methodologies, marking a pivotal shift towards a more resilient and adaptive modeling framework. Furthermore, there is an explicit outline of how our research broadens the current landscape of financial data analysis. Integrating ordered fuzzy numbers into LOB data modeling not only enhances the predictive capabilities of machine learning algorithms but also opens up new avenues for analyzing market behaviors previously obscured by the limitations of level-based data representation.

The rest of the paper is organized as follows. In Section 2, the background and related research are provided. Section 3 describes limit order data and the standard representation scheme. A brief review of the foundational concept of ordered fuzzy numbers is presented in Section 4, followed by the introduction of a new method for representing Limit Order Book data. A range of experiments designed to test the robustness of the new LOB representation under different data perturbations is presented in Section 5. A benchmark dataset and model are also outlined in this section. A summary of the findings and considerations for future work and extensions are discussed in Section 6.

2. Literature review and related work

The literature on machine learning models, particularly their application to financial market forecasting and high-frequency trading, is extensive and constantly evolving. Historically, the predictability of stock markets has been a significant area of interest in the economic literature. While there are differences of opinion on market efficiency, there is a consensus on market predictability to some extent, as evidenced by previous studies. For example, paper [5] highlights the short-term predictive power of dividend yields for excess returns and the role of discount and short rate movements in explaining dividend yield variations, paper [6] documents the predictability of excess returns across various financial markets, linking it to expectational errors in forecasting. The fact that the variance risk premium can predict stock market returns, with similar patterns observed across multiple countries, was revealed in [7]. The significant out-of-sample predictive power of stock market returns by separating them into three components was shown in [8]. Mandelbrot's works discuss the application of fractals and multifractals in financial markets, illustrating how these mathematical concepts can be used to analyze market behavior over different time scales [9,10].

High-frequency trading (HFT) and stock market prediction have been recognized as complex areas, prompting interdisciplinary research approaches. For example, paper [11] used a logistic regression model to predict price jumps between trades, while paper [12] developed a multivariate linear model for short-term stock price movements. Also, neural networks have been highlighted for application, given the chaotic nature of the data [13]. Other studies have expanded the application of machine learning, especially neural network architectures, incorporating innovative approaches to forecast currency rates and stock indexes and devise trading strategies. For instance, a spatial neural network model has been developed to model spatial distributions efficiently within limit order books, significantly outperforming traditional models by leveraging information deep within the order book for risk management [14]. Another study introduced a method to infer patterns of quote volatility and price momentum reflective of algorithmic trading behaviors, enhancing the detection and prediction of market movements [15]. The efficacy of machine learning in high-frequency market making was explored, emphasizing the importance of back-testing classifier performance under trade execution constraints [16]. Research on Recurrent Neuro-Evolution for forecasting foreign currency exchange rates achieved high accuracy, showcasing the potential of genetic programming in financial predictions [17]. The application of artificial neural networks was empirically tested on foreign exchange market data, highlighting the importance of input selection for time-series predictions [18]. A trigonometric functional link artificial neural network (FLANN) model was developed for short and long-term stock market predictions, demonstrating the utility of technical and macroeconomic indicators in forecasting [19]. Machine learning has also been applied to design medium-frequency trading strategies for US Treasury note futures, showing profitability [20]. Continuous time Bayesian networks were investigated for their causality expression capabilities, offering a new model for high-frequency financial data that outperforms older models [21]. The hierarchical Hidden Markov Model captured market sentiment dynamics in the USD/COP market, learning natural trading strategies and demonstrating superior performance [22]. These advancements underscore the growing sophistication and diverse application of neural networks in financial market analysis and strategy development.

Recent advancements have significantly boosted interest in deploying machine learning algorithms for predicting Limit Order Book (LOB) data. For example, these efforts encompass the establishment of the first publicly available high-frequency limit order market dataset for mid-price prediction, featuring data from five NASDAQ Nordic market stocks over ten days, offering a robust benchmark for comparing state-of-the-art methodologies [23]. Investigations into tensor-based learning algorithms for high-frequency trading have demonstrated superior performance in mid-price prediction over traditional vector-based approaches by effectively utilizing multilinear methods on a dataset containing over 4 million limit orders [24]. A deep learning approach such as multilayer perceptrons or recursive neural networks to uncover a universal and stationary price formation mechanism from billions of electronic market quotes and equities transactions provides stable out-of-sample prediction accuracy across various stocks and periods [25–27]. The temporal-aware neural Bag-of-Features model was designed for high-frequency LOB data, which employs radial basis function and accumulation layers to model both short-term and long-term dynamics, significantly outperforming traditional methods [28]. A neural network layer architecture incorporating bilinear projection and an attention mechanism was proposed for HFT, highlighting crucial temporal information, and achieving state-of-the-art results with fewer computations [29]. The application of deep learning using CNNs to predict stock price movements from large-scale, high-frequency time-series data derived from financial exchange order books showcasing superior performance to traditional models [30,31]. A novel method for creating stationary features to overcome the non-stationary nature of financial data, allowing effective application of deep learning models like LSTM networks and CNNs in predicting mid-price movements in the LOB, was presented in [32,33] and also self-attention transformer networks were explored in research [34,35]. Given the stochastic nature of financial time series, these algorithms often involve preprocessing or feature extraction. Techniques such as principal component analysis (PCA) and linear discriminant analysis (LDA) have been implemented in this regard [28]. These initiatives highlight the rich potential of machine learning and deep learning techniques in enhancing financial market analysis's prediction accuracy and efficiency, especially in the complex and fast-paced environment of high-frequency trading. Readers interested in a more extensive literature review are referred to the review publication [36].

Machine learning models' effectiveness largely hinges on how data is represented. In neural networks, both the learning of representations and the forecasting steps are intertwined within the framework of the network and are collectively trained to optimize the same objective function. Here, the initial representation of LOB serves as the foundational input for the neural networks, shaping the entirety of the model. Current research primarily uses a price-level data representation scheme when applying deep learning models to LOB data, especially in mentioned above studies [25–27,29,32–35]. In the work [37], attention was drawn to the fact that there needs to be more literature regarding the compatibility of this representation scheme with deep learning models. This work showed that the price-level data representation scheme is sensitive to perturbations in the data, which leads to a decrease in the performance of advanced machine learning models. Recognizing the critical insights this work offers, the issue of data perturbation's impact on deep learning models' performance was positioned at the heart of this paper.

Despite the considerable advances in financial market forecasting, particularly in modeling limit order book data, our extensive literature review has identified several gaps that necessitate further research:

- Current methodologies predominantly utilize price-level-based data representation schemes when applying machine learning models to LOB data. While these methods have proven effective to a degree, they inherently assume market homogeneity and neglect the dynamic nature of order books, leading to potential misinterpretations of market conditions. This study has found a lack of comprehensive models that can accurately encapsulate the complex, multi-dimensional aspects of LOB data, including an ever-changing number of price levels in the order book, different for the buy and sell side.
- The traditional data representation techniques are highly sensitive to subtle perturbations in the LOB data, such as minor fluctuations in order placements and cancellations. This sensitivity can adversely affect the performance of predictive models, rendering them less reliable under real-market conditions. A clear research gap exists in developing robust representation schemes that remain invariant or minimally affected by these frequent, minor data alterations.
- While mathematical concepts have been integrated into financial modeling, applying advanced mathematical structures, such as ordered fuzzy numbers, remains largely unexplored in LOB data representation. Fuzzy numbers offer a promising avenue for enhancing model robustness and accuracy by better handling the uncertainties and imprecisions inherent in financial data. The potential of these advanced mathematical concepts to improve LOB data modeling and prediction has not been fully investigated in existing literature.

By addressing these gaps, our study aims to advance financial market analysis and improve the predictive performance of models dealing with LOB data. Through the proposed novel approach of representing LOB data using ordered fuzzy numbers, we seek to tackle the abovementioned challenges by providing a more robust, dynamic, and generalizable framework for financial market forecasting.

3. Limit Order Book data

The Limit Order Book is a dynamic record in the order-driven financial market that catalogues all current buy (bid) and sell (ask) orders that have been placed but not executed or canceled. These orders are for specific assets and have a predetermined price and volume. LOB provides a comprehensive overview of an asset's current market demand and supply. In electronic marketplaces, the LOB updates dynamically in real-time as new orders arrive, existing orders are canceled or modified, or orders are executed [38].

The LOB is divided into two main sections. The bid side contains buy orders sorted in descending order based on their price, and the ask side contains sell orders sorted in ascending order based on their price. Each LOB side is divided into distinct levels based on submitted prices. Each level is characterized by price and volume of order.

Prices and volumes at different levels are often represented as vectors. The LOB vector representation helps capture the spatial structure and evolution of the market over time [23,37,39]. Spatially, price and volume at each level are interrelated. The spatial relationship at different levels is heterogeneous because there is no fixed gap between adjacent price levels. For example, using a level-based representation, a LOB snapshot can be represented as a vector

$$s_t = \{p_a^i(t), v_a^i(t), p_b^i(t), v_b^i(t)\}_{i=1}^{L}, \quad (1)$$

where $p_a^i(t)$, $p_b^i(t)$ are the ask and bid prices for price level i at time t and $v_a^i(t)$, $v_b^i(t)$ are the ask and bid volumes, respectively. The time sequences of these snapshots represent the evolution of the market, and this data structure can be represented as $S \in \mathbb{R}^{T\times 4L}$, where T is the history length and L is the number of (nonzero) price levels considered for each side.

This vector structure, while effective and intuitive from the point of view of human understanding and compatible with the matching engines in exchanges, has several disadvantages from the point of view of input data for machine learning models [37]. The spatial structure at different levels is not homogeneous because it is not assumed that adjacent price levels have constant spacing, which is the basic assumption for convolutional neural networks (CNNs) due to the parameter-sharing mechanism. In addition, this representation can sometimes be unstable due to dynamic changes in price levels, e.g., the previous best bid/ask data may suddenly move to second place when a new order with a better price is placed. Similarly, the appearance of a relatively small order can dramatically change the norm of this vector. Moreover, only the first L levels (usually 10) are considered in this representation, so these levels may take orders with a small volume while kicking out levels with a much larger volume.

4. Proposed representation of LOB

4.1. Why ordered fuzzy numbers?

Ordered fuzzy numbers (OFNs) are a specific type of fuzzy numbers introduced by Kosinski et al. (also called Kosiński's fuzzy numbers) in the series of papers [40–44]. OFNs, in contrast to classical fuzzy numbers [45,46], are defined by ordered pairs of continuous real functions defined on the interval [0, 1], i.e.,

$$A = (f, g) \text{ with } f, g : [0, 1] \to \mathbb{R} \ as \text{ continuous functions}, \quad (2)$$

instead of the classical membership function $\mu_A : \mathbb{R} \to [0, 1]$. The ordered fuzzy number and the ordered fuzzy number as a fuzzy number in classical meaning are presented in Fig. 2

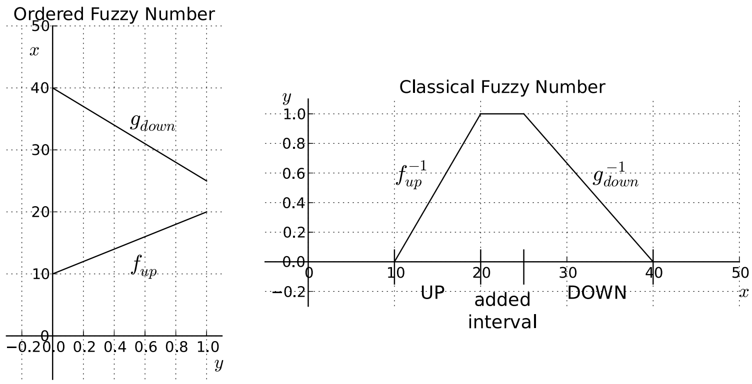

Fig. 2. Graphical interpretation of OFN and a OFN presented as fuzzy number in classical meaning [47].

The main advantage of OFNs is that the basic arithmetic operations $\{+, -, \cdot, \div\}$ are defined as the pairwise operations of their elements, which leads to the existence of the neutral elements of addition and multiplication. This fact means that the result of an arithmetic operation is not always a fuzzy number with larger support. Thanks to this, we can build fuzzy models based on ordered fuzzy numbers in the form of classical equations without losing accuracy. Essential mathematical functions such as log, exp, $sqrt$, and more can be defined in a similar way (see [48]). Moreover, an OFN is a generalization of the real number in the sense that each real number a can be represented as the OFN with constant functions f and g equal to a.

Ordered fuzzy numbers have found many applications in various areas, such as modeling dynamic changes, decision-making, DDoS attack analysis, and many others [49–53]. In particular, in our previous works [47,54–56], they have been used to model/represent high-frequency stock market data in the form of a well-defined mathematical object, based on which it becomes possible to build more robust and flexible models that better reflect the complexity and uncertainty characteristic of financial markets.

Fig. 3 shows a graphical representation of the LOB snapshot (real data, KGHM on 2017-01-03 09:22:52.299826, only price levels in range ±100% of mid-price) and an illustration of the corresponding demand and supply (the number of shares we can sell or buy at a fixed price limit) for the same limit order book data, respectively. It can be noted that the demand and supply lines can be identified with functions f and g. However, to be an OFN, these functions must be defined on the interval $[0, 1]$. For this purpose, a transformation method was developed, which is presented in the section below.

4.2. Limit order book as ordered fuzzy number

One of the interesting areas of the limit order book analysis is modeling and measuring liquidity. The ability to find and estimate intraday liquidity quickly and accurately is extremely valuable but also very challenging. A perfectly liquid market is one in which any amount of a given security can be instantaneously converted into cash and back to securities at no cost. In the real world, a liquid market is one where the transaction costs associated with this conversion are minimized [57]. Market liquidity is not unidimensional and can be understood in the following aspects [58]:

- the quantity of securities that are traded (depth)
- the ability of the security prices to quickly recover after a liquidity shock (resiliency)
- the costs incurred in trading security (tightness)
- the time taken to execute a trade (immediacy)
- the intensity of trading volume impact on security prices (breadth)

In this paper, the conversion of LOB to OFN is proposed in such a way that OFN, in a simple way, can illustrate the depth of a given instrument, i.e., it shows how many shares can be sold or bought at a given moment and how it will affect the share price.

Let $\{p_a^i(t), v_a^i(t)\}_{i=1}^{L_a}$ and $\{p_b^i(t), v_b^i(t)\}_{i=1}^{L_b}$ be a complete snapshot of LOB at time t (all price levels), where $p_a^i(t)$, $p_b^i(t)$ are the ask and bid prices for price level i at time t and $v_a^i(t)$, $v_b^i(t)$ are the ask and bid volumes, respectively, and L_a, L_b are the number of (nonzero) price levels considered for ask and bid side, respectively. Moreover, let $p_r(t)$ be a reference price at time t (e.g. mid-price or open price). Then the function f and g of OFN $A_t = (f_t, g_t)$ at time t are defined as follows:

$$f_t(x) = -\begin{cases} 0 & \text{if} \quad (1-x)p_r(t) > \mu_a^1(t), \\ \sum_{i=1}^{L_a} v_a^i(t) & \text{if} \quad (1-x)p_r(t) \le \mu_a^{L_a}(t), \\ \left(\sum_{i=1}^{l_a} v_a^i(t)\right)\left(1 + \dfrac{\mu_a^{l_a}(t) - (1-x)p_r(t)}{(1-x)p_r(t) - p_a^{l_a+1}(t)}\right) & \text{otherwise} \end{cases} \tag{3}$$

$$g_t(x) = \begin{cases} 0 & \text{if} \quad (1+x)p_r(t) < \mu_b^1(t), \\ \sum_{i=1}^{L_b} v_b^i(t) & \text{if} \quad (1+x)p_r(t) \ge \mu_b^{L_b}(t), \\ \left(\sum_{i=1}^{l_b} v_b^i(t)\right)\left(1 + \dfrac{\mu_b^{l_b}(t) - (1+x)p_r(t)}{(1+x)p_r(t) - p_b^{l_b+1}(t)}\right) & \text{otherwise} \end{cases} \tag{4}$$

where $\mu_a^i(t)$ and $\mu_b^i(t)$ are volume-weighted average prices at time t from level 1 to i for ask and bid prices, respectively, l_a is the lowest i that satisfies the relation $\mu_a^i < (1-x)p_r(t)$ and l_b is the lowest i that satisfies the relation $\mu_b^i > (1+x)p_r(t)$.

Interpretation

Fig. 4 shows an example of an ordered fuzzy number generated by the limit order book of KGHM on 2017-01-03 09:22:52.299826. The values of $|f(x)|$ and $g(x)$ show the size of trades that can be made for a given level of potential cost $(1-x)$, i.e., the percentage difference between the real average transaction price and the reference price for the ask and bid side, respectively. For example, if an investor wishes to purchase 30,000 shares of KGHM all at once, the average cost per share would be 94.97. Compared to the current mid-price of 94.52, this would result in an approximate loss of 0.4% if the investor could have bought the shares at the mid-price. Likewise, if the investor wants to sell 10,000 shares right now, the average selling price per share would be 94.22. This would lead to an approximate loss of 0.3% compared to the current mid-price.

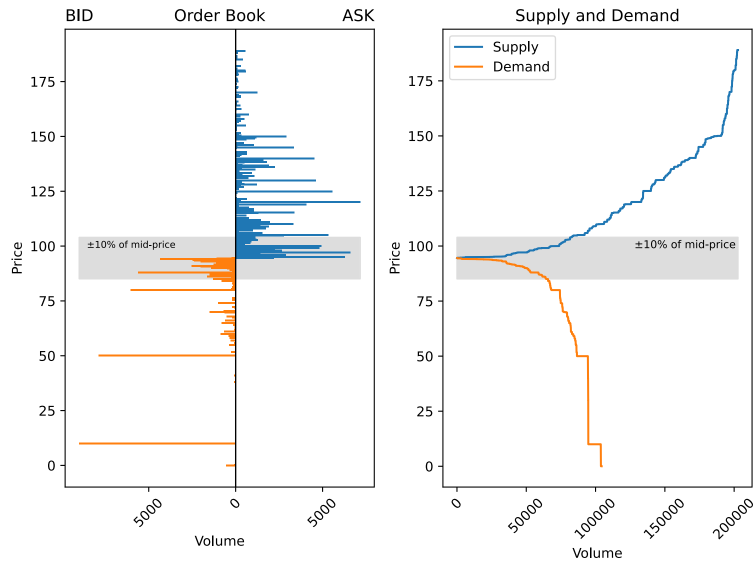

Fig. 3. Limit order book snapshot for KGHM on 2017-01-03 09:22:52.299826 and the corresponding demand and supply chart.

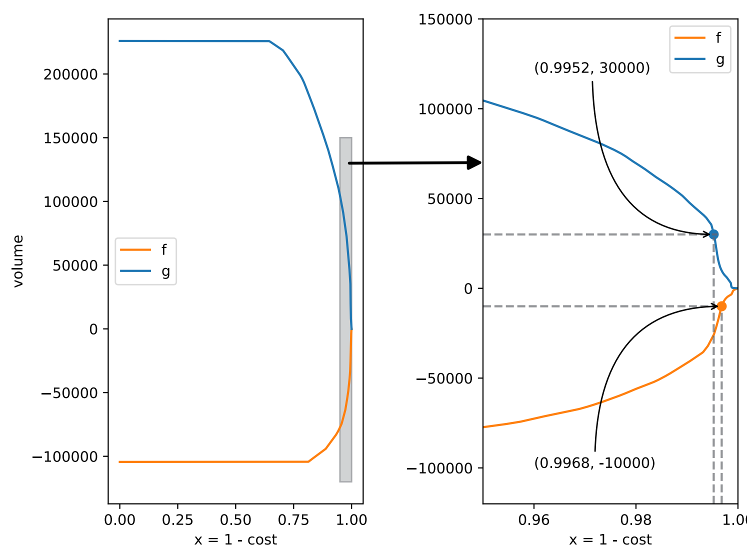

Fig. 4. The ordered fuzzy number generated by limit order book of KGHM on 2017-01-03 09:22:52.299826.

Additional parameters and modifications for practical use cases

The general definition of an ordered fuzzy number generated by limit order book data has been defined above. In order to better adapt it to specific practical applications, a few additional parameters and modifications described below are proposed for consideration.

Numbers of price levels. Considering all orders in the limit order book can be problematic and often inappropriate, there may be orders that cannot be executed in any given session due to price fluctuation constraints. Therefore, it is suggested to consider only price levels within specific reference price bounds (± daily price fluctuation limit, e.g., ±10%). The result of applying this restriction is shown in Fig. 5.

Maximum cost. Fig. 5 also illustrates that the bulk of the pertinent data is concentrated close to x=1. This concentration stems from the limited scope of examined price levels and the method of computing cost as a percentage difference between the average transaction price and the reference price. Therefore, it is recommended to implement a parameter that sets the upper limit for the considered cost (e.g., single transaction price fluctuation limit, e.g., 3.5%). Subsequently, adjust the x-axis linearly so that x=0 aligns with the maximum predetermined cost and x=1 aligns with the zero cost. The result of this transformation for the assumed maximum cost of 3.5% is shown in Fig. 6.

Discretization. In prior research, for numerical calculations, an ordered fuzzy number $A = (f, g)$ was represented as an ordered pair of vectors $(\mathbf{f}, \mathbf{g})$. These vectors $\mathbf{f}$ and $\mathbf{g}$ are constructed from the function values of f and g, respectively. These values are calculated at $(M+1)$ points, which are obtained by uniformly discretizing the interval $[0, 1]$, i.e.

$$\begin{aligned}\mathbf{f} &= [f(0), f(dx), f(2\cdot dx), \dots, f((M-1)\cdot dx), f(1)],\\ \mathbf{g} &= [g(0), g(dx), g(2\cdot dx), \dots, g((M-1)\cdot dx), g(1)],\end{aligned} \qquad dx = \frac{1}{M}. \quad (5)$$

This numerical representation of OFN allows for convenient implementation of arithmetic operations, resulting in ordered fuzzy numbers with functions f and g of any shape. In the case of LOB data representation,

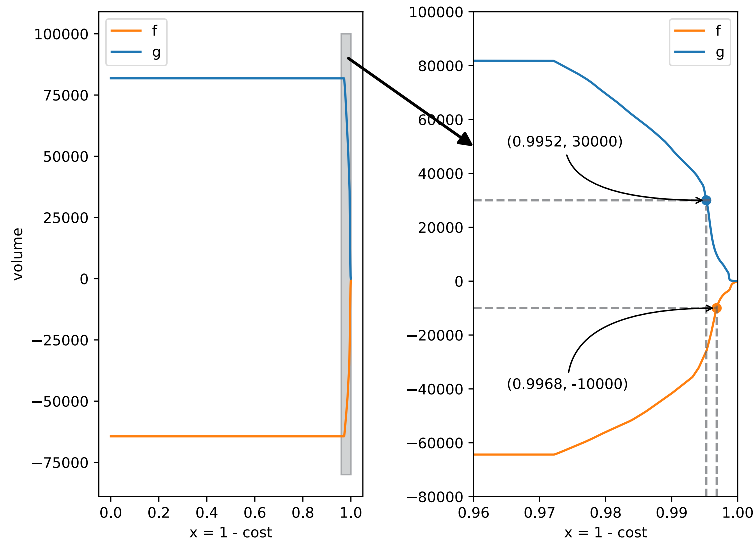

Fig. 5. The ordered fuzzy number generated by limit order book of KGHM on 2017-01-03 09:22:52.299826 under restriction to price levels in range of ±10% of mid-price.

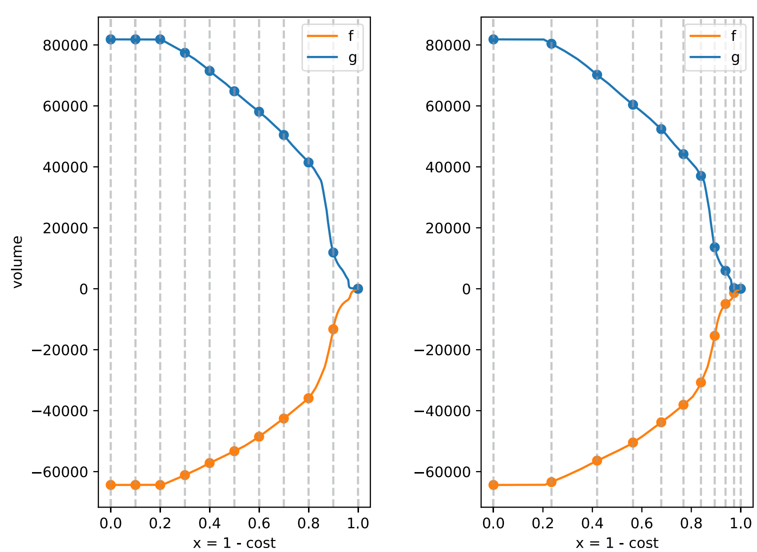

Fig. 6. The ordered fuzzy number generated by the limit order book of KGHM on 2017-01-03 09:22:52.299826 under restriction to price levels in the range of ±10% of mid-price and maximum cost of 3.5%. The labeled markers indicate the distribution of points when using linear and geometric progressions for discretization, respectively.

replacing the linear scale with a logarithmic scale (a geometric progression) to increase the density of points close to x=1 was proposed (see Fig. 6).

5. Experiments and results

5.1. Benchmark dataset and model

A commonly used LOB dataset for model testing is the FI-2010 dataset [23], which is the first publicly available benchmark dataset of high-frequency limit order book data. This dataset contains time series data for five Nasdaq Nordic stock market stocks for ten consecutive days. However, this publicly available data is based on only ten price levels and has already been pre-processed (normalized). Therefore, it is impossible to prepare input data for deep learning models based on them using the LOB data representation method proposed in this paper. Moreover, as mentioned in [33], a 10-day dataset is inadequate for thoroughly evaluating an algorithm's robustness and capacity for generalization, given that the issue of overfitting to historical data is significant and a signal is typically expected to remain stable over several months. That is why they also trained and tested their model on their dataset of limit order book data of one-year length for instruments that are among the most liquid stocks listed on the London Stock Exchange. Unfortunately, this dataset is not publicly available.

Acknowledgment is extended for the financial support received from The National Science Centre, which was crucial for acquiring significant historical data from the Warsaw Stock Exchange (WSE), including comprehensive information on all orders and transactions. Following consultations and approval from the WSE, an open-access version of this dataset has been made available to the research community. An introductory dataset description is provided below, and the dataset has been made publicly available in the Mendeley Data Repository[1],

[1] Marszałek, Adam (2023), "WSELOB-2017: The year-long database of limit order books for the five biggest companies listed on the Warsaw Stock Exchange", Mendeley Data, V1, doi: 10.17632/3g4mhdp899.1

Table 1
Statistical description of the dataset segmented by individual stocks.

Stock	LOB snapshots	LOB snapshots per day	Mean gap
KGHM	19, 500, 582	78, 002.328	0.361515 s
PKNORLEN	18, 261, 547	73, 046.188	0.386032 s
PKOBP	16, 322, 175	65, 288.7	0.431902 s
PZU	15, 743, 871	62, 975.484	0.447772 s
PEKAO	9, 022, 376	36, 089.504	0.781342 s

ensuring easy access and facilitating further research by the academic community.

WSE LOB dataset. A year-long dataset was compiled for the five biggest stocks on the Warsaw Stock Exchange in 2017: KGHM, PKNORLEN, PKOBP, PZU, and PEKAO; these are among the most liquid stocks on the WSE. This dataset covers all LOB updates (messages) and spans every trading day from January 1, 2017, to December 24, 2017. From the raw message books, one can reconstruct complete limit order books. The dataset focuses on time series data of the LOB, including only price levels within a ±10% daily price fluctuation limit, based on the actual mid-price. Each level in the series includes both price and volume information. Data coverage is during standard trading hours between 9:00 AM and 4:50 PM (CET or CEST, as applicable), excluding auctions and moments without changes in selected levels. The WSE dataset spans a year and includes almost 79 million data points. Each stock experiences, on average, about 63,000 events per day, and these events occur at inconsistent time intervals. The time gap between any two successive events can range from a fraction of a second to several minutes. The average time gap in the dataset is 0.481713 s. Statistics descriptions for individual stocks are presented in Table 1. For experiments, the initial nine months of data are used for training (with the separation of the last 10% of the records as validation data) and the final three months for testing. Given the high-frequency nature of the data, the 3-month test period includes millions of data points, making it adequate for evaluating the model's effectiveness and accuracy.

DeepLOB model. The DeepLOB model, as referenced in [33], was chosen as the benchmark model to evaluate the proposed method on the whole WSE dataset. This model is notable for being the first hybrid deep neural network designed to predict stock price movements using high-frequency limit order book data. It combines convolutional layers with Long Short-Term Memory (LSTM) units to achieve superior predictive performance compared to other existing algorithms that also use LOB data for feature extraction as of the time it was published. The authors of the DeepLOB model validated its performance using two datasets. The first was the FI-2010 benchmark LOB dataset, and the second was a comprehensive, year-long dataset featuring 134 million data points from the London Stock Exchange. We chose to use only the DeepLOB model for our experiments due to computational constraints, which include the need to train the model multiple times for various assets and input data types. Another reason for our choice was the similarity between the London Stock Exchange dataset used in the original DeepLOB model and WSE dataset. The similarity between the WSE dataset and that of the LSE dataset primarily stems from three aspects: the duration of the data, the inclusion of the most liquid assets, and exchange system similarities. Both datasets cover one year, allowing for a comprehensive analysis of market dynamics. Focusing on the most liquid stocks ensures that the data represents assets with significant trading activity, facilitating meaningful comparisons between market behaviors. The datasets originate from comparable exchange systems, with both markets operating in a manner that supports high-frequency trading and the detailed analysis of limit order book data. The higher liquidity of the LSE compared to the WSE is recognized.

5.2. Inputs, normalization and labeling

For the assessment of the new LOB data representation method's efficacy, two separate sets of input data were prepared for each asset.

The first set employs the traditional level-based LOB representation, as outlined in Section 3. In this method, each LOB state includes ten levels on both the bid and ask sides, featuring both price and volume data. This results in 40 features per timestamp. The 50 most recent states of the LOB are utilized as input for modeling purposes. In particular, a single input is defined as

$$X^{(1)} = [x_1^{(1)}, x_2^{(1)}, \dots, x_t^{(1)}, \dots, x_{50}^{(1)}]^T \in \mathbb{R}^{50\times 40}, \tag{6}$$

where

$$x_t^{(1)} = [p_a^i(t), v_a^i(t), p_b^i(t), v_b^i(t)]_{i=1}^{10}, \tag{7}$$

and $p_a^i(t)$, $p_b^i(t)$ are the ask and bid prices for price level i at time t and $v_a^i(t)$, $v_b^i(t)$ are the ask and bid volumes, respectively.

The second set comprises LOB snapshots formulated using the OFN approach, detailed in Section 4. The ±10% fluctuation limit was chosen based on the Polish stock market's regulatory and operational realities. In the Warsaw Stock Exchange, these are the established thresholds for the maximum price change that can occur during a single trading session and ±3.5% for individual transactions. This regulatory framework is designed to prevent excessive volatility and maintain market stability. In this case, a single input is designated as

$$X^{(2)} = [x_1^{(2)}, x_2^{(2)}, \dots, x_t^{(2)}, \dots, x_{50}^{(2)}]^T \in \mathbb{R}^{50\times 40}, \tag{8}$$

where

$$x_t^{(2)} = [(1 - r_i r_{max})p_r(t), f_t(r_i)/10^6, (1 + r_i r_{max})p_r(t), g_t(r_i)/10^6]_{i=1}^{10}, \tag{9}$$

where $A_t = (f_t, g_t)$ is the ordered fuzzy numbers generated by LOB data at time t, p_r is the reference price calculated as mid-price at time t, $r_{max} = 0.035$ is the maximum considered cost, and r_i are 10 points from the interval $(0, 1]$ spaced evenly on a log scale (a geometric progression, i.e. 0.02709816, 0.06153943, 0.10531364, 0.16094986, 0.23166248, 0.32153691, 0.43576567, 0.58094831, 0.76547279, 1).

In the same manner as previous studies [23,33,37], the z-score standardization to normalize our data for both types of inputs was employed. The mean and standard deviation from the preceding five days to normalize each day's data were used, applying this separately for each financial instrument. A static normalization approach is unsuitable for a one-year dataset because financial time series often undergo changing trends. This dynamic method ensures that the normalized data generally falls within a reasonable range.

The aim is centered around forecasting the micro-movements of the mid-price, which is delineated as follows

$$p_t = \frac{p_a^{(1)}(t) + p_b^{(1)}(t)}{2}. \tag{10}$$

Directly comparing the mid-prices p_t and p_{t+k} can result in a noisy label set. To address this, a smoothing labeling method similar to one used in previous research [23,30,33] was employed. Let m_- denote the mean of the previous k mid-prices and m_+ denote the mean of the next k mid-prices

$$m_-(t) = \frac{1}{k}\sum_{i=0}^{k} p_{t-i}, \tag{11}$$

$$m_+(t) = \frac{1}{k}\sum_{i=0}^{k} p_{t+i}, \tag{12}$$

where p_t is the mid-price defined in Eq. (10) and k is the prediction horizon. Calculating the percentage change l_t in the mid-price for determining its direction is performed using a following equation

$$l_t = \frac{m_+(t) - m_-(t)}{m_-(t)}. \tag{13}$$

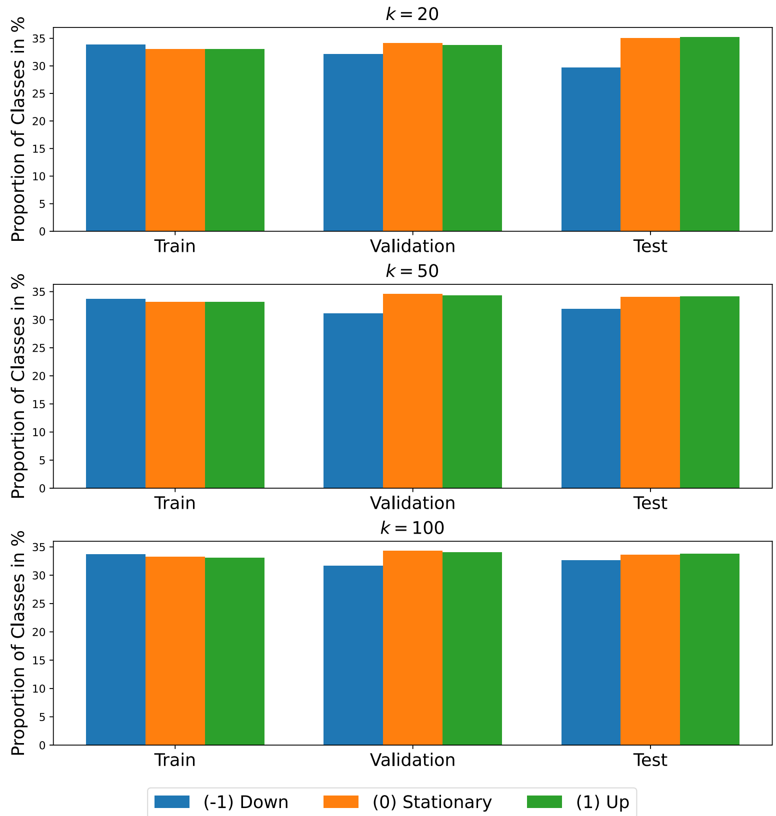

Fig. 7. Label class balancing for train, validation and test sets for different prediction horizons k for the WSE dataset.

Table 2
The values of threshold α_k for specific prediction horizons k for individual stocks.

Stock \ k	20	50	100
KGHM	0.0000263	0.000102	0.000155
PKNORLEN	0.000022	0.000085	0.000145
PKOBP	0.0000207	0.000089	0.000147
PZU	0.000029	0.0000845	0.000136
PEKAO	0.00005	0.00012	0.000185

Subsequently, the movement is classified into three classes: -1 for downward movement, 0 for stationary, and $+1$ for upward movement. The final labels are determined based on a threshold α_k for the percentage change l_t. If $l_t > \alpha_k$ or $l_t < -\alpha_k$, the movement is classified as upward $(+1)$ or downward (-1), respectively. For all other cases, it is considered stationary (0).

Table 2 lists the threshold values α_k for specific prediction horizons k. These thresholds were chosen to ensure a balanced distribution of classes in the training dataset. However, they also lead to a roughly even distribution of classes in validation and test sets, as demonstrated in Fig. 7.

5.3. Data perturbation

A straightforward data perturbation technique, inspired by the approach introduced in [37], was utilized to evaluate the stability of the proposed method for LOB data representation. In specific LOB data related to stocks, the price gap between adjacent levels sometimes exceeds the tick size (the smallest allowable price change). This phenomenon is notably common in small-tick stocks and can cause a complete shift in the LOB even with placing a minimum-size order at a price between existing levels. This perturbation method assumes that small orders are placed at vacant price levels beyond the best ask/bid prices, ensuring that the mid-price remains unchanged, thereby keeping prediction labels intact. This technique is illustrated through the use of a real LOB example. The first part (A) of Fig. 8 shows a LOB snapshot with ten price levels on each of the ask and bid sides (labeled L1-L10) before any perturbation. The mid-price in this snapshot is 93.93, with a bid–ask spread of 0.04. Assuming a tick size of 0.01 and a minimum order size of 1, some price levels, like 93.96 and 93.97 on the ask side and 93.87 and 93.86 on the bid side, are empty. To perturb this data, one can place minimum-size orders at these empty levels, which appears inconsequential as they neither affect the mid-price nor add significant volume.

Upon perturbation, nearly half of the original price level details become obscured (see Fig. 8 (B)). For instance, ask-side levels L6 to L10 vanish post-perturbation, while the remaining levels shift in their LOB positioning. This perturbation has two critical effects from a machine-learning standpoint. First, it dramatically alters the 40-dimensional input space; for example, the Euclidean distance between the 40-dimensional vectors pre and post-perturbation is 6129.09 (0.22 when the volume is normalized by a factor of 10^6), even though the total order volume applied is merely 8. Second, it limits the model's ability to "see" the market, as represented by the gray areas masked out in the LOB visualization after perturbation.

It is worth noting that in the proposed representation method, the number of price levels considered is variable, depending on the price range. Although a broader range of price levels is considered, perturbation is applied only between L1 and L10 to maintain the same level of disruption. As a result, this method is highly resilient to such perturbations because new orders will remove none of the existing levels. Moreover, the impact of small orders is minimal. For instance, the Euclidean distance between the 40-dimensional vectors generated by our method before and after perturbation is 0.02 (1464.44 when the

A Original LOB Data

	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
ask price	93.95	93.98	93.99	94.00	94.01	94.06	94.09	94.10	94.11	94.14
ask volume	600	100	874	4211	440	25	200	10	100	1000
bid price	93.91	93.90	93.89	93.88	93.85	93.84	93.83	93.80	93.76	93.74
bid volume	60	300	173	85	250	100	124	80	300	1000

B Perturbed LOB Data

	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10
ask price	93.95	93.96	93.97	93.98	93.99	94.00	94.01	94.02	94.03	94.04
ask volume	600	1	1	100	874	4211	440	1	1	1
bid price	93.91	93.90	93.89	93.88	93.87	93.86	93.85	93.84	93.83	93.82
bid volume	60	300	173	85	1	1	250	100	124	1

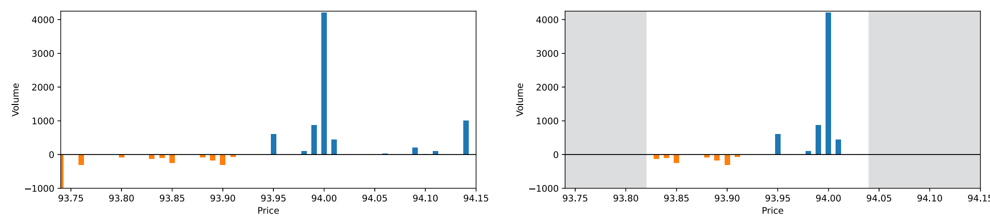

Fig. 8. (A) Original LOB data with 10 levels on ask and bid side without perturbation. (B) LOB data with 10 levels after data perturbation.

cumulative volume, i.e., values of functions f and g are not normalize by a factor of 10^6).

Note that in the data perturbation method outlined above, all available gaps are filled with new small orders. An extra parameter, P, is introduced to enhance the flexibility of the approach, indicating the percentage of empty price levels to be filled with orders. The empirical studies will investigate various levels of perturbation, specifically 0% (no perturbation), 1%, 5%, 10%, 25%, and 50%.

5.4. Experiments settings

The publicly accessible version of the DeepLOB model[2] is applied for all experiments, with the training settings, including the use of categorical cross-entropy loss and the ADAM optimization algorithm, maintained as per the original study's specifications. The learning is stopped if there is no improvement in validation accuracy over 20 consecutive epochs, with a maximum of 50 epochs. Training employs mini-batches of 64 samples, as suggested in [33,59]. All models are built using Keras [60] based on the TensorFlow backend [61] and are trained using a mirrored strategy of distributed learning on computing machines equipped with two NVIDIA Tesla V100 GPUs.

Models are trained separately for each of the five assets (KGHM, PKNORLEN, PKOBP, PZU, and PEKAO) in two variants of input data (standard and proposed) and for three forecast horizons ($k \in \{20, 50, 100\}$), which gave $5 \times 2 \times 3 = 30$ different configurations for training. Models are trained using a non-perturbed dataset, and data perturbation is applied at various levels ($P \in \{0\%, 1\%, 5\%, 10\%, 25\%, 50\%\}$) during the testing phase to evaluate model performance. In addition, all models were re-trained from scratch five times each for added statistical robustness of results.

5.5. DeepLOB model performance

Table 3 showcases the testing efficacy of the DeepLOB model in predicting price movements within the WSE dataset. The results are averaged across five stocks, with individual results for each stock detailed in Appendix A. The table explores variations in perturbation levels (P), prediction horizons (k), and input types (level-based $X^{(1)}$ and the newly proposed $X^{(2)}$). The model's performance is assessed using four distinct metrics: Accuracy (%), Precision (%), Recall (%), and F-score (%). Accuracy (%) is calculated as the percentage of test sample predictions that precisely align with the actual outcomes. This is considered an unbalanced accuracy score, while the remaining metrics are weighted averages across different classes. The confusion matrices are presented in Fig. 9 to facilitate a detailed examination of the results.

First, in a disturbance-free scenario ($P = 0\%$), the DeepLOB model shows performance comparable to that outlined in article [33] for the LSE dataset. Like the findings there, the model's effectiveness declines as the prediction horizon expands. While the impact of changing input types is minor, the proposed input method slightly outperforms all metrics across various forecasting horizons. This suggests that proposed input representation method captures critical information about the limit order book as effectively as traditional level-based representation schemes.

Second, in line with the observations made in the work [37], the model's performance deteriorates when unexpected data distortions occur, particularly with the traditional input representation. Compared to a perturbation-free environment, an observed decrease in accuracy ranges from approximately 7% ($P = 1\%$) to as much as 23% ($P = 50\%$). The confusion matrix data further indicates that, under the traditional input scheme, the model increasingly leans towards generating near-zero positive predictions as noise levels escalate. Contrastingly, this degradation is far less severe using our proposed input representation, ranging from less than 0.5% ($P = 1\%$) to around 4% ($P = 50\%$). This finding supports our hypothesis regarding the stability and resilience of the proposed input data representation to emerging noises in the LOB data.

To substantiate the significance of the findings further, the Wilcoxon signed-rank test was applied to the experimental data [62]. This statistical analysis was conducted to rigorously compare the performance metrics of the DeepLOB model with traditional and proposed input representation methods under varying noise levels. The Wilcoxon test, chosen for its suitability for non-parametric data, confirmed that the differences in performance between the two input methods were statistically significant. Specifically, the test results showed a significant advantage of proposed input method over the traditional one across all perturbation levels (see Table 3). This statistical confirmation underscores the robustness and effectiveness of the proposed input representation in maintaining model performance amidst data distortions, providing a solid statistical foundation for hypothesis on the resilience of the proposed method against noise in the LOB data.

Recognizing the potential value of transfer learning in broadening the applicability of predictive models in finance, Table 4 shows the experiment results of DeepLOB trained on KGHM and tested on the rest of the WSE dataset for various perturbation levels, prediction horizons, and types of inputs. The results are averaged across four stocks, with individual results for each stock detailed in Appendix B. The general

[2] https://github.com/zcakhaa/DeepLOB-Deep-Convolutional-Neural-Networks-for-Limit-Order-Books

Table 3
Experiment results on WSE dataset for various of perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	64.53	**65.45*****	67.1	**67.5**	64.53	**65.45*****	64.9	**65.76*****
0%	50	62.11	**62.56*****	61.57	**62.31*****	62.11	**62.56****	61.77	**62.42*****
	100	61.57	**62.06****	61.04	**61.49**	61.57	**62.06*****	61.24	**61.69***
	20	55.05	**65.22*****	54.79	**67.16*****	55.05	**65.22*****	54.57	**65.53*****
1%	50	55.27	**62.41*****	54.27	**62.21*****	55.27	**62.41*****	53.34	**62.3*****
	100	54.81	**62.01*****	53.64	**61.45*****	54.81	**62.01*****	53.38	**61.64*****
	20	47.84	**64.36*****	48.78	**65.9*****	47.84	**64.36*****	46.2	**64.65*****
5%	50	47.85	**61.78*****	48.59	**61.82*****	47.85	**61.78*****	44.44	**61.8*****
	100	48.07	**61.74*****	47.99	**61.23*****	48.07	**61.74*****	45.23	**61.42*****
	20	45.21	**63.5*****	46.82	**64.72*****	45.21	**63.5*****	42.65	**63.77*****
10%	50	45.4	**61.1*****	46.7	**61.42*****	45.4	**61.1*****	41.26	**61.24*****
	100	45.74	**61.42*****	46.03	**60.97*****	45.74	**61.42*****	42.4	**61.15*****
	20	42.49	**61.91*****	44.77	**62.69*****	42.49	**61.91*****	38.82	**62.12*****
25%	50	42.71	**59.44*****	44.43	**60.54*****	42.71	**59.44*****	37.9	**59.84*****
	100	42.83	**60.61*****	43.48	**60.4*****	42.83	**60.61*****	38.87	**60.49*****
	20	41.7	**61.14*****	43.99	**61.78*****	41.7	**61.14*****	37.99	**61.33*****
50%	50	41.58	**58.42*****	42.97	**60.03*****	41.58	**58.42*****	36.93	**58.94*****
	100	41.39	**60.02*****	42.02	**60.05*****	41.39	**60.02*****	37.49	**60.03*****

Note: ***, ** and * mean rejection of null hypotheses that the distribution of the differences $X^{(1)} - X^{(2)}$ is symmetric about zero at 1%, 5%, and 10% level.

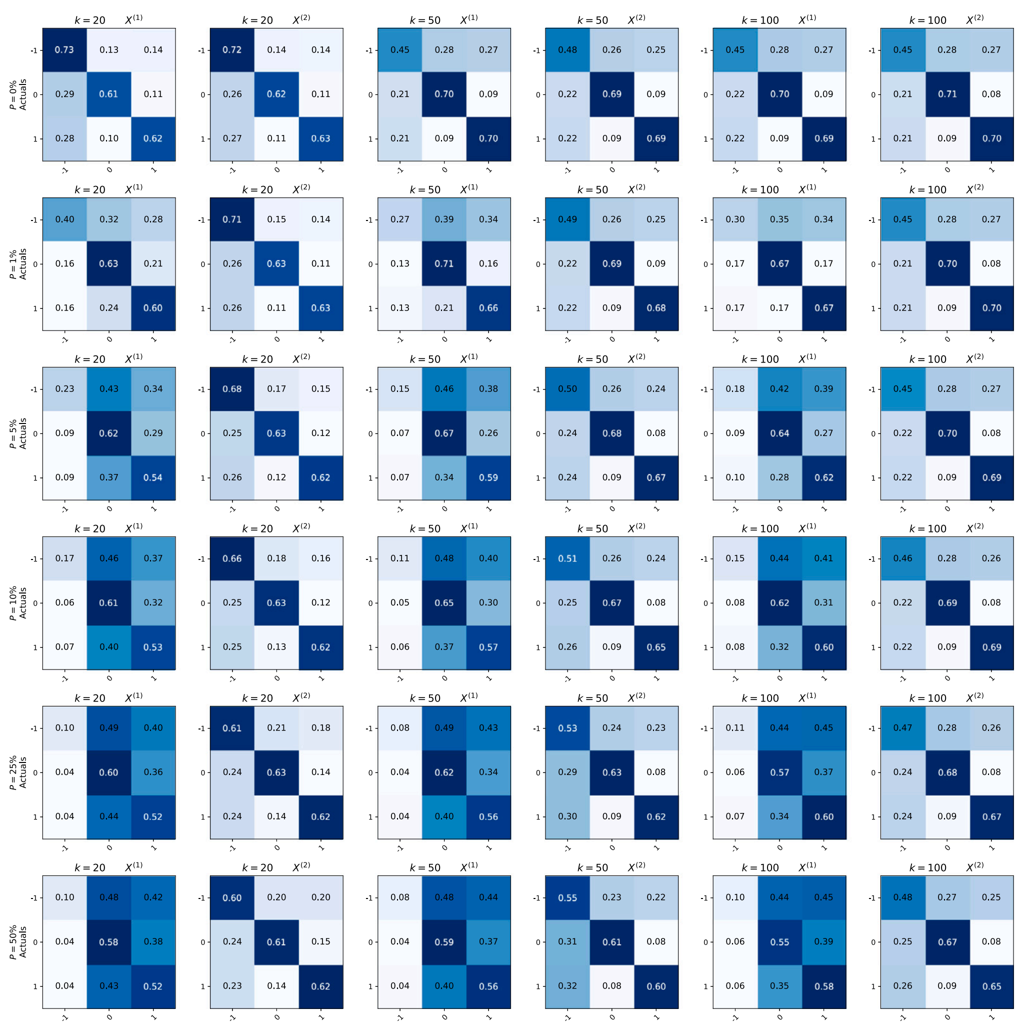

Fig. 9. Confusion matrices for corresponding experimental results in Table 3.

Table 4
Experiment results of DeepLOB trained on KGHM and tested on the rest of the WSE dataset for various of perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	63.38	**63.81*****	**65.12**	64.83	63.38	**63.81*****	63.65	**64.05*****
0%	50	61.07	**61.52****	60.6	**61.13****	61.07	**61.52****	60.69	**61.14****
	100	60.49	**60.88**	**60.59****	60.08	60.49	**60.88**	**60.41**	60.19
	20	54.66	**63.66*****	55.72	**64.62*****	54.66	**63.66*****	52.24	**63.88*****
1%	50	54.56	**61.4*****	53.35	**61.04*****	54.56	**61.4*****	51.94	**61.04*****
	100	53.71	**60.8*****	53.69	**60.0*****	53.71	**60.8*****	52.39	**60.11*****
	20	47.7	**63.04*****	49.21	**63.81*****	47.7	**63.04*****	43.7	**63.21*****
5%	50	47.85	**60.89*****	47.1	**60.67*****	47.56	**60.89*****	43.15	**60.59*****
	100	46.87	**60.53*****	47.11	**59.77*****	46.87	**60.53*****	42.61	**59.86*****
	20	44.9	**62.42*****	46.85	**63.04*****	44.9	**62.42*****	40.2	**62.51*****
10%	50	44.94	**60.36*****	45.49	**60.32*****	44.94	**60.36*****	40.09	**60.12*****
	100	44.26	**60.21*****	44.51	**59.5*****	44.26	**60.21*****	39.41	**59.56*****
	20	42.01	**61.19*****	44.84	**61.64*****	42.01	**61.19*****	36.57	**61.09*****
25%	50	41.76	**59.09*****	41.18	**59.62*****	41.76	**59.09*****	36.44	**59.0*****
	100	41.15	**59.47*****	42.05	**58.99*****	41.15	**59.47*****	35.94	**58.93*****
	20	41.28	**60.57*****	42.98	**60.94*****	41.28	**60.57*****	35.92	**60.38*****
50%	50	40.7	**58.4*****	41.36	**59.23*****	40.7	**58.4*****	35.18	**58.35*****
	100	40.02	**59.06*****	40.98	**58.76*****	40.02	**59.06*****	34.67	**58.59*****

Note: ***, ** and * mean rejection of null hypotheses that the distribution of the differences $X^{(1)} - X^{(2)}$ is symmetric about zero at 1%, 5%, and 10% level.

Table 5
Experiment results of five mainstream deep learning models trained on KGHM for various of perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	**65.42**	65.06	**68.98***	68.24	**65.42**	65.06	**65.75**	65.34
0%	50	**62.01**	61.75	**61.69**	61.46	**62.01**	61.75	**61.77****	61.55
	100	**60.95**	60.82	**60.52**	60.35	**60.95**	60.82	**60.63**	60.49
	20	56.69	**64.86*****	58.14	**67.88*****	56.69	**64.86*****	55.69	**65.14*****
1%	50	53.46	**61.62*****	54.75	**61.37*****	53.46	**61.62*****	52.69	**61.45*****
	100	51.6	**60.75*****	52.12	**60.29*****	51.6	**60.75*****	50.43	**60.42*****
	20	48.24	**64.07*****	50.32	**66.59*****	48.24	**64.07*****	45.12	**64.34*****
5%	50	44.86	**61.11*****	47.84	**61.01*****	44.86	**61.11*****	41.58	**61.03*****
	100	43.49	**60.38*****	44.31	**60.0*****	43.49	**60.38*****	39.22	**60.09*****
	20	45.94	**63.26*****	48.23	**65.4*****	45.94	**63.26*****	42.77	**63.5*****
10%	50	42.84	**60.57*****	46.33	**60.69*****	42.84	**60.57*****	39.47	**60.59*****
	100	41.34	**59.97*****	42.06	**59.71*****	41.34	**59.97*****	36.7	**59.75*****
	20	43.65	**61.64*****	45.57	**63.31*****	43.65	**61.64*****	40.39	**61.76*****
25%	50	40.67	**59.24*****	43.09	**60.01*****	40.67	**59.24*****	36.84	**59.45*****
	100	38.89	**58.8*****	39.3	**59.0*****	38.89	**58.8*****	34.04	**58.74*****
	20	42.67	**60.75*****	44.9	**62.23*****	42.67	**60.75*****	39.17	**60.78*****
50%	50	39.58	**58.48*****	43.4	**59.64*****	39.58	**58.48*****	35.2	**58.75*****
	100	37.65	**57.94*****	40.4	**58.5*****	37.65	**57.94*****	32.75	**57.96*****

Note: ***, ** and * mean rejection of null hypotheses that the distribution of the differences $X^{(1)} - X^{(2)}$ is symmetric about zero at 1%, 5%, and 10% level.

conclusions in this case are the same as in the case of the results presented in Table 3. This preliminary exploration aims to shed light on how models trained on data from one company might perform when applied to data from other companies, offering insights into the models' adaptability and potential for broader market analysis.

5.6. Other deep learning models performance

In order to examine whether the obtained conclusions about the resistance of the proposed method to disturbances in the data are independent of the choice of model, five other deep learning models were trained, including a convolutional network (CNN [30]), a LSTM [25], an Attention-augmented-Bilinear-Network with two hidden layers (C(TABL) [29]) and two extensions of DeepLOB (DeepLOB-Seq2Seq and DeepLOB-Attention [35]). Table 5 showcases the testing efficacy of these five mainstream deep learning models in predicting price movements of KGHM. The results are averaged across five models, with individual results for each model detailed in Appendix C. The table explores variations in perturbation levels (P), prediction horizons (k), and input types (level-based $X^{(1)}$ and the newly proposed $X^{(2)}$).

Upon examining both collective and individual outcomes, it is evident that the proposed method for representing input data demonstrates enhanced stability when faced with data disturbances across all models considered. Conversely, the traditional representation scheme yields slightly superior results with undisturbed data on average.

Notably, the latest variations of the DeepLOB model deliver performances closely aligned with the original model, particularly excelling in scenarios with longer forecasting durations. In particular, the DeepLOB-Attention model achieved the best performance with longer forecasting horizons. Consequently, the evaluation of the proposed approach was

Table 6
Experiment results of DeepLOB-Attention model on WSE dataset for various of perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	64.05	**65.07*****	66.79	**67.41****	64.05	**65.07*****	64.38	**65.34*****
0%	50	62.46	**63.07*****	62.12	**62.78****	62.46	**63.07*****	62.11	**62.74*****
	100	62.16	**63.13*****	61.92	**62.88*****	62.16	**63.13*****	61.79	**62.8*****
	20	55.66	**64.88*****	56.78	**67.1*****	55.66	**64.88*****	53.94	**65.14*****
1%	50	54.88	**62.96*****	55.05	**62.68*****	54.88	**62.96*****	51.88	**62.66*****
	100	54.34	**63.06*****	54.8	**62.81*****	54.34	**63.06*****	51.08	**62.74*****
	20	48.01	**64.15*****	50.68	**66.04*****	48.01	**64.15*****	43.43	**64.37*****
5%	50	47.37	**62.49*****	49.49	**62.31*****	47.37	**62.49*****	41.5	**62.31*****
	100	46.87	**62.72*****	49.16	**62.48*****	46.87	**62.72*****	40.93	**62.48*****
	20	45.52	**63.43*****	48.26	**65.09*****	45.52	**63.43*****	40.38	**63.59*****
10%	50	44.96	**62.01*****	49.28	**61.98*****	44.96	**62.01*****	38.35	**61.94*****
	100	44.4	**62.33*****	47.07	**62.16*****	44.4	**62.33*****	37.79	**62.17*****
	20	42.95	**62.11*****	44.78	**63.59*****	42.95	**62.11*****	37.24	**62.14*****
25%	50	42.27	**60.89*****	42.53	**61.3*****	42.27	**60.89*****	35.15	**61.01*****
	100	41.74	**61.3*****	42.77	**61.44*****	41.74	**61.3*****	34.67	**61.3*****
	20	42.19	**61.43*****	44.27	**62.97*****	42.19	**61.43*****	36.78	**61.38*****
50%	50	41.24	**60.21*****	43.05	**60.92*****	41.24	**60.21*****	34.48	**60.41*****
	100	40.58	**60.6*****	43.22	**61.0*****	40.58	**60.6*****	33.86	**60.69*****

Note: ***, ** and * mean rejection of null hypotheses that the distribution of the differences $X^{(1)} - X^{(2)}$ is symmetric about zero at 1%, 5%, and 10% level.

expanded by incorporating results obtained from this model for the additional four companies within the WSE dataset. This was conducted to assess the generalizability and robustness of the DeepLOB-Attention model in various scenarios. Results on the entire WSE dataset are displayed in Table 6, whereas details for individual stocks are provided in C. These results further affirm the advantage and stability of the new input data representation.

In contrast, other models generally underperform, irrespective of the forecasting horizon. The C(TABL) model, despite its lowest performance with undisturbed data, is distinguished by its most minor compared to the others performance degradation in response to increasing data disturbances for both considered types of input data, marking it as a promising avenue for future research.

6. Conclusion

This paper introduced a novel approach to representing Limit Order Book data, leveraging the Ordered Fuzzy Numbers concept. The study was motivated by the challenges associated with the traditional level-based representation of LOB data, particularly its sensitivity to data perturbations. The proposed representation scheme aims to effectively capture the dynamic behavior of LOBs while being considerably more robust to noise in the data. The proposed method was rigorously evaluated using year-long market data from the Warsaw Stock Exchange for the five largest companies.

The results indicate several key insights. In a disturbance-free environment, the proposed OFN-based representation demonstrated comparable performance to traditional level-based schemes, slightly outperforming them in all evaluated metrics across various forecasting horizons. This confirms the efficacy of OFN in capturing critical information present in LOBs. One of the most compelling findings is the robustness of the proposed OFN representation to data perturbations. Traditional level-based representation schemes showed a significant decrease in prediction accuracy with increasing noise levels, dropping by up to 23% under certain conditions. In stark contrast, the OFN representation maintained a remarkably stable performance, with accuracy deteriorating by no more than 4%.

It is acknowledged that while the proposed model for representing LOB data has shown promising stability and predictive accuracy, especially in the face of data perturbations, there are inherent limitations and assumptions within this study that warrant discussion. Primarily, the research was conducted within the context of the Warsaw Stock Exchange, which may not fully encapsulate the dynamics and liquidity of larger markets such as the London Stock Exchange. Moreover, the model's reliance on the specific structure of the LOB and the assumption of market conditions remaining consistent with the historical data used for training may not hold across different periods or unforeseen market conditions. Future research could address these limitations by testing the model across a broader range of markets and conditions, potentially enhancing its robustness and applicability.

In conclusion, the OFN-based representation serves as a promising input method for machine learning models like DeepLOB and opens doors to other applications, such as defining liquidity measures or employing fuzzy time series methods. The results highlight the utility and robustness of the OFN-based representation, making it a promising alternative for future research and practical applications. The future work would mainly focus on applying the proposed representation scheme to define new liquidity measures because capturing the aspects of liquidity, such as the depth of the market, was the main inspiration for developing the method. Moreover, future work may focus on exploring the applicability of this method to other types of financial instruments and investigating the benefits of integrating the proposed representation with other machine learning models.

CRediT authorship contribution statement

Adam Marszałek: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Tadeusz Burczyński:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Link to the Mendeley repository: Marszałek, Adam (2023), ''WSEL-OB-2017: The year-long database of limit order books for the five biggest companies listed on the Warsaw Stock Exchange", Mendeley Data, V1, doi: https://dx.doi.org/10.17632/3g4mhdp899.1.

Acknowledgments

We are most grateful to the Warsaw Stock Exchange (WSE), which approved releasing the open-access version of our data to share with the broader research community. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016455. This work was financially supported from the grant no. 2019/03/X/ST6/01370 funded by The National Science Centre (NCN), Poland.

Appendix A. DeepLOB model performance for individual stocks

This section showcases the remaining tables and confusion matrices that display the outcomes generated by the DeepLOB model for individual stocks (see Figs. A.10–A.14).

Table A.7
Experiment results on KGHM for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	65.97	**66.59**	68.52	**68.86**	65.97	**66.59**	66.3	**66.87**
0%	50	**62.92**	62.85	62.6	**62.77**	**62.92**	62.85	62.68	**62.76**
	100	**62.77**	61.96	**62.6**	61.45	**62.77**	61.96	**62.67**	61.64
	20	48.68	**66.1**	49.68	**68.12**	48.68	**66.1**	44.83	**66.37**
1%	50	51.04	**62.5**	48.99	**62.54**	51.04	**62.5**	47.8	**62.48**
	100	50.27	**61.77**	50.27	**61.28**	50.27	**61.77**	48.61	**61.46**
	20	39.47	**64.07**	42.63	**65.25**	39.47	**64.07**	31.82	**64.31**
5%	50	40.82	**61.06**	39.9	**61.64**	40.82	**61.06**	33.73	**61.25**
	100	40.66	**60.97**	40.44	**60.58**	40.66	**60.97**	32.81	**60.69**
	20	38.04	**61.93**	40.67	**62.53**	38.04	**61.93**	30.04	**62.06**
10%	50	39.11	**59.5**	38.81	**60.81**	39.11	**59.5**	31.96	**59.89**
	100	39.02	**60.01**	38.88	**59.83**	39.02	**60.01**	30.55	**59.79**
	20	36.95	**57.85**	39.29	**57.91**	36.95	**57.85**	29.02	**57.52**
25%	50	37.61	**55.75**	37.68	**59.28**	37.61	**55.75**	30.66	**56.39**
	100	37.38	**57.76**	37.4	**58.42**	37.38	**57.76**	29.53	**57.8**
	20	36.38	**55.91**	38.01	**55.7**	36.38	**55.91**	28.98	**55.24**
50%	50	36.55	**53.64**	34.73	**58.32**	36.55	**53.64**	29.57	**54.26**
	100	36.17	**56.32**	35.91	**57.74**	36.17	**56.32**	28.91	**56.55**

Table A.8
Experiment results on PKNORLEN for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	64.09	**65.2**	66.6	**66.61**	64.09	**65.2**	64.6	**65.54**
0%	50	61.98	**62.28**	60.7	**61.17**	61.98	**62.28**	60.82	**61.35**
	100	61.2	**61.86**	60.01	**60.75**	61.2	**61.86**	59.62	**60.77**
	20	50.48	**64.83**	50.03	**66.07**	50.48	**64.83**	48.19	**65.15**
1%	50	51.11	**62.02**	49.87	**61.02**	51.11	**62.02**	45.89	**61.24**
	100	51.85	**61.81**	50.2	**60.72**	51.85	**61.81**	46.31	**60.76**
	20	41.85	**63.56**	43.46	**64.37**	41.85	**63.56**	35.92	**63.8**
5%	50	41.94	**60.95**	42.23	**60.52**	41.94	**60.95**	34.5	**60.65**
	100	43.5	**61.51**	42.51	**60.5**	43.5	**61.51**	35.78	**60.64**
	20	40.46	**62.38**	42.87	**62.95**	40.46	**62.38**	34.19	**62.55**
10%	50	40.2	**59.75**	41.31	**60.14**	40.2	**59.75**	32.77	**59.86**
	100	41.45	**61.14**	41.1	**60.28**	41.45	**61.14**	33.95	**60.45**
	20	39.31	**60.28**	41.47	**60.67**	39.31	**60.28**	32.83	**60.35**
25%	50	38.49	**56.74**	37.92	**59.4**	38.49	**56.74**	31.69	**57.52**
	100	38.8	**60.0**	37.51	**59.72**	38.8	**60.0**	31.57	**59.76**
	20	38.65	**59.21**	39.34	**59.75**	38.65	**59.21**	32.42	**59.22**
50%	50	37.39	**54.71**	35.48	**58.96**	37.39	**54.71**	31.47	**55.75**
	100	36.92	**59.04**	35.76	**59.33**	36.92	**59.04**	30.15	**59.09**

Table A.9
Experiment results on PKOBP for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	64.75	**65.53**	68.18	**68.23**	64.75	**65.53**	64.96	**65.74**
0%	50	61.96	**62.42**	61.79	**62.98**	61.96	**62.42**	61.84	**62.61**
	100	61.33	**62.15**	61.12	**62.2**	61.33	**62.15**	61.15	**62.15**
	20	63.17	**65.53**	65.11	**68.23**	63.17	**65.53**	63.38	**65.74**
1%	50	60.51	**62.42**	59.97	**62.97**	60.51	**62.42**	59.97	**62.6**
	100	60.15	**62.15**	59.7	**62.2**	60.15	**62.15**	59.69	**62.15**
	20	57.56	**65.53**	57.44	**68.22**	57.56	**65.53**	57.35	**65.74**
5%	50	55.48	**62.42**	55.58	**62.97**	55.48	**62.42**	53.03	**62.6**
	100	56.07	**62.16**	55.7	**62.2**	56.07	**62.16**	54.43	**62.15**
	20	52.67	**65.53**	52.88	**68.23**	52.67	**65.53**	51.28	**65.74**
10%	50	51.43	**62.42**	53.0	**62.97**	51.43	**62.42**	47.06	**62.6**
	100	52.43	**62.16**	52.86	**62.2**	52.43	**62.16**	49.53	**62.15**
	20	47.27	**65.53**	48.67	**68.23**	47.27	**65.53**	43.78	**65.74**
25%	50	46.77	**62.42**	50.59	**62.97**	46.77	**62.42**	40.42	**62.6**
	100	47.73	**62.16**	49.9	**62.2**	47.73	**62.16**	42.97	**62.15**
	20	45.87	**65.52**	47.72	**68.23**	45.87	**65.52**	41.84	**65.73**
50%	50	45.34	**62.41**	49.85	**62.97**	45.34	**62.41**	38.63	**62.6**
	100	46.01	**62.16**	49.34	**62.2**	46.01	**62.16**	40.54	**62.15**

Table A.10
Experiment results on PZU for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	63.71	**65.11**	66.29	**68.03**	63.71	**65.11**	64.05	**65.4**
0%	50	61.66	**62.86**	61.51	**62.67**	61.66	**62.86**	61.51	**62.74**
	100	61.5	**63.09**	61.63	**62.65**	61.5	**63.09**	61.48	**62.79**
	20	62.3	**65.11**	64.04	**68.02**	62.3	**65.11**	62.58	**65.4**
1%	50	60.66	**62.86**	60.35	**62.66**	60.66	**62.86**	60.37	**62.74**
	100	60.53	**63.09**	60.66	**62.65**	60.53	**63.09**	60.5	**62.79**
	20	57.87	**65.11**	58.02	**68.02**	57.87	**65.11**	57.88	**65.4**
5%	50	57.35	**62.87**	56.71	**62.67**	57.35	**62.87**	56.47	**62.75**
	100	57.5	**63.09**	57.52	**62.65**	57.5	**63.09**	57.28	**62.79**
	20	54.25	**65.1**	54.18	**68.02**	54.25	**65.1**	53.85	**65.4**
10%	50	54.84	**62.87**	54.22	**62.67**	54.84	**62.87**	53.38	**62.75**
	100	55.14	**63.09**	55.04	**62.65**	55.14	**63.09**	54.63	**62.79**
	20	49.89	**65.11**	50.31	**68.02**	49.89	**65.11**	48.67	**65.4**
25%	50	51.41	**62.86**	51.1	**62.67**	51.41	**62.86**	49.12	**62.74**
	100	51.79	**63.09**	51.58	**62.66**	51.79	**63.09**	50.66	**62.8**
	20	48.99	**65.11**	49.53	**68.02**	48.99	**65.11**	47.68	**65.4**
50%	50	50.35	**62.87**	50.11	**62.67**	50.35	**62.87**	47.89	**62.75**
	100	50.52	**63.08**	50.34	**62.65**	50.52	**63.08**	49.1	**62.79**

Appendix B. DeepLOB model performance for transfer learning

This section presents tables that outline the results of applying the DeepLOB model in transfer learning scenarios across different stocks. Specifically, the model, initially trained on KGHM stock data, was subsequently tested on data from other stocks to evaluate its predictive performance. See Tables B.12–B.15.

Appendix C. Other deep learning models performance on KGHM

This section presents tables that outline the results of applying the other five mainstream deep learning models on KGHM and, in case of DeepLOB-Attention, on the rest of the WSE dataset. See Tables C.16–C.24.

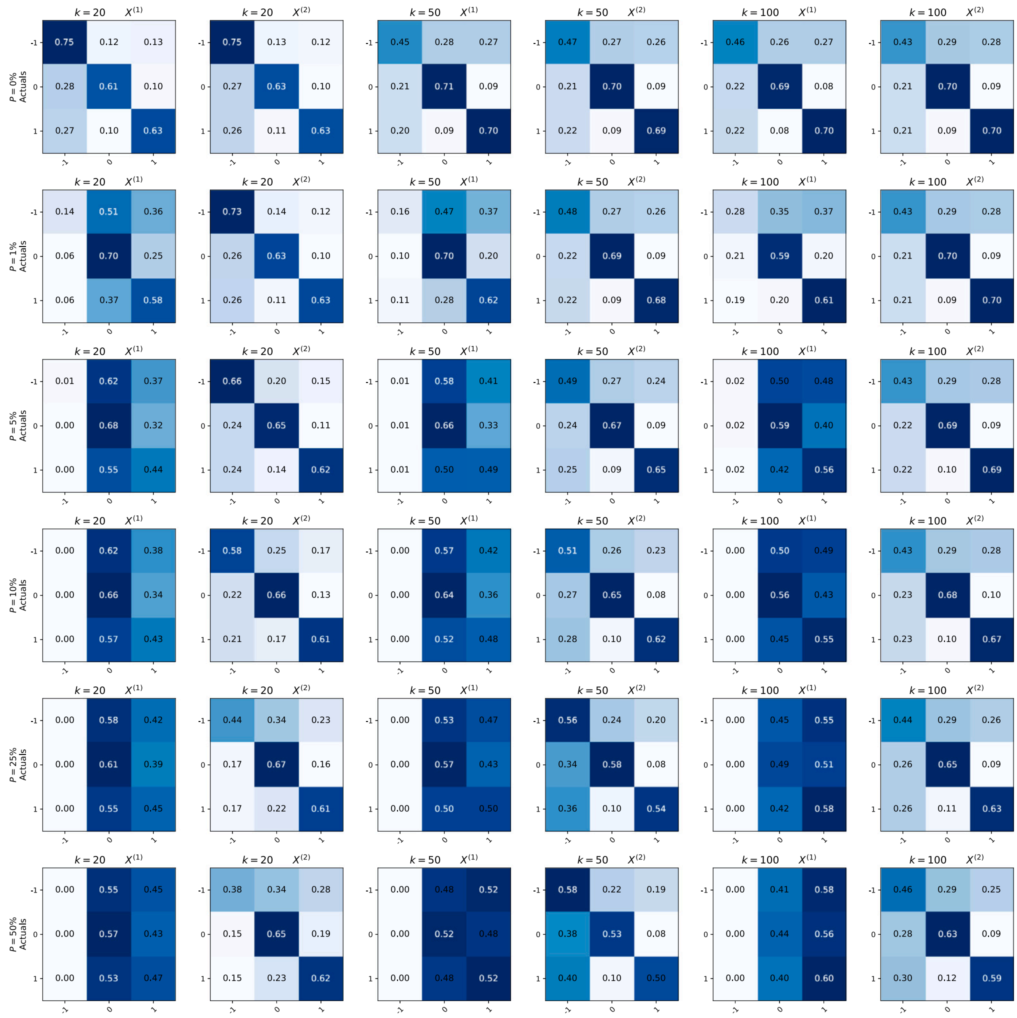

Fig. A.10. Confusion matrices for corresponding experimental results in Table A.7.

Table A.11
Experiment results on PEKAO for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	64.0	**64.35**	**65.62**	65.36	64.0	**64.35**	64.33	**64.59**
0%	50	62.02	**62.27**	61.6	**62.04**	62.02	**62.27**	61.73	**62.11**
	100	**60.64**	60.54	**60.8**	60.02	**60.64**	60.54	**60.65**	60.17
	20	47.99	**64.09**	49.78	**65.0**	47.99	**64.09**	42.83	**64.31**
1%	50	52.13	**62.16**	52.55	**61.93**	52.13	**62.16**	45.34	**62.0**
	100	48.83	**60.5**	49.1	**59.98**	48.83	**60.5**	43.49	**60.12**
	20	39.28	**63.05**	45.42	**63.67**	39.28	**63.05**	31.68	**63.22**
5%	50	41.4	**61.65**	45.17	**61.42**	41.4	**61.65**	32.68	**61.49**
	100	38.88	**60.32**	40.3	**59.79**	38.88	**60.32**	30.1	**59.93**
	20	37.72	**62.05**	44.22	**62.48**	37.72	**62.05**	29.99	**62.17**
10%	50	39.02	**61.2**	42.06	**61.0**	39.02	**61.2**	30.13	**61.06**
	100	37.12	**60.11**	38.6	**59.59**	37.12	**60.11**	27.93	**59.72**
	20	36.56	**60.23**	40.34	**60.57**	36.56	**60.23**	28.88	**60.31**
25%	50	37.0	**60.12**	37.16	**60.13**	37.0	**60.12**	28.39	**60.07**
	100	35.33	**59.61**	36.34	**59.15**	35.33	**59.61**	26.18	**59.24**
	20	36.43	**59.4**	38.4	**59.92**	36.43	**59.4**	29.92	**59.53**
50%	50	36.12	**59.48**	36.16	**59.72**	36.12	**59.48**	29.53	**59.5**
	100	34.6	**59.22**	34.74	**58.87**	34.6	**59.22**	28.06	**58.91**

Table B.12
Experiment results of DeepLOB trained on KGHM and tested on PKNORLEN for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	63.99	**64.33**	65.7	**65.61**	63.99	**64.33**	64.41	**64.68**
0%	50	61.59	**61.87**	61.44	**62.04**	61.59	**61.87**	61.47	**61.91**
	100	**62.23**	61.89	**61.82**	61.16	**62.23**	61.89	**61.96**	61.35
	20	49.67	**63.96**	49.53	**65.1**	49.67	**63.96**	45.56	**64.28**
1%	50	50.82	**61.51**	48.49	**61.77**	50.82	**61.51**	46.89	**61.59**
	100	50.1	**61.71**	49.44	**60.99**	50.1	**61.71**	47.5	**61.17**
	20	41.57	**62.56**	43.0	**63.21**	41.57	**62.56**	34.66	**62.75**
5%	50	41.09	**60.13**	39.05	**60.81**	41.09	**60.13**	34.39	**60.36**
	100	40.67	**61.01**	40.63	**60.39**	40.67	**61.01**	32.76	**60.52**
	20	40.27	**61.1**	41.14	**61.34**	40.27	**61.1**	32.97	**61.12**
10%	50	39.17	**58.64**	37.72	**59.87**	39.17	**58.64**	32.43	**59.03**
	100	38.7	**60.14**	38.46	**59.68**	38.7	**60.14**	30.49	**59.72**
	20	38.86	**58.34**	40.39	**58.11**	38.86	**58.34**	31.65	**57.94**
25%	50	36.79	**55.32**	33.03	**58.12**	36.79	**55.32**	30.26	**55.98**
	100	36.06	**58.21**	36.83	**58.36**	36.06	**58.21**	28.42	**58.02**
	20	38.44	**57.03**	37.59	**56.56**	38.44	**57.03**	31.96	**56.4**
50%	50	35.55	**53.64**	32.85	**57.17**	35.55	**53.64**	29.1	**54.35**
	100	34.75	**57.09**	34.87	**57.74**	34.75	**57.09**	27.63	**57.06**

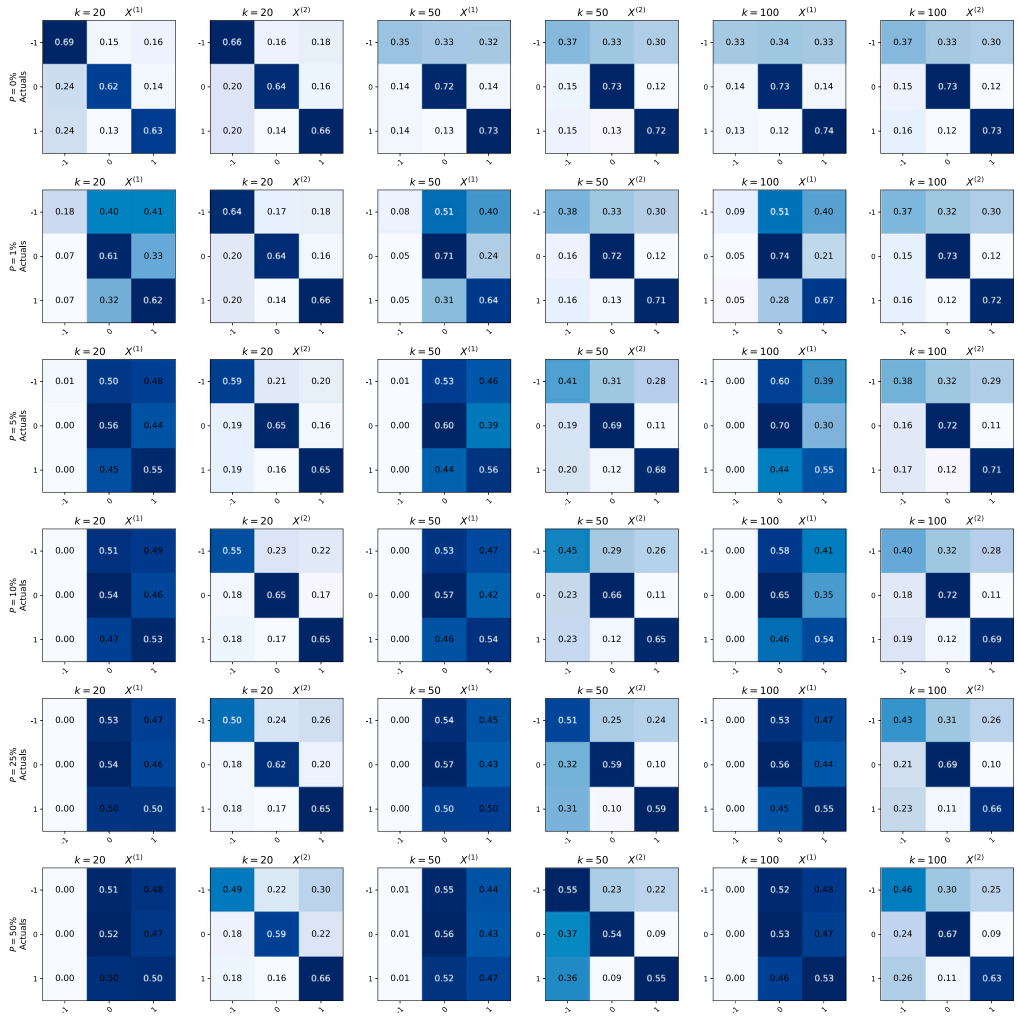

Fig. A.11. Confusion matrices for corresponding experimental results in Table A.8.

Table B.13
Experiment results of DeepLOB trained on KGHM and tested on PKOBP for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	64.07	**64.48**	**67.31**	65.92	64.07	**64.48**	64.27	**64.71**
0%	50	60.74	**61.03**	**60.62**	60.57	60.74	**61.03**	**60.62**	60.5
	100	59.86	**60.52**	**60.71**	59.94	59.86	**60.52**	**60.07**	59.88
	20	62.16	**64.49**	63.84	**65.92**	62.16	**64.49**	62.35	**64.71**
1%	50	59.42	**61.04**	58.98	**60.58**	59.42	**61.04**	59.03	**60.5**
	100	58.54	**60.51**	59.15	**59.93**	58.54	**60.51**	58.65	**59.87**
	20	56.15	**64.49**	56.05	**65.92**	56.15	**64.49**	55.89	**64.71**
5%	50	54.97	**61.04**	54.49	**60.58**	54.97	**61.04**	53.6	**60.51**
	100	53.91	**60.51**	54.15	**59.93**	53.91	**60.51**	53.54	**59.87**
	20	51.58	**64.49**	51.81	**65.92**	51.58	**64.49**	50.29	**64.71**
10%	50	51.22	**61.04**	51.34	**60.58**	51.22	**61.04**	49.0	**60.5**
	100	50.1	**60.52**	50.61	**59.93**	50.1	**60.52**	49.13	**59.87**
	20	46.6	**64.49**	47.83	**65.92**	46.6	**64.49**	43.68	**64.72**
25%	50	46.32	**61.04**	47.7	**60.58**	46.32	**61.04**	42.87	**60.51**
	100	45.42	**60.51**	46.91	**59.93**	45.42	**60.51**	43.31	**59.87**
	20	45.21	**64.49**	46.83	**65.92**	45.21	**64.49**	41.72	**64.71**
50%	50	44.75	**61.04**	46.66	**60.58**	44.75	**61.04**	40.74	**60.5**
	100	43.88	**60.52**	45.99	**59.94**	43.88	**60.52**	40.97	**59.88**

Table B.14
Experiment results of DeepLOB trained on KGHM and tested on PZU for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	62.29	**63.4**	63.7	**64.26**	62.29	**63.4**	62.59	**63.62**
0%	50	60.15	**61.55**	59.37	**60.89**	60.15	**61.55**	59.56	**60.99**
	100	59.42	**61.57**	59.88	**60.54**	59.42	**61.57**	59.54	**60.73**
	20	60.42	**63.4**	60.88	**64.26**	60.42	**63.4**	60.56	**63.62**
1%	50	58.93	**61.55**	57.86	**60.89**	58.93	**61.55**	57.94	**61.0**
	100	58.05	**61.57**	58.2	**60.54**	58.05	**61.57**	57.97	**60.73**
	20	54.82	**63.4**	54.7	**64.26**	54.82	**63.4**	54.06	**63.62**
5%	50	55.06	**61.55**	53.86	**60.89**	55.06	**61.55**	52.9	**60.99**
	100	53.97	**61.57**	53.61	**60.54**	53.97	**61.57**	53.16	**60.73**
	20	50.79	**63.4**	51.32	**64.27**	50.79	**63.4**	48.9	**63.62**
10%	50	51.96	**61.55**	51.16	**60.89**	51.96	**61.55**	48.97	**60.99**
	100	50.79	**61.57**	50.46	**60.54**	50.79	**61.57**	49.3	**60.73**
	20	46.57	**63.4**	48.3	**64.26**	46.57	**63.4**	43.16	**63.61**
25%	50	48.12	**61.55**	48.18	**60.89**	48.12	**61.55**	44.09	**60.99**
	100	47.37	**61.57**	47.4	**60.54**	47.37	**61.57**	44.63	**60.74**
	20	45.6	**63.4**	47.68	**64.26**	45.6	**63.4**	41.75	**63.62**
50%	50	47.15	**61.55**	47.51	**60.89**	47.15	**61.55**	42.7	**61.0**
	100	46.51	**61.56**	46.76	**60.53**	46.51	**61.56**	42.93	**60.73**

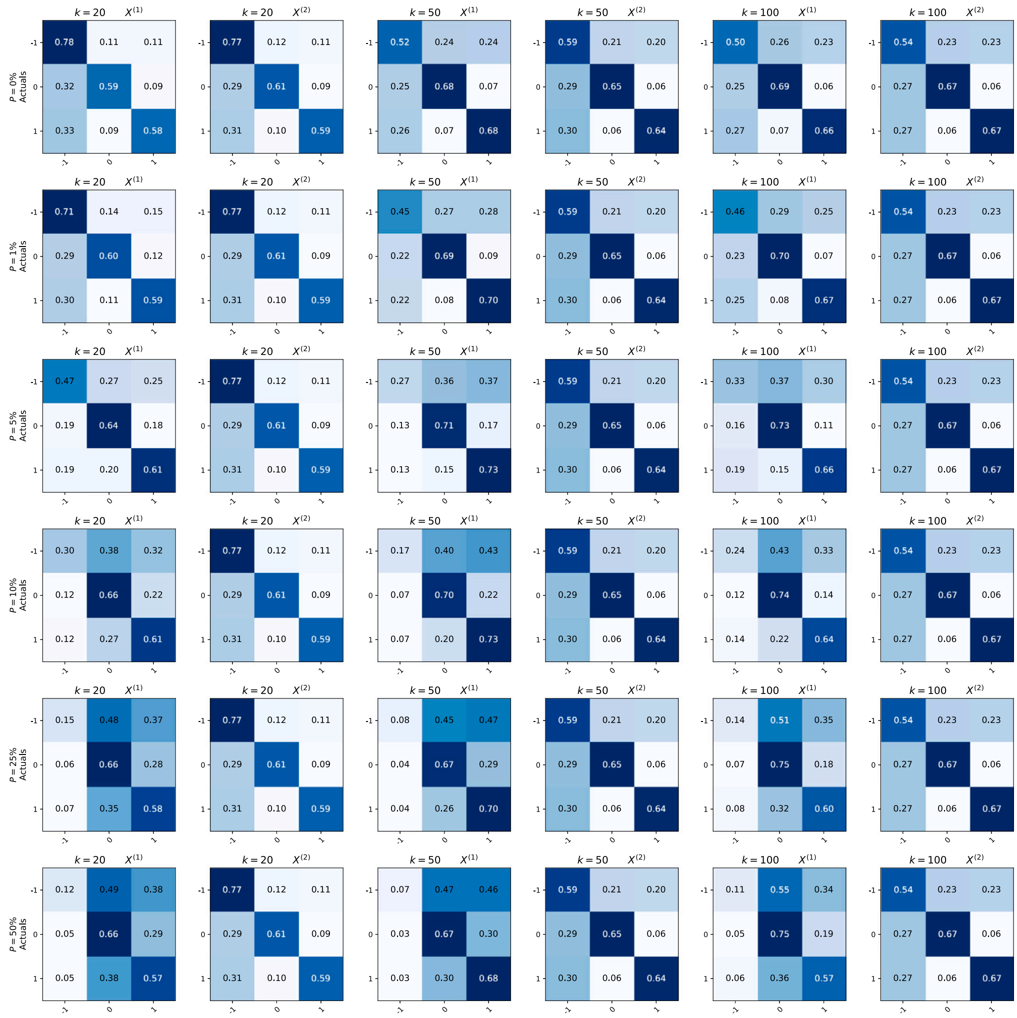

Fig. A.12. Confusion matrices for corresponding experimental results in Table A.9.

Table B.15
Experiment results of DeepLOB trained on KGHM and tested on PEKAO for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	**63.15**	63.03	**63.77**	63.51	**63.15**	63.03	**63.34**	63.19
0%	50	**61.82**	61.63	60.98	**61.03**	**61.82**	61.63	61.11	**61.17**
	100	**60.45**	59.53	**59.93**	58.67	**60.45**	59.53	**60.07**	58.79
	20	46.39	**62.79**	48.63	**63.19**	46.39	**62.79**	40.48	**62.92**
1%	50	49.08	**61.48**	48.05	**60.91**	49.08	**61.48**	43.88	**61.05**
	100	48.16	**59.41**	47.96	**58.56**	48.16	**59.41**	45.43	**58.68**
	20	38.26	**61.71**	43.1	**61.83**	38.26	**61.71**	30.21	**61.74**
5%	50	39.14	**60.85**	40.99	**60.4**	39.14	**60.85**	31.7	**60.51**
	100	38.95	**59.03**	40.05	**58.21**	38.95	**59.03**	30.99	**58.31**
	20	36.97	**60.69**	43.14	**60.62**	36.97	**60.69**	28.63	**60.59**
10%	50	37.42	**60.2**	41.73	**59.94**	37.42	**60.2**	29.97	**59.97**
	100	37.46	**58.62**	38.52	**57.87**	37.46	**58.62**	28.73	**57.94**
	20	36.0	**58.52**	42.86	**58.25**	36.0	**58.52**	27.78	**58.1**
25%	50	35.8	**58.45**	35.83	**58.87**	35.8	**58.45**	28.54	**58.5**
	100	35.74	**57.6**	37.08	**57.12**	35.74	**57.6**	27.41	**57.1**
	20	35.87	**57.38**	39.81	**57.04**	35.87	**57.38**	28.25	**56.78**
50%	50	35.33	**57.38**	38.41	**58.28**	35.33	**57.38**	28.17	**57.56**
	100	34.92	**57.08**	36.28	**56.83**	34.92	**57.08**	27.13	**56.71**

Table C.16
Experiment results of CNN on KGHM for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	**65.83**	65.28	**69.52**	68.81	**65.83**	65.28	**66.17**	65.58
0%	50	**62.1**	61.92	**62.28**	62.06	**62.1**	61.92	**62.16**	61.96
	100	**61.05**	61.03	60.54	**60.99**	**61.05**	61.03	60.61	**60.91**
	20	59.19	**65.15**	61.44	**68.51**	59.19	**65.15**	59.4	**65.45**
1%	50	49.83	**61.84**	56.35	**62.0**	49.83	**61.84**	50.27	**61.9**
	100	46.05	**60.99**	51.67	**60.96**	46.05	**60.99**	46.24	**60.87**
	20	49.55	**64.64**	51.22	**67.5**	49.55	**64.64**	49.47	**64.94**
5%	50	39.01	**61.51**	47.12	**61.79**	39.01	**61.51**	36.84	**61.62**
	100	37.66	**60.74**	43.45	**60.78**	37.66	**60.74**	35.24	**60.65**
	20	47.26	**64.2**	48.77	**66.69**	47.26	**64.2**	47.02	**64.5**
10%	50	37.93	**61.1**	44.68	**61.53**	37.93	**61.1**	35.68	**61.27**
	100	36.64	**60.49**	40.85	**60.61**	36.64	**60.49**	34.34	**60.43**
	20	46.23	**63.16**	49.21	**64.96**	46.23	**63.16**	45.36	**63.44**
25%	50	37.35	**60.24**	45.71	**61.07**	37.35	**60.24**	33.98	**60.53**
	100	36.07	**59.82**	40.33	**60.1**	36.07	**59.82**	33.05	**59.8**
	20	47.74	**62.25**	53.97	**63.42**	47.74	**62.25**	46.46	**62.46**
50%	50	38.9	**60.03**	50.54	**61.0**	38.9	**60.03**	35.3	**60.34**
	100	36.44	**59.34**	43.36	**59.71**	36.44	**59.34**	32.42	**59.35**

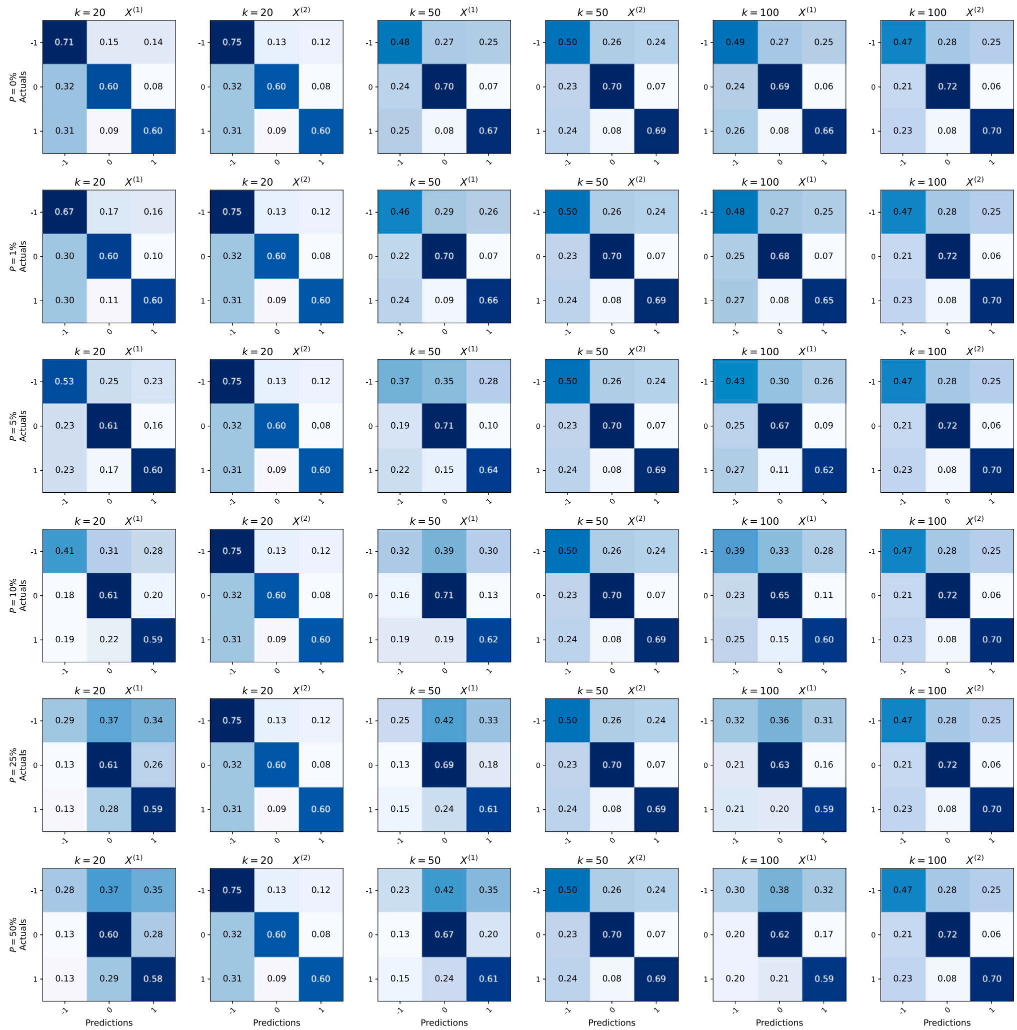

Fig. A.13. Confusion matrices for corresponding experimental results in Table A.10.

Table C.17
Experiment results of LSTM on KGHM for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	**64.44**	63.07	**68.12**	66.69	**64.44**	63.07	**64.77**	63.31
0%	50	**59.49**	59.37	**59.26**	58.98	**59.49**	59.37	**59.36**	59.11
	100	56.86	**57.18**	56.11	**56.25**	56.86	**57.18**	56.39	**56.53**
	20	58.85	**63.03**	59.26	**66.64**	58.85	**63.03**	58.97	**63.27**
1%	50	55.45	**59.34**	57.41	**58.97**	55.45	**59.34**	56.07	**59.09**
	100	53.73	**57.14**	53.87	**56.24**	53.73	**57.14**	53.72	**56.51**
	20	48.35	**62.9**	47.98	**66.49**	48.35	**62.9**	47.41	**63.14**
5%	50	47.53	**59.18**	51.1	**58.9**	47.53	**59.18**	48.06	**58.99**
	100	47.19	**56.97**	47.01	**56.16**	47.19	**56.97**	46.8	**56.41**
	20	44.4	**62.75**	44.06	**66.3**	44.4	**62.75**	43.31	**62.99**
10%	50	44.08	**59.04**	46.27	**58.85**	44.08	**59.04**	44.27	**58.9**
	100	43.4	**56.76**	42.53	**56.07**	43.4	**56.76**	42.24	**56.29**
	20	39.98	**62.37**	39.96	**65.9**	39.98	**62.37**	39.25	**62.6**
25%	50	39.91	**58.57**	40.34	**58.64**	39.91	**58.57**	39.4	**58.56**
	100	38.32	**56.22**	37.81	**55.79**	38.32	**56.22**	37.16	**55.93**
	20	37.02	**62.1**	37.39	**65.6**	37.02	**62.1**	36.22	**62.33**
50%	50	37.27	**58.2**	37.22	**58.43**	37.27	**58.2**	35.49	**58.27**
	100	35.39	**55.72**	35.38	**55.52**	35.39	**55.72**	34.38	**55.56**

Table C.18
Experiment results of C(TABL) on KGHM for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	**65.53**	64.13	**70.08**	67.46	**65.53**	64.13	**65.83**	64.41
0%	50	**62.05**	60.88	**61.09**	60.62	**62.05**	60.88	**61.35**	60.73
	100	**60.64**	59.8	**59.79**	59.13	**60.64**	59.8	**60.03**	59.37
	20	**64.43**	64.11	**69.18**	67.42	**64.43**	64.11	**64.65**	64.4
1%	50	60.21	**60.88**	**61.08**	60.62	60.21	**60.88**	60.48	**60.72**
	100	57.17	**59.8**	56.32	**59.13**	57.17	**59.8**	56.49	**59.37**
	20	61.09	**64.03**	64.37	**67.22**	61.09	**64.03**	61.08	**64.32**
5%	50	54.03	**60.84**	59.5	**60.59**	54.03	**60.84**	54.46	**60.69**
	100	50.25	**59.78**	48.66	**59.12**	50.25	**59.78**	47.5	**59.36**
	20	58.65	**63.94**	60.64	**67.0**	58.65	**63.94**	58.57	**64.23**
10%	50	51.35	**60.82**	56.96	**60.58**	51.35	**60.82**	51.28	**60.68**
	100	46.99	**59.77**	45.72	**59.11**	46.99	**59.77**	42.58	**59.35**
	20	55.66	**63.75**	56.5	**66.52**	55.66	**63.75**	55.56	**64.03**
25%	50	49.1	**60.73**	53.73	**60.52**	49.1	**60.73**	48.49	**60.6**
	100	43.94	**59.74**	43.29	**59.1**	43.94	**59.74**	38.81	**59.34**
	20	54.21	**63.58**	54.98	**66.15**	54.21	**63.58**	54.06	**63.87**
50%	50	47.53	**60.66**	51.88	**60.47**	47.53	**60.66**	46.66	**60.54**
	100	43.28	**59.71**	42.28	**59.1**	43.28	**59.71**	39.46	**59.32**

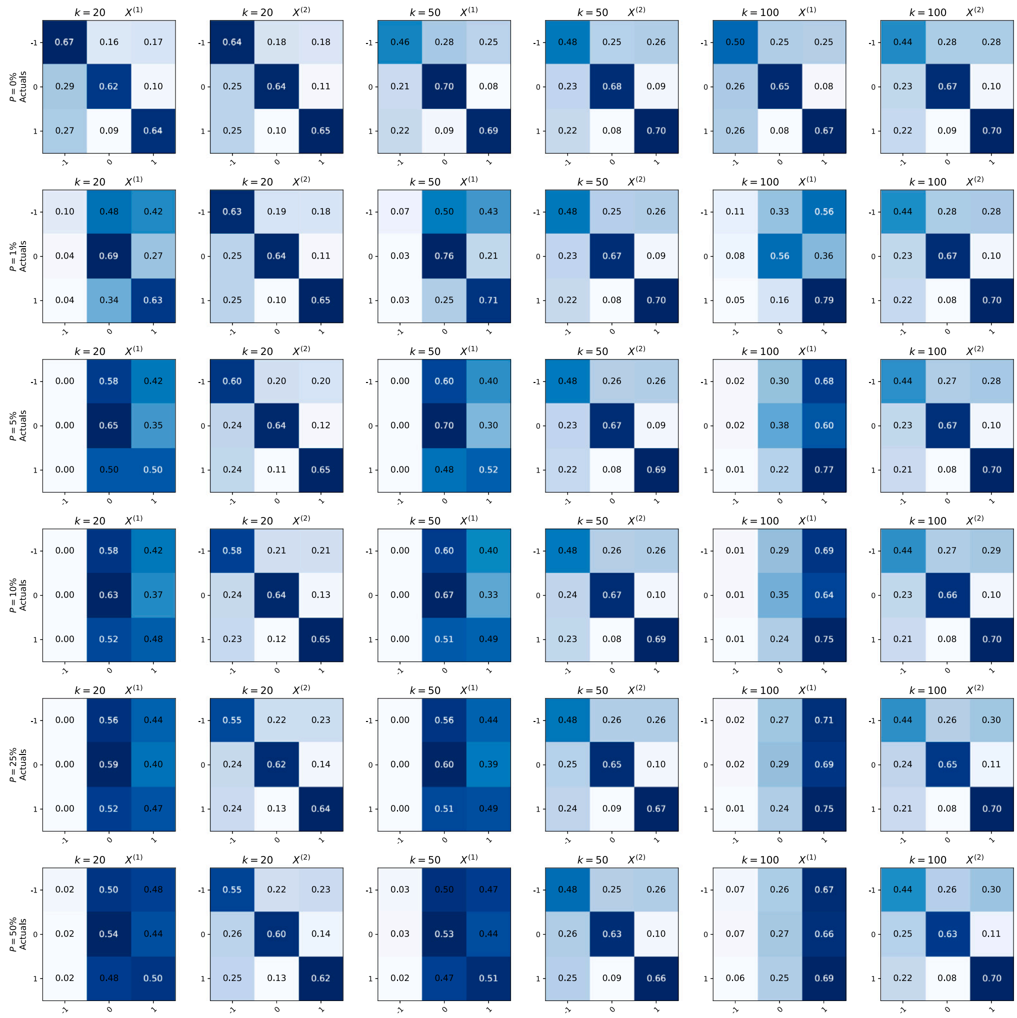

Fig. A.14. Confusion matrices for corresponding experimental results in Table A.11.

Table C.19
Experiment results of DeepLOB-Seq2Seq on KGHM for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	65.72	**66.33**	68.3	**69.47**	65.72	**66.33**	66.06	**66.64**
0%	50	**63.27**	63.2	62.8	**62.93**	**63.27**	63.2	62.94	**63.03**
	100	**63.22**	63.03	**62.96**	62.69	**63.22**	63.03	**63.06**	62.8
	20	50.18	**65.96**	50.28	**68.78**	50.18	**65.96**	46.56	**66.26**
1%	50	51.08	**62.97**	49.38	**62.77**	51.08	**62.97**	47.39	**62.84**
	100	50.73	**62.9**	49.27	**62.59**	50.73	**62.9**	46.98	**62.69**
	20	41.21	**64.44**	44.1	**66.22**	41.21	**64.44**	34.03	**64.71**
5%	50	41.66	**61.98**	41.09	**62.1**	41.66	**61.98**	34.14	**62.03**
	100	40.89	**62.22**	41.97	**62.11**	40.89	**62.22**	32.86	**62.12**
	20	39.86	**62.88**	44.69	**63.89**	39.86	**62.88**	32.77	**63.1**
10%	50	40.4	**60.91**	40.98	**61.53**	40.4	**60.91**	33.11	**61.16**
	100	39.82	**61.45**	42.31	**61.65**	39.82	**61.45**	32.18	**61.5**
	20	38.26	**59.91**	42.04	**60.26**	38.26	**59.91**	30.82	**59.99**
25%	50	38.68	**58.24**	41.27	**60.5**	38.68	**58.24**	31.72	**58.87**
	100	38.16	**59.11**	39.55	**60.58**	38.16	**59.11**	31.14	**59.52**
	20	37.24	**58.62**	37.14	**58.98**	37.24	**58.62**	29.75	**58.66**
50%	50	37.23	**56.61**	38.83	**59.95**	37.23	**56.61**	30.01	**57.34**
	100	36.58	**57.41**	39.17	**59.92**	36.58	**57.41**	29.44	**57.93**

Table C.20
Experiment results of DeepLOB-Attention on KGHM for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
	20	65.6	**66.48**	**68.88**	68.76	65.6	**66.48**	65.94	**66.77**
0%	50	63.14	**63.36**	**63.03**	62.7	63.14	**63.36**	**63.05**	62.94
	100	62.99	**63.09**	**63.2**	62.7	62.99	**63.09**	**63.06**	62.84
	20	50.81	**66.03**	50.54	**68.07**	50.81	**66.03**	48.87	**66.31**
1%	50	50.73	**63.1**	49.53	**62.49**	50.73	**63.1**	49.24	**62.72**
	100	50.34	**62.93**	49.48	**62.54**	50.34	**62.93**	48.7	**62.68**
	20	41.03	**64.34**	43.94	**65.54**	41.03	**64.34**	33.61	**64.57**
5%	50	42.05	**62.04**	40.41	**61.69**	42.05	**62.04**	34.42	**61.83**
	100	41.45	**62.17**	40.47	**61.82**	41.45	**62.17**	33.73	**61.93**
	20	39.54	**62.53**	43.0	**63.11**	39.54	**62.53**	32.21	**62.67**
10%	50	40.44	**60.97**	42.76	**60.97**	40.44	**60.97**	33.02	**60.95**
	100	39.87	**61.35**	38.88	**61.12**	39.87	**61.35**	32.19	**61.17**
	20	38.11	**59.02**	40.15	**58.92**	38.11	**59.02**	30.96	**58.76**
25%	50	38.31	**58.41**	34.39	**59.33**	38.31	**58.41**	30.59	**58.7**
	100	37.96	**59.12**	35.54	**59.41**	37.96	**59.12**	30.04	**59.12**
	20	37.14	**57.19**	41.01	**57.03**	37.14	**57.19**	29.36	**56.58**
50%	50	36.98	**56.89**	38.51	**58.34**	36.98	**56.89**	28.57	**57.24**
	100	36.53	**57.5**	41.8	**58.24**	36.53	**57.5**	28.06	**57.64**

Table C.21
Experiment results of DeepLOB-Attention on PKNORLEN for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
0%	20	63.89	**64.89**	65.77	**66.34**	63.89	**64.89**	64.33	**65.24**
	50	62.48	**62.9**	61.21	**61.67**	62.48	**62.9**	61.15	**61.64**
	100	62.23	**63.26**	61.18	**62.26**	62.23	**63.26**	60.77	**61.86**
1%	20	51.78	**64.53**	51.49	**65.71**	51.78	**64.53**	49.38	**64.84**
	50	51.03	**62.69**	49.64	**61.48**	51.03	**62.69**	46.43	**61.52**
	100	50.51	**63.12**	49.74	**62.1**	50.51	**63.12**	45.05	**61.81**
5%	20	42.16	**63.23**	45.02	**63.69**	42.16	**63.23**	34.91	**63.37**
	50	40.94	**61.8**	45.08	**60.78**	40.94	**61.8**	32.51	**61.0**
	100	40.35	**62.51**	45.81	**61.51**	40.35	**62.51**	32.0	**61.55**
10%	20	40.51	**62.0**	44.08	**62.06**	40.51	**62.0**	32.41	**61.97**
	50	39.11	**60.87**	45.87	**60.2**	39.11	**60.87**	29.88	**60.41**
	100	38.2	**61.69**	43.86	**60.88**	38.2	**61.69**	29.36	**61.07**
25%	20	39.21	**59.97**	42.69	**59.73**	39.21	**59.97**	30.95	**59.69**
	50	37.39	**58.85**	38.12	**59.1**	37.39	**58.85**	28.04	**58.91**
	100	36.09	**59.74**	35.98	**59.63**	36.09	**59.74**	27.2	**59.64**
50%	20	38.67	**59.03**	40.58	**58.83**	38.67	**59.03**	31.84	**58.71**
	50	36.54	**57.71**	35.98	**58.41**	36.54	**57.71**	28.64	**57.96**
	100	35.0	**58.6**	34.74	**58.86**	35.0	**58.6**	27.39	**58.69**

Table C.22
Experiment results of DeepLOB-Attention on PKOBP for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
0%	20	64.51	**65.25**	67.6	**68.41**	64.51	**65.25**	64.73	**65.45**
	50	62.04	**62.87**	61.67	**63.21**	62.04	**62.87**	61.75	**62.96**
	100	61.99	**63.11**	61.6	**63.1**	61.99	**63.11**	61.7	**63.09**
1%	20	62.86	**65.25**	64.34	**68.41**	62.86	**65.25**	63.07	**65.45**
	50	60.52	**62.87**	59.89	**63.21**	60.52	**62.87**	59.7	**62.96**
	100	60.48	**63.11**	59.87	**63.1**	60.48	**63.11**	59.71	**63.09**
5%	20	56.98	**65.25**	56.68	**68.41**	56.98	**65.25**	56.41	**65.45**
	50	55.0	**62.87**	55.72	**63.21**	55.0	**62.87**	51.38	**62.96**
	100	54.76	**63.11**	55.52	**63.11**	54.76	**63.11**	51.37	**63.09**
10%	20	52.33	**65.25**	52.69	**68.41**	52.33	**65.25**	49.88	**65.45**
	50	50.84	**62.87**	53.82	**63.2**	50.84	**62.87**	44.73	**62.96**
	100	50.34	**63.11**	53.47	**63.11**	50.34	**63.11**	44.58	**63.09**
25%	20	47.43	**65.25**	49.05	**68.41**	47.43	**65.25**	42.39	**65.45**
	50	46.39	**62.87**	52.45	**63.2**	46.39	**62.87**	38.59	**62.96**
	100	45.88	**63.11**	52.27	**63.11**	45.88	**63.11**	38.27	**63.09**
50%	20	46.16	**65.25**	48.28	**68.41**	46.16	**65.25**	40.56	**65.45**
	50	44.9	**62.87**	51.87	**63.21**	44.9	**62.87**	36.81	**62.96**
	100	44.28	**63.11**	51.8	**63.11**	44.28	**63.11**	36.3	**63.09**

Table C.23
Experiment results of DeepLOB-Attention on PZU for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
0%	20	62.95	**64.75**	66.87	**68.56**	62.95	**64.75**	63.26	**65.05**
	50	62.18	**63.53**	62.26	**63.94**	62.18	**63.53**	62.21	**63.69**
	100	62.38	**64.3**	62.44	**64.73**	62.38	**64.3**	62.4	**64.47**
1%	20	62.1	**64.75**	65.11	**68.56**	62.1	**64.75**	62.38	**65.05**
	50	61.27	**63.53**	61.09	**63.94**	61.27	**63.53**	61.16	**63.69**
	100	61.37	**64.3**	61.21	**64.73**	61.37	**64.3**	61.26	**64.47**
5%	20	58.88	**64.75**	59.72	**68.56**	58.88	**64.75**	59.05	**65.05**
	50	58.18	**63.52**	57.36	**63.94**	58.18	**63.52**	57.35	**63.69**
	100	58.2	**64.3**	57.4	**64.73**	58.2	**64.3**	57.4	**64.47**
10%	20	56.06	**64.75**	56.12	**68.56**	56.06	**64.75**	56.0	**65.05**
	50	55.64	**63.52**	54.79	**63.93**	55.64	**63.52**	54.02	**63.69**
	100	55.68	**64.3**	54.7	**64.73**	55.68	**64.3**	54.13	**64.47**
25%	20	52.48	**64.75**	52.44	**68.56**	52.48	**64.75**	51.84	**65.05**
	50	52.31	**63.52**	51.98	**63.93**	52.31	**63.52**	49.47	**63.69**
	100	52.46	**64.3**	51.71	**64.73**	52.46	**64.3**	49.75	**64.47**
50%	20	51.82	**64.75**	51.83	**68.56**	51.82	**64.75**	51.16	**65.05**
	50	51.27	**63.52**	51.1	**63.93**	51.27	**63.52**	48.18	**63.69**
	100	51.34	**64.3**	50.73	**64.73**	51.34	**64.3**	48.35	**64.47**

Table C.24
Experiment results of DeepLOB-Attention on PEKAO for various perturbation levels (P), prediction horizons (k) and types of inputs (level-based $X^{(1)}$ and proposed $X^{(2)}$).

P	k	Metrics (%)							
		Accuracy		Precision		Recall		F1-score	
		$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$	$X^{(1)}$	$X^{(2)}$
0%	20	63.33	**63.96**	64.83	**64.98**	63.33	**63.96**	63.65	**64.21**
	50	62.48	**62.69**	**62.43**	62.37	62.48	**62.69**	62.36	**62.49**
	100	61.22	**61.9**	61.15	**61.61**	61.22	**61.9**	61.02	**61.72**
1%	20	50.76	**63.8**	52.4	**64.77**	50.76	**63.8**	45.98	**64.04**
	50	50.83	**62.61**	55.11	**62.29**	50.83	**62.61**	42.88	**62.41**
	100	48.97	**61.84**	53.7	**61.55**	48.97	**61.84**	40.7	**61.66**
5%	20	41.01	**63.19**	48.05	**63.99**	41.01	**63.19**	33.16	**63.39**
	50	40.69	**62.24**	48.9	**61.95**	40.69	**62.24**	31.85	**62.05**
	100	39.59	**61.52**	46.62	**61.25**	39.59	**61.52**	30.13	**61.35**
10%	20	39.16	**62.61**	45.42	**63.32**	39.16	**62.61**	31.4	**62.79**
	50	38.76	**61.83**	49.14	**61.6**	38.76	**61.83**	30.1	**61.68**
	100	37.91	**61.19**	44.45	**60.98**	37.91	**61.19**	28.69	**61.04**
25%	20	37.53	**61.55**	39.58	**62.33**	37.53	**61.55**	30.06	**61.72**
	50	36.93	**60.81**	35.71	**60.93**	36.93	**60.81**	29.04	**60.78**
	100	36.31	**60.24**	38.36	**60.32**	36.31	**60.24**	28.11	**60.19**
50%	20	37.16	**60.91**	39.64	**62.04**	37.16	**60.91**	31.0	**61.12**
	50	36.49	**60.05**	37.82	**60.73**	36.49	**60.05**	30.21	**60.19**
	100	35.77	**59.47**	37.01	**60.06**	35.77	**59.47**	29.18	**59.57**

References

[1] C.A. Parlour, D.J. Seppi, CHAPTER 3 - limit order markets: A survey, in: A.V. Thakor, A.W. Boot (Eds.), Handbook of Financial Intermediation and Banking, in: Handbooks in Finance, Elsevier, San Diego, 2008, pp. 63–96, http://dx.doi.org/10.1016/B978-044451558-2.50007-6, URL https://www.sciencedirect.com/science/article/pii/B9780444515582500076.

[2] I. Rosu, et al., Liquidity and Information in Order Driven Markets, Chicago Booth School of Business, Citeseer, 2010.

[3] H. Buehler, L. Gonon, J. Teichmann, B. Wood, Deep hedging, Quant. Finance 19 (2019) 1–21, http://dx.doi.org/10.1080/14697688.2019.1571683.

[4] M. Wiese, R. Knobloch, R. Korn, P. Kretschmer, Quant GANs: deep generation of financial time series, Quant. Finance 20 (9) (2020) 1419–1440.

[5] A. Ang, G. Bekaert, Stock return predictability: Is it there? Rev. Financ. Stud. 20 (3) (2006) 651–707, http://dx.doi.org/10.1093/rfs/hhl021, arXiv:https://academic.oup.com/rfs/article-pdf/20/3/651/24428929/hhl021.pdf.

[6] P. Bacchetta, E. Mertens, E. van Wincoop, Predictability in financial markets: What do survey expectations tell us? J. Int. Money Finance 28 (3) (2009) 406–426, http://dx.doi.org/10.1016/j.jimonfin.2008.09.001, URL https://www.sciencedirect.com/science/article/pii/S026156060800140X.

[7] T. Bollerslev, J. Marrone, L. Xu, H. Zhou, Stock return predictability and variance risk premia: Statistical inference and international evidence, J. Financ. Quant. Anal. 49 (3) (2014) 633–661, http://dx.doi.org/10.1017/S0022109014000453.

[8] M.A. Ferreira, P. Santa-Clara, Forecasting stock market returns: The sum of the parts is more than the whole, J. Financ. Econ. 100 (3) (2011) 514–537, http://dx.doi.org/10.1016/j.jfineco.2011.02.003, URL https://www.sciencedirect.com/science/article/pii/S0304405X11000365.

[9] B. Mandelbrot, R.L. Hudson, The Misbehavior of Markets: A Fractal View of Financial Turbulence, Basic books, 2007.

[10] B.B. Mandelbrot, How fractals can explain what's wrong with Wall Street, Sci. Am. 15 (9) (2008) 2008.

[11] B. Zheng, E. Moulines, F. Abergel, Price jump prediction in limit order book, J. Math. Finance 03 (2012) http://dx.doi.org/10.2139/ssrn.2026454.

[12] J. Liu, S. Park, Behind stock price movement: Supply and demand in market microstructure and market influence, J. Trading 10 (2015) 150610092100003, http://dx.doi.org/10.3905/jot.2015.2015.1.042.

[13] J. Levendovszky, F. Kia, Prediction based - High frequency trading on financial time series, Period. Polytech. Electr. Eng. 56 (2012) 29, http://dx.doi.org/10.3311/PPee.7165.

[14] J. Sirignano, Deep learning for limit order books, Quant. Finance 19 (2018) 1–22, http://dx.doi.org/10.1080/14697688.2018.1546053.

[15] D. Bogoev, A. Karam, An empirical detection of high frequency trading strategies, in: 6th International Conference of the Financial Engineering and Banking Society, 2016, pp. 10–12.

[16] M. Dixon, High frequency market making with machine learning, 2016, November.

[17] M. Rehman, G.M. Khan, S.A. Mahmud, Foreign currency exchange rates prediction using cgp and recurrent neural network, IERI Procedia 10 (2014) 239–244.
[18] S. Galeshchuk, Neural networks performance in exchange rate prediction, Neurocomputing 172 (2016) 446–452.
[19] R. Majhi, G. Panda, G. Sahoo, Development and performance evaluation of FLANN based model for forecasting of stock markets, Expert Syst. Appl. 36 (3) (2009) 6800–6808.
[20] A. Sharang, C. Rao, Using machine learning for medium frequency derivative portfolio trading, 2015, arXiv preprint arXiv:1512.06228.
[21] J. Hallgren, T. Koski, Testing for causality in continuous time bayesian network models of high-frequency data, 2016, arXiv preprint arXiv:1601.06651.
[22] J. Sandoval, G. Hernández, Computational visual analysis of the order book dynamics for creating high-frequency foreign exchange trading strategies, Procedia Comput. Sci. 51 (2015) 1593–1602.
[23] A. Ntakaris, M. Magris, J. Kanniainen, M. Gabbouj, A. Iosifidis, Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods, J. Forecast. 37 (8) (2018) 852–866, http://dx.doi.org/10.1002/for.2543, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/for.2543 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/for.2543.
[24] D.T. Tran, M. Magris, J. Kanniainen, M. Gabbouj, A. Iosifidis, Tensor representation in high-frequency financial data for price change prediction, in: 2017 IEEE Symposium Series on Computational Intelligence, SSCI, IEEE, 2017, pp. 1–7.
[25] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, A. Iosifidis, Using deep learning to detect price change indications in financial markets, in: 2017 25th European Signal Processing Conference, EUSIPCO, IEEE, 2017, pp. 2511–2515.
[26] M. Mahfouz, A. Filos, C. Chtourou, J. Lockhart, S.A. Assefa, M.M. Veloso, D. Mandić, T.H. Balch, On the importance of opponent modeling in auction markets, 2019, arXiv: Computational Finance. URL https://api.semanticscholar.org/CorpusID:208513898.
[27] J. Sirignano, R. Cont, Universal features of price formation in financial markets: perspectives from deep learning, Quant. Finance 19 (9) (2019) 1449–1459.
[28] N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, A. Iosifidis, Temporal bag-of-features learning for predicting mid price movements using high frequency limit order book data, IEEE Trans. Emerg. Top. Comput. Intell. 4 (6) (2020) 774–785, http://dx.doi.org/10.1109/TETCI.2018.2872598.
[29] D.T. Tran, A. Iosifidis, J. Kanniainen, M. Gabbouj, Temporal attention-augmented bilinear network for financial time-series data analysis, IEEE Trans. Neural Netw. Learn. Syst. 30 (5) (2018) 1407–1418.
[30] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, A. Iosifidis, Forecasting stock prices from the limit order book using convolutional neural networks, in: 2017 IEEE 19th Conference on Business Informatics, Vol. 1, CBI, IEEE, 2017, pp. 7–12.
[31] Z. Zhang, S. Zohren, S. Roberts, Bdlob: Bayesian deep convolutional neural networks for limit order books, 2018, arXiv preprint arXiv:1811.10041.
[32] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, A. Iosifidis, Using deep learning for price prediction by exploiting stationary limit order book features, Appl. Soft Comput. 93 (2020) 106401.
[33] Z. Zhang, S. Zohren, S. Roberts, DeepLOB: Deep convolutional neural networks for limit order books, IEEE Trans. Signal Process. 67 (11) (2019) 3001–3012, http://dx.doi.org/10.1109/TSP.2019.2907260.
[34] J. Wallbridge, Transformers for limit order books, Econom. Model. Cap. Mark. - Asset Pricing eJ. (2020) URL https://api.semanticscholar.org/CorpusID:211677995.
[35] Z. Zhang, S. Zohren, Multi-horizon forecasting for limit order books: Novel deep learning approaches and hardware acceleration using intelligent processing units, 2021, arXiv:2105.10430.
[36] I. Zaznov, J. Kunkel, A. Dufour, A. Badii, Predicting stock price changes based on the limit order book: A survey, Mathematics 10 (8) (2022) http://dx.doi.org/10.3390/math10081234, URL https://www.mdpi.com/2227-7390/10/8/1234.
[37] Y. Wu, M. Mahfouz, D. Magazzeni, M. Veloso, How robust are limit order book representations under data perturbation?, 2021, arXiv:2110.04752.
[38] M.D. Gould, M.A. Porter, S. Williams, M. McDonald, D.J. Fenn, S.D. Howison, Limit order books, Quant. Finance 13 (11) (2013) 1709–1742, http://dx.doi.org/10.1080/14697688.2013.803148.
[39] R. Huang, T. Polak, LOBSTER: The Limit Order Book Reconstructor, Tech. Rep, School of Business and Economics, Humboldt Universität zu Berlin, 2011.
[40] W. Kosiński, P. Prokopowicz, D. Ślęzak, Drawback of fuzzy arthmetics - New intutions and propositions, Proc. Meth. Aritif. Intell. (2002) 231–237.
[41] W. Kosiński, P. Prokopowicz, D. Ślęzak, On algebraic operations on fuzzy numbers, in: M.A. Kłopotek, S.T. Wierzchoń, K. Trojanowski (Eds.), Intelligent Information Processing and Web Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 353–362.
[42] W. Kosiński, P. Prokopowicz, D. Ślęzak, Ordered fuzzy numbers, Bull. Pol. Acad. Sci. 51 (3) (2003) 327–338.
[43] W. Kosiński, P. Prokopowicz, Algebra of fuzzy numbers, Math. Appl. 32 (46/05) (2004).
[44] W. Kosiński, On soft computing and modelling, Imag. Process. Commun. 11 (2006) 71–82.
[45] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning—I, Inf. Sci. 8 (3) (1975) 199–249.
[46] D. Dubois, H. Prade, Operations on fuzzy numbers, Int. J. Syst. Sci. 9 (6) (1978) 613–626.
[47] A. Marszałek, T. Burczyński, Ordered fuzzy GARCH model for volatility forecasting, in: J. Kacprzyk, E. Szmidt, S. Zadrożny, K.T. Atanassov, M. Krawczak (Eds.), Advances in Fuzzy Logic and Technology 2017, Springer International Publishing, Cham, 2018, pp. 480–492.
[48] P. Prokopowicz, Flexible and simple methods of calculations on fuzzy numbers with the ordered fuzzy numbers model, in: L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh, J.M. Zurada (Eds.), Artificial Intelligence and Soft Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 365–375.
[49] P. Prokopowicz, The use of ordered fuzzy numbers for modelling changes in dynamic processes, Inform. Sci. 470 (2019) 1–14, http://dx.doi.org/10.1016/j.ins.2018.08.045, URL https://www.sciencedirect.com/science/article/pii/S0020025517303705.
[50] E. Roszkowska, D. Kacprzak, The fuzzy saw and fuzzy TOPSIS procedures based on ordered fuzzy numbers, Inform. Sci. 369 (2016) 564–584, http://dx.doi.org/10.1016/j.ins.2016.07.044, URL https://www.sciencedirect.com/science/article/pii/S0020025516305254.
[51] K. Piasecki, E. Roszkowska, On application of ordered fuzzy numbers in ranking linguistically evaluated negotiation offers, Adv. Fuzzy Syst. 2018 (2018) 1–12, http://dx.doi.org/10.1155/2018/1569860.
[52] Ł. Apiecionek, J.M. Czerniak, W.T. Dobrosielski, D. Ewald, New proposal of fuzzy observation of ddos attack, in: Uncertainty and Imprecision in Decision Making and Decision Support: Cross-Fertilization, New Models and Applications: Selected Papers from BOS-2016 and IWIFSGN-2016 Held on October 12-14, 2016 in Warsaw, Poland, Springer, 2018, pp. 240–250.
[53] P. Prokopowicz, J. Czerniak, D. Mikołajewski, Ł. Apiecionek, D. Ślęzak, Theory and Applications of Ordered Fuzzy Numbers: A Tribute to Professor Witold Kosiński, Springer Nature, 2017.
[54] A. Marszałek, T. Burczyński, Modeling and forecasting financial time series with ordered fuzzy candlesticks, Inform. Sci. 273 (2014) 144–155, http://dx.doi.org/10.1016/j.ins.2014.03.026, URL https://www.sciencedirect.com/science/article/pii/S0020025514003107.
[55] A. Marszałek, T. Burczyński, Fuzzy portfolio diversification with ordered fuzzy numbers, in: L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh, J.M. Zurada (Eds.), Artificial Intelligence and Soft Computing, Springer International Publishing, Cham, 2017, pp. 279–291.
[56] A. Marszałek, T. Burczyński, Ordered fuzzy random variables: Definition and the concept of normality, Inform. Sci. 545 (2021) 415–426, http://dx.doi.org/10.1016/j.ins.2020.08.120, URL https://www.sciencedirect.com/science/article/pii/S0020025520308860.
[57] L.E. Harris, Liquidity, Trading Rules, and Electronic Trading Systems, in: Monograph Series in Finance and Economics ; Monograph 1990-4, New York University Salomon Center, New York, 1991.
[58] T. Lybek, A. Sarr, Measuring liquidity in financial markets, 2003, http://dx.doi.org/10.5089/9781451875577.001, International Monetary Fund, IMF Working Papers 02.
[59] N.S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, P.T.P. Tang, On large-batch training for deep learning: Generalization gap and sharp minima, in: International Conference on Learning Representations, 2017, URL https://openreview.net/forum?id=H1oyRlYgg.
[60] F. Chollet, et al., Keras, 2015, GitHub. URL https://github.com/fchollet/keras.
[61] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, Software available from tensorflow.org, URL https://www.tensorflow.org/.
[62] W. Conover, Practical Nonparametric Statistics, 3. ed, Wiley Series in Probability and Statistics, Wiley, New York, NY [u.a.], 1999, URL http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+24551600X&sourceid=fbw_bibsonomy.