Artificial Intelligence Review (2024) 57:116
https://doi.org/10.1007/s10462-024-10715-4

Lob-based deep learning models for stock price trend prediction: a benchmark study

Matteo Prata[1] · Giuseppe Masi[1] · Leonardo Berti[1] · Viviana Arrigoni[1] · Andrea Coletta[2] · Irene Cannistraci[1] · Svitlana Vyetrenko[2] · Paola Velardi[1] · Novella Bartolini[1]

Accepted: 31 January 2024 / Published online: 13 April 2024

Abstract
The recent advancements in Deep Learning (DL) research have notably influenced the finance sector. We examine the robustness and generalizability of fifteen state-of-the-art DL models focusing on Stock Price Trend Prediction (SPTP) based on Limit Order Book (LOB) data. To carry out this study, we developed LOBCAST, an open-source framework that incorporates data preprocessing, DL model training, evaluation, and profit analysis. Our extensive experiments reveal that all models exhibit a significant performance drop when exposed to new data, thereby raising questions about their real-world market applicability. Our work serves as a benchmark, illuminating the potential and the limitations of current approaches and providing insight for innovative solutions.

✉ Matteo Prata
prata@di.uniroma1.it

Giuseppe Masi
masi.g@di.uniroma1.it

Leonardo Berti
berti.1883894@studenti.uniroma1.it

Viviana Arrigoni
viviana.arrigoni@uniroma1.it

Andrea Coletta
coletta@di.uniroma1.it

Irene Cannistraci
cannistraci@di.uniroma1.it

Svitlana Vyetrenko
svitlana.s.vyetrenko@jpmchase.com

Paola Velardi
velardi@di.uniroma1.it

Novella Bartolini
bartolini@di.uniroma1.it

[1] Department of Computer Science, Sapienza University of Rome, Rome, Italy

[2] J.P. Morgan AI Research, New York, NY, USA

Keywords Stock price trend prediction · Deep learning · Benchmark

1 Introduction

Predicting stock market prices is a complex endeavour due to the intrinsic and tangled nature of a multitude of factors that influence the market, including macroeconomic conditions, natural events, and investor sentiment Engle et al. (2013). Professional traders and researchers usually forecast price movements by understanding key market properties, such as volatility or liquidity, and recognizing patterns to anticipate future market trends Bouchaud et al. (2018). Effective mathematical models are essential for capturing complex market dependencies. The recent surge in artificial intelligence has led to significant work in using machine learning algorithms to predict future market trends (Cao 2022; Jiang 2021; Sezer et al. 2020). Recent Deep Learning (DL) models have achieved over 88% in F1-Score in predicting market trends in simulated settings using historical data Tran et al. (2021). However, replicating these performances in real markets is challenging, suggesting a possible *simulation-to-reality* gap (Liu et al. 2022; Zaznov et al. 2022).

Due to the broad interest in this field, in recent years, many researchers have proposed survey papers that study the vast literature on stock market predictions. Most of these surveys propose meticulous classification and analysis of the existing literature and review of the implemented models based on the results shown in the original papers. Nevertheless, most of these works consist of desk-based research, and only a few of them propose new experiments to validate the collected results. For the first time, in this paper, we benchmark the most recent and promising DL approaches to Stock Price Trend Prediction (SPTP) based on Limit Order Book (LOB) data, one of the most valuable information sources available to traders on the stock markets.

The LOB aggregates orders for shares of a given stock over time, characterizing each order by its associated price and volume of shares. SPTP is the problem of forecasting the stock price trend based on LOB data, classifying it as either *upward*, *downward*, or *stable*.

We compare novel data-driven approaches from Machine Learning (ML) and DL that analyze the market at its finest resolution, using high-frequency LOB data. Our benchmark evaluates their robustness and generalizability (Pineau et al. 2021; Gundersen and Kjensmo 2018; Baker 2016). In particular, we assess the models' robustness by comparing the stated performance with our reproduced results on the same dataset FI-2010 Ntakaris et al. (2018). We also assess their generalizability by testing their performance on unseen market scenarios using LOBSTER data [13].

Furthermore, we enrich our experiments by including classical ML tree-based models and ensemble methods. In addition, we also provide a profit analysis by conducting a trading simulation. Our code is organized in a modular framework, called LOBCAST, which is openly accessible for users. Our findings reveal that while the best models exhibit robustness, achieving solid F1-Scores on FI-2010, they show poor generalizability, as their performance significantly drops when applied to unseen LOB market data. Our experiments show that most of the attention-based DL models outperform the other approaches. Our results provide insightful evidence of possible weaknesses of the current state-of-the-art in SPTP, which allow us to add a critical discussion about how to improve models' generalizability, data labelling, and representation.

The main contributions of our work are the following:

- We release a highly modular open-source framework called **LOBCAST**,[1] to pre-process data, train, and test stock market models. Our framework employs the latest DL libraries to provide all researchers an easy, performing, and maintainable solution. Furthermore, to support future studies, we release two meta-learning models and a backtesting environment for profit analysis.
- We evaluate existing LOB-based stock market trend predictors, showing that most of them overfit the FI-2010 dataset with remarkably lower performance on unseen stock data.
- In order to guide model selection in real-world applications, we evaluate the sensitivity of the models to the data labeling parameters, compare the performance of both DL and non-DL models, and evaluate and discuss the financial performance of existing models under different market scenarios.
- We discuss the strengths and limitations of existing methodology and identify areas for future research toward more reliable, robust, and reproducible approaches to stock market prediction.

The remainder of this paper is organized as follows: in Sect. 2, we review existing work; in Sect. 3, we introduce the stock trend prediction problem; in Sect. 4 we present all the machine learning models scrutinised in this work and in Sect. 5 we describe the datasets used to train the model for the task; in Sect. 6 we benchmark the analyzed approaches. In Sect. 7, we discuss future directions and conclusions reached by our research. Finally, we devote Appendix 9. to a detailed description of the selected models, whose robustness and generalizability study is further enriched by the additional experimental results reported in Appendix 10..

2 Related work

The increasing interest in DL for price trend prediction motivated several researchers to collect and analyze State-Of-the-Art (SOTA) solutions in benchmark surveys. The study by Jiang (2021) analyzes papers published between 2017 and 2019 that focused on stock price and market index prediction. In their literature review, the authors studied Neural Network (NN) structures and evaluation metrics used in selected papers, as well as implementation and reproducibility. This work was extended by Kumbure et al. (2022), including an in-depth analysis of the data (i.e., market indices and input variables used for stock market predictions). Ozbayoglu et al. (2020) provide a comprehensive overview of SOTA DL and ML algorithms that are commonly used for finance applications. The authors then survey numerous papers tackling some of such applications with DL and ML models, e.g., portfolio management, fraud detection, and risk assessment. With a similar approach, Sezer et al. (2020) summarize the most used DL models for several finance applications, including stock price forecasting. The work by Hu et al. (2021) surveys 86 papers on stock and foreign exchange price prediction. The authors review the datasets, variables, models, and performance metrics used in each surveyed article. In Nti et al. (2020), the authors conduct a systematic and critical review of 122 papers for stock prediction. To evaluate the results of the surveyed papers, the authors also implement three baselines DL and ML algorithms

[1] The code is publicly available at https://github.com/matteoprata/LOBCAST

which are commonly exploited in the reviewed literature: Decision Trees (DTs), Support Vector Machine (SVM) and Artificial Neural Network (ANN). The authors show that ANN achieves the best performance in terms of different error metrics, followed by DTs and SVM.

In contrast to the aforementioned works that have primarily surveyed and reviewed the literature on the SPTP task in general, our focus is specifically on papers addressing this task using LOB data, as will be discussed in Sect. 3. Moreover, our work does not narrow to evaluating the results reported in the surveyed papers and to proposing a validation through classical baselines. Instead, we also evaluate the generalizability of the models by running tests on different datasets.

Several studies include sentiment analysis data for price trend prediction. The works by Shah et al. (2022) and Al-Alawi and Alaali (2023) analyze solutions based on sentiment analysis through Natural Language Processing (NLP) to investigate the impact of social media on the stock market, showing that this combination improves the accuracy of stock prediction models. Similar conclusions are reported in Nguyen et al. (2015), Li et al. (2014).

A different research topic than the one proposed in this paper is the design of recommend systems to select the most profitable stocks. This field of research relies on the observation that some investors might be interested in predicting the top profitable k stocks instead of price trends. Saha et al. (2021) propose a new measure for stock ranking prediction to maximize investors' profit. The work in Alsulmi (2022) explores rank-based ML-based approaches and identifies a feature set that contains various statistics indicating the performance of stock market companies that can be used to train several ranking models. Song et al. use two DL models to design learning-to-rank algorithms to construct equity portfolios based on sentiment news Song et al. (2017).

Rundo et al. (2019) presented a comprehensive overview of traditional and ML-based approaches for stock market prediction and highlighted some limitations of traditional approaches, showing that DL models outperform them in terms of accuracy. Similar findings are reported by Mintarya et al. (2023). Lim and Zohren (2021) discussed recent developments in hybrid DL models, which combine statistical and learning components for both one-step-ahead and multi-horizon time-series forecasting. Similarly, Shah et al. (2019) discussed hybrid approaches in their work on the SOTA algorithms commonly applied to stock market prediction. Additionally, they provided a taxonomy of computational approaches for stock market analysis and prediction. Olorunnimbe and Viktor (2023) explore applications of DL in finance and stock markets, with a particular emphasis on works proposing backtesting meeting the requirements for real-world use. They reviewed various scenarios of DL models in finance, with a focus on trade strategy, price prediction, portfolio management, and others. The authors also underline whether the surveyed papers are reproducible. Nevertheless, the reproducibility is not studied by means of experiments but rather just by checking whether the authors of the collected papers provided an open-source code. Finally, L. Lucchese et al. (2022) study a similar problem focusing on order book-driven predictability by using deep learning techniques. They focus on the order book representation, including a novel representation of volumes, and they evaluate the relation between the price predictability and how far ahead we can actually predict. They meticulously address questions related to data representation, yet they focus their study on only three models. Differently from this work, our study considers a broader range of existing works, providing related implementation and comparative analysis.

Several works exploit technical indicators such as moving average convergence/divergence, momentum analysis, volume on balance, and relative strength index to predict

market trends with ML (see e.g., Fei and Zhou 2023; LAI et al. 2019; Ratto et al. 2018). In our paper, we benchmark papers that propose models for stock price trend prediction relying on LOB data. While data using technical indicators offer explainability and can be effective for market predictions, our work studies models that only include market data at its finest granularity and aims to investigate if DL models can extract useful information from raw data going beyond technical indicators.

Our work is the first to provide a benchmark of recent DL approaches applied to the SPTP task, utilizing LOB data. In contrast to previous work, we re-implement the surveyed papers to evaluate their robustness and generalizability. Furthermore, we have released an open-source framework designed for data preprocessing, model training, and testing. Our framework also incorporates profit analysis capabilities that users can exploit to test their own price trend prediction model.

3 The stock price trend prediction problem

The common ground that unifies the models studied in this paper is the goal of solving the SPTP problem via Deep Neural Networks (DNNs) trained on LOB data. LOB data are particularly enlightening as they provide raw and granular information on stocks' trades. By observing the LOB in a fixed period of time, SPTP models return a distribution over the possible future market movements.

3.1 Limit order book (LOB)

A stock exchange employs a matching engine for storing and matching the orders issued by the trading agents. This is achieved by updating the so-called Limit Order Book (LOB) data structure. Each security (tradable asset) has a LOB, recording all the outstanding bid and ask orders currently available on an exchange or a trading platform. The shape of the order book gives traders a simultaneous view of the market demand and supply.

There are three major types of orders. *Market orders* are executed immediately at the best available price. *Limit orders*, instead, include the specification of a desired target price: a limit sell [buy] order will be executed only when it is matched to a buy [sell] order whose price is greater [lower] than or equal to the target price. Finally, a *cancel order* removes a previously submitted limit order.

Figure 1 depicts an example of a LOB snapshot, characterized by *buy* orders (*bid*) and *sell* orders (*ask*) of different prices. A *level*, shown on the horizontal axis, represents the number of shares with the same price either on the bid or ask side. In the example of Fig. 1, there are three bid and three ask levels. The *best bid* is the price of the shares with the highest price on the buy side; analogously, the *best ask* is the price of the shares with the lowest price on the sell side. When the former exceeds or equals the latter, the corresponding limit ask and bid orders are executed. The LOB is updated with each event (order insertion/modification/cancellation) and can be sampled at regular time intervals.

In Huang and Stoll (1994), Tran et al. (2022), Pascual and Veredas (2003), Cao et al. (2008), Cao et al. (2009), Duong and Kalev (2014) it has been empirically demonstrated, using both linear and non-linear models, that the orders behind the best bid and ask prices have a significant impact to price discovery and contain information about short-term future price movements, supporting the hypothesis that leveraging deeper levels of the limit order book is essential for improving the performance of SPTP tasks. This is the main

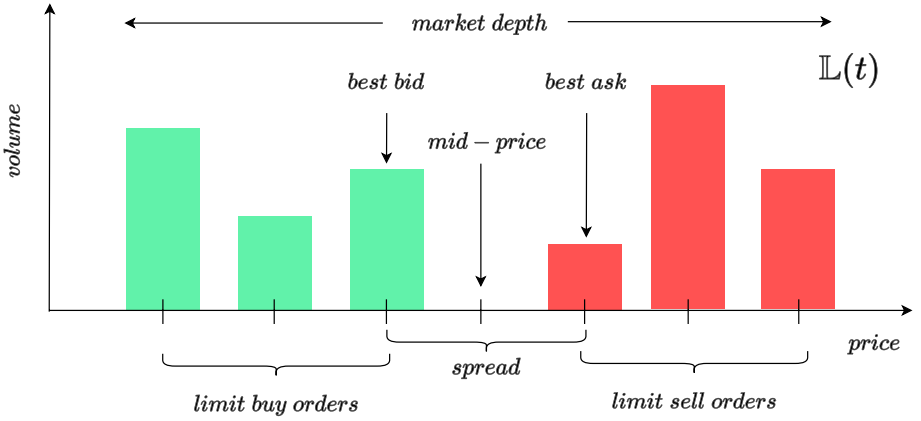


Fig. 1 An example of LOB

reason for not restricting the input to the best levels. Additionally, deep learning models are well suited to handle high-dimensional input.

We represent the evolution of a LOB as a time series $\mathbb{L}$, where each $\mathbb{L}(t) \in \mathbb{R}^{4L}$ is called a LOB record, for $t = 1, \ldots, N$, being N the number of LOB observations and L the number of levels. In particular,

$$\mathbb{L}(t) = \{P^s(t), V^s(t)\}_{s \in \{\texttt{ask},\texttt{bid}\}}$$

where $P^{\texttt{ask}}(t), P^{\texttt{bid}}(t) \in \mathbb{R}^L$ represent the prices of levels 1 to L of the LOB, on the *ask* ($s = \texttt{ask}$) side and *bid* ($s = \texttt{bid}$) side, respectively, at time t. Analogously, $V^{\texttt{ask}}(t), V^{\texttt{bid}}(t) \in \mathbb{R}^L$ represent the volumes. This means that for each t and every $j \in \{1, \ldots, L\}$ on the *ask* side, $V_j^{\texttt{ask}}(t)$ shares can be sold at price $P_j^{\texttt{ask}}(t)$. The *mid-price* $m(t)$ of the stock at time t, is defined as the average value between the best bid and the best ask,

$$m(t) = \frac{P^{\texttt{ask}}(t) + P^{\texttt{bid}}(t)}{2}$$

On average, if most of the executed orders are on the ask [bid] side, the mid-price increases [decreases] accordingly.

3.2 Trend definition

We use a ternary classification for trends: `U` ("upward") if the price trend is increasing; `D` ("downward") for decreasing prices; and `S` ("stable") for prices with negligible variations. Among all the possible single values, mid-prices provide the most reliable indication of the actual stock price for equity markets. Nevertheless, because of the market's inherent fluctuations and shocks, they can exhibit highly volatile trends. For this reason, using a direct comparison of consecutive mid-prices, i.e., $m(t)$ and $m(t + 1)$, for stock price labelling would result in a noisy labelled dataset. As a result, labelling strategies typically employ smoother mid-price functions instead of raw mid-prices. Such functions consider mid-prices over arbitrarily long time intervals, called *horizons*. Our experiments adopt the

$\mathbb{M}(T)$

	side →	*ask*			*bid*		
	volume →	V_1	$\cdots$	V_L	V_1	$\cdots$	V_L
	price →	P_1	$\cdots$	P_L	P_1	$\cdots$	P_L
time	$t-h$	$v_1(t-h)$	$\cdots$	$v_L(t-h)$	$v_1(t-h)$	$\cdots$	$v_L(t-h)$
		$p_1(t-h)$	$\cdots$	$p_1(t-h)$	$p_1(t-h)$	$\cdots$	$p_1(t-h)$
	$t-h+1$	$v_1(t-h+1)$	$\cdots$	$v_L(t-h+1)$	$v_1(t-h+1)$	$\cdots$	$v_L(t-h+1)$
		$p_1(t-h+1)$	$\cdots$	$p_L(t-h+1)$	$p_1(t-h+1)$	$\cdots$	$p_L(t-h+1)$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	t	$v_1(t)$	$\cdots$	$v_L(t)$	$v_1(t)$	$\cdots$	$v_L(t)$
		$p_1(t)$	$\cdots$	$p_L(t)$	$p_1(t)$	$\cdots$	$p_L(t)$

Fig. 2 An example of market observation

labelling proposed in Ntakaris et al. (2018) and repurposed in several other SOTA solutions we selected for benchmarking. The adopted labelling strategy compares the current mid-price to the average mid-prices $a^+(k,t)$ in a future *horizon* of k time units, formally:

$$a^+(k,t) = \frac{1}{k}\sum_{i=1}^{k} m(t+i). \tag{1}$$

The average mid-prices are used to define a static threshold $\theta \in (0,1)$ that is used to identify an interval around the current mid-price and define the class of the trend at time t as follows:

$$\begin{gathered} \mathtt{U} : a^+(k,t) > m(t)(1+\theta),\ \mathtt{D} : a^+(k,t) < m(t)(1-\theta), \\ \mathtt{S} : a^+(k,t) \in [m(t)(1-\theta), m(t)(1+\theta)]. \end{gathered} \tag{2}$$

With this labelling, we beat the effect of mid-price fluctuations by considering their average over a desired horizon k and considering a trend to be stable when the average mid-price variations do not change significantly, thus avoiding over-fitting. We highlight that timestamp t can come either from a homogeneous or an event-based process. In our experiments, we consider the latter approach. Hence, the horizon k is expressed in the number of future events.

3.3 Models I/O

Given the time series of a LOB $\mathbb{L}$ and a temporal window $T = [t-h, t]$, $h \in \mathbb{N}$, we can extract *market observations* on T, $\mathbb{M}(T)$, by considering the sub-sequence of LOB observations starting from time $t-h$ up to t. Fig. 2 gives a representation of a market

observation $\mathbb{M}(T) \in \mathbb{R}^{h \times 4L}$. The market observation over the window $[t-h, t]$ is associated with the label computed through Eqs. 1 and 2 at time t. An SPTP predictor takes as an input a market observation and outputs a probability distribution over the trend classes `U`, `D`, and `S`.

4 Models

We selected and surveyed 13 SOTA models based on DL for the SPTP task using LOB data. Models selection was made based on their prominence in the literature and widespread usage. These models represent a diverse set, covering various DL architectures and methods. These models were published in papers between 2017 and 2022 and are described in detail in Appendix 9.. We also include in our analysis two additional baselines, namely Multilayer Perceptron (MLP) and Long-Short Term Memory (LSTM), which were used as a benchmark in Tsantekidis et al. (2017a) and in Tsantekidis et al. (2020), respectively. All proposed models are based on DNNs and were originally trained and tested on the FI-2010 dataset. We also study two ensemble methods, described in Sect. 4.2. Table 1 summarizes the most peculiar characteristics of the studied models, which we comment on in Sect. 4.1.

4.1 Summary of models

Table 1 summarizes the most peculiar characteristics of the selected models. The *temporal shape* represents the length of the input market observation for the model. In the table, the *features shape* refers to the number of features used by the models to infer the trend in the original papers. In the Table, we also indicate whether the authors released the code, and if so, whether they have used PyTorch Paszke et al. (2019) or TensorFlow Abadi et al. (2016). This is relevant because to ensure consistency and compatibility within our proposed framework, based on PyTorch Lightning, we found it necessary to re-implement models for which the code was not available or was only available in Tensorflow. We made every effort to validate and verify the correctness of our re-implementations, including making our code publicly available, which facilitates collaboration and scrutiny from the research community. We remark that to improve the reproducibility of the results, it would be advisable for the research community to publish the code developed to carry out the experiments.

In High-Frequency Trading (HFT) and algorithmic trading in general, minimizing latency between model querying and order placement is of utmost importance Gomber and Haferkorn (2015). To explore this aspect, we analyzed the inference time in milliseconds of all models, based on the experiments reported in Sect. 6.3. As shown in Table 1, DEEPLOB, DEEPLOBAT, AXIALLOB, TRANSLOB, and ATNBoF had inference times in the order of milliseconds, potentially unsuitable for HFT applications compared to other models with shorter times. Finally, we have reported the number of trainable parameters for each model. A noteworthy observation is that the average number of parameters is very low compared to other classical deep learning fields, such as computer vision He et al. (2016) and natural language processing (Devlin et al. 2018; Brown et al. 2020). This leads us to conjecture that current systems are inadequate in effectively handling the complexity of LOB data, as we will verify in the rest of this paper.

Table 1 Relevant characteristics of the selected models

	Temporal shape (h)	Features shape	Code available	Nr trainable parameters	Inference time (ms)
Tsantekidis et al. (2017a) MLP (2017)	100	40	✗	10^6	0.08
Tsantekidis et al. (2017a) LSTM (2017)	100	40	✗	$1.6 \cdot 10^4$	0.21
Tsantekidis et al. (2017b) CNN1 (2017)	100	40	✗	$3.5 \cdot 10^4$	0.36
Tran et al. (2018) CTABL (2018)	10	40	TensorFlow	$1.1 \cdot 10^4$	0.48
Zhang et al. (2019) DEEPLOB (2019)	100	40	PyTorch	$1.4 \cdot 10^5$	1.31
Passalis et al. (2019) DAIN (2019)	15	144	PyTorch	$2.1 \cdot 10^6$	0.15
Tsantekidis et al. (2020) CNNLSTM (2020)	300	42	✗	$5.3 \cdot 10^4$	0.50
Tsantekidis et al. (2020) CNN2 (2020)	300	40	✗	$2.8 \cdot 10^5$	0.49
Wallbridge (2020) TRANSLOB (2020)	100	40	TensorFlow	$1.1 \cdot 10^5$	2.40
Passalis et al. (2020) TLONBoF (2020)	15	144	PyTorch	$6.5 \cdot 10^5$	0.43
Tran et al. (2021) BINCTABL (2021)	10	40	✗	$1.1 \cdot 10^4$	0.71
Zhang et al. (2021) DEEPLOBATT (2021)	50	40	TensorFlow	$1.8 \cdot 10^5$	1.73
Guo and Chen (2022) DLA (2022)	5	144	✗	$1.2 \cdot 10^5$	0.23
Tran et al. (2022) ATNBoF (2022)	100	40	PyTorch	$1.3 \cdot 10^7$	3.90
Kisiel and Gorse (2022) AXIALLOB (2021)	40	40	✗	$2 \cdot 10^4$	1.91

4.2 Ensemble methods

To explore the possibility of achieving new SOTA performance by combining the predictions of all 15 models, we have implemented two ensemble methods: MAJORITY and METALOB.

The MAJORITY ensemble assigns the class label that appears most frequently among the predictions of the classifiers. To account for variations in the performance of individual classifiers, we incorporate a weighting scheme based on their F1-Scores. This ensures that predictions from higher-performing models carry more influence in the final decision.

The METALOB meta-classifier is implemented as a multilayer perceptron (MLP) with two fully connected layers. It is designed to learn how to effectively combine the outputs of the 15 DL models, which serve as the base classifiers to produce the final output. The input

Table 2 Class balancing on FI-2010

Horizon k	Train Set {U, S, D} (%)	Val Set {U, S, D} (%)	Test Set {U, S, D} (%)
1	20 – 60 – 20	19 – 63 – 18	15 – 71 – 14
2	26 – 49 – 25	24 – 52 – 24	20 – 62 – 18
3	30 – 41 – 29	27 – 46 – 27	23 – 56 – 21
5	35 – 30 – 35	32 – 37 – 31	28 – 47 – 25
10	41 – 18 – 41	37 – 26 – 37	34 – 34 – 32

to the meta-classifier is a 1D tensor with a probability distribution over the trends (*up, stationary, down*) for each of the models, resulting in a tensor of $3 \cdot 15$ elements.

5 Datasets

LOB data are not often publicly available and very expensive: stock exchanges (e.g., NASDAQ) provide fine-grained data only for high fees. The high cost and low availability restrict the application and development of DL algorithms in the research community. In the sections that follow, we will introduce the reader to two datasets that will be used to analyze the performances of the models under the robustness and generalizability point of view, that are FI-2010 and LOB-2021/2022 respectively.

5.1 FI-2010 to test robustness

The most widely spread public LOB dataset is **FI-2010** which is licensed under *Creative Commons Attribution 4.0 International (CC BY 4.0)* and was proposed in 2017 by Ntakaris et al. Ntakaris et al. (2018) with the objective of evaluating the performance of machine learning models on the SPTP task. The dataset consists of LOB data from five Finnish companies: Kesko Oyj, Outokumpu Oyj, Sampo, Rautaruukki, and Wärtsilä Oyj of the NASDAQ Nordic stock market. Data spans the time period between June 1st to June 14th, 2010, corresponding to 10 trading days (trading happens only on business days). About 4 million limit order messages are stored for ten levels of the LOB. The dataset has an event-based granularity, meaning that the time series records are not uniformly spaced in time. LOB observations are sampled at intervals of 10 *events*, resulting in a total of 394,337 events. This dataset has the intrinsic limitation of being already pre-processed (filtered, normalized, and labelled) so that the original LOB cannot be backtracked, thus hampering thorough experimentation. Additionally, the labelling method employed is found to be prone to instability, as demonstrated by Zhang et al. in Zhang et al. (2019).

The dataset provides the time series and the classes relative to five horizons $k \in \mathcal{K} = \{1, 2, 3, 5, 10\}$ by leveraging the trend definitions described in Eq. 2. Such a labelling scheme is very sensitive to the threshold θ regarding the resulting balancing between "upward", "downward" and "stable" trends. Table 2 shows the class balancing for different horizons $k \in \mathcal{K}$. The authors of the dataset employed a single threshold $\theta = 0.002$ for all horizons, but it balances only the case of $k = 5$. As it can be observed, the stationary class S is progressively less predominant in favour of the upward and

downward classes. In our experimental campaign, the class imbalance is not addressed to guarantee a fair robustness evaluation since the considered works do not claim to have done so.

5.2 LOB-2021/2022 to test generalizability

To test the generalizability of the models in a more realistic scenario, we used data extracted from *LOBSTER* [13], an online LOB *data provider* for order book data, which is publicly available for the research community with an annual fee. LOBSTER limit order books are reconstructed directly from NASDAQ traded stocks. To compare the performance of the algorithms in a wide range of scenarios, we have created a large LOB dataset, including several stocks and time periods. The chosen pool of stocks includes those from the top 50% more liquid stocks of NASDAQ. To construct a diversified evaluation scenario, we selected six stocks, namely: SoFi Technologies (SOFI), Netflix (NFLX), Cisco Systems (CSCO), Wing Stop (WING), Shoals Technologies Group (SHLS), and Landstar System (LSTR). The periods in consideration are *July 2021* (2021-07-01 to 2021-07-15, 10 trading days) making up **LOB-2021**, and *February 2022* (2022-02-01 to 2022-02-15, 10 trading days) making up **LOB-2022**. The selection of these two periods aimed to capture data from periods with different levels of market volatility. February 2022 exhibited higher volatility compared to July 2021, largely influenced by the Ukrainian crisis. This allows for an assessment of models across varying market conditions. Further details on stock selection and processing are provided in the following two paragraphs.

5.2.1 Stocks selection

To make up *LOB-2021* and *LOB-2022* and consider variegated evaluation scenarios, we curated a pool of 630 stocks from NASDAQ exchange with a market capitalization that ranged from $\sim$ 2 Billion to $\sim$ 3 Trillion dollars. Data was gathered from NASDAQ Stock Screener (Nasdaq, https://www.nasdaq.com/market-activity/stocks/screener). From the pool of stocks, we generated 6 clusters with *t-distributed Stochastic Neighbor Embedding* (*t*-SNE) to capture stock differences in the years 2021-2023. We used the following features: daily return, hourly return, volatility, outstanding shares, P/E ratio, and market capitalization. The P/E ratio indicates the ratio between the price of a stock (P) and the company's annual earnings per share (E). The analysis led to the identification of 6 stocks that are nearest to the cluster centroids of the generated 3-dimensional latent space. The stocks make up the set $\mathcal{S}$ = {SOFI, NFLX, CSCO, WING, SHLS, LSTR}. Table 3 captures the main features of these stocks for the period of July 2021. The selected stocks have very variable average daily returns, the minimum being SHLS and the maximum being NFLX. Daily and Hourly returns highlight that some stocks are more volatile than others. The market capitalization represents the total value of the outstanding common shares stockholders own. Stocks show different class balancing in the training set. CSCO is the stock with a major imbalance toward the stable class, whereas NFLX and LSTR are more unbalanced towards the up and down classes, respectively. In Sect. 6, we analyze the reasons behind the occurrence of class imbalance specific to individual stocks and discuss its impact. The mid-price movements for these two periods and the selected stocks are depicted in Fig. 3.

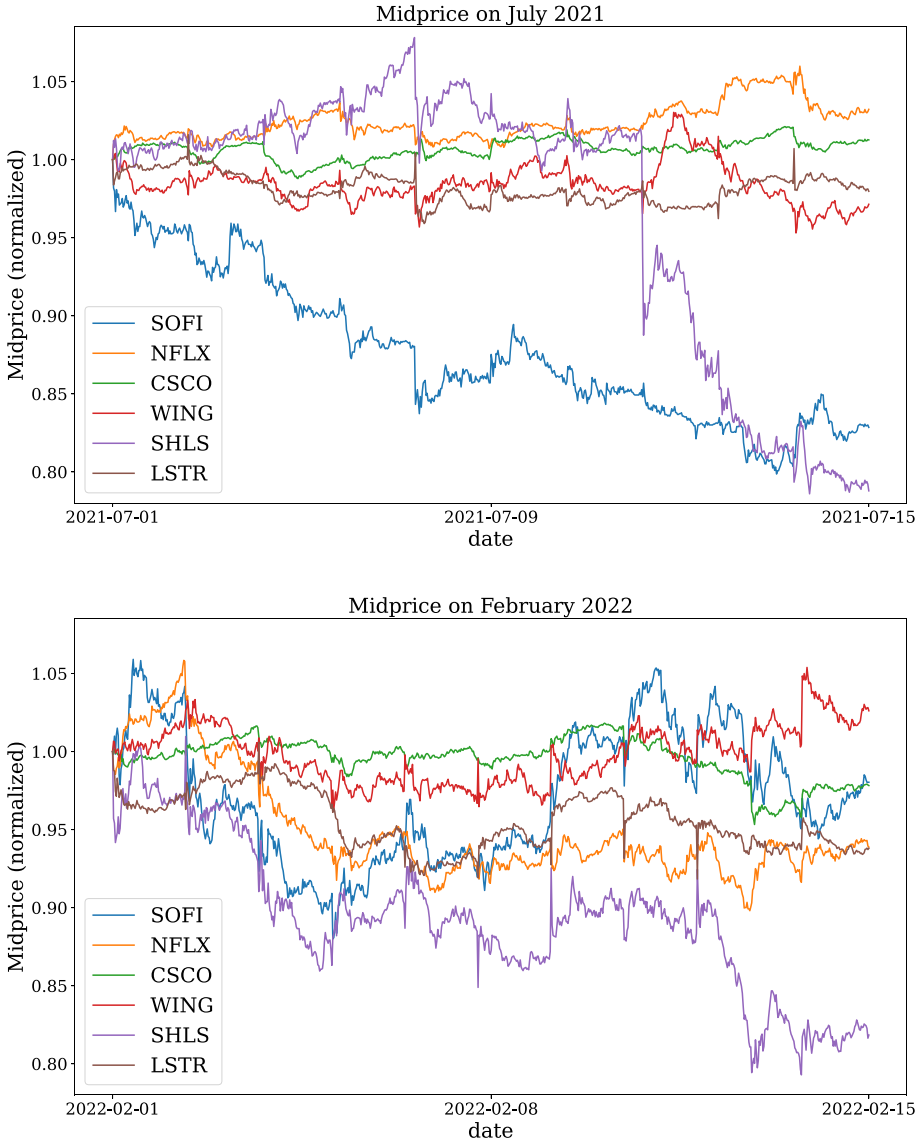


Fig. 3 Stocks returns from day 0 for LOB-2021/2022

5.2.2 Stock processing

We build *LOB-2021* and *LOB-2022* resembling the structure of the FI-2010 dataset, described in the previous section and proposed in Ntakaris et al. (2018). In particular, to generate the LOB-2021/2022 datasets, we utilize data from the LOBSTER data provider, which consists of LOB records (i.e., $\mathbb{L}(t)$ vectors) resulting from events caused by traders at the exchange. LOBSTER associates these records with the specific events that caused changes in the LOB. We isolated the following types of events: order

Table 3 LOB-2021 stocks main features

Stock	Daily Return (%)	Hourly Return (%)	Market Cap.	P/E Ratio	Train Set { `U`, `S`, `D` } (%), $k = 5$	Train Set (%), $k = 5$
SOFI	-2.3 ± 3.1	-0.3 ± 1.2	$4.26 \cdot 10^9$	−27.84	41 – 19 – 40	14.8
NFLX	0.6 ± 1.7	0.05 ± 0.6	$1.58 \cdot 10^{11}$	38.28	45 – 5 – 50	21.7
CSCO	0.2 ± 0.7	0.02 ± 0.4	$2 \cdot 10^{11}$	17.59	18 – 65 – 17	46.2
WING	-0.3 ± 3.2	-0.04 ± 0.9	$6.06 \cdot 10^9$	96.87	44 – 7 – 49	6.1
SHLS	-2.4 ± 4.9	-0.3 ± 1.9	$4.05 \cdot 10^9$	26.24	42 – 14 – 44	7.4
LSTR	0.1 ± 2.8	-0.03 ± 0.73	$6.16 \cdot 10^9$	16.55	48 – 5 – 47	3.8

submissions, *deletions*, and *executions*, which account for almost all the events in the markets.

For each stock in the set $\mathcal{S}$ we construct a *stock time series* of LOB records $\mathbb{L}_s(t) \in \mathbb{R}^{4L}$, with $L = 10, s \in \mathcal{S}$, N_s being the amount of records of the stock s in the considered temporal interval (e.g., (2021-07-01, 2021-07-15) for LOB-2021), $t \in [1, N_s]$. We recall that the $4 \cdot 10$ features represent the prices and volumes on the buy and sell sides for the ten levels of the LOB. We highlight that the time series $\mathbb{L}_s$ are non-uniform in time since LOB events can occur at irregular intervals driven by traders' actions. We do not impose temporal uniformization. Instead, we sample the market observation every ten events, similarly to the stock processing performed for FI-2010 dataset. Furthermore, we do not account for liquidity beyond the 10th order level in the LOB. This approximation is necessary to ensure computational tractability while retaining the most influential levels. It is a commonly employed technique in stock market prediction models, also employed in FI-2010.

Each stock time series $\mathbb{L}_s$ is split into *training*, *validation*, and *testing* sets using a 6-2-2 days split. Normalization is performed on stock time series using a z-score approach, separately normalizing the prices and volumes. The mean and standard deviation are calculated from the union of the training and validation splits for all stock time series. These statistics are then used to normalize the entire dataset, including the test splits. The final dataset is constructed by vertical stacking (i.e., concatenating along the rows) the six training splits (i.e., one for each stock), six validation splits, and six test splits in this order.

The dataset is used to extract market observations with a sliding window approach, as explained in Sect. 3. The training set is randomly permuted according to the standard procedure adopted by the SOTA papers in this field.

Labelling market observations is accomplished by leveraging the trend definitions described in Eq. 2, mapping market observations to the corresponding trend based on a predefined prediction horizon $k \in \mathcal{K}$. It is important to note that a new dataset is generated for each prediction horizon $\mathcal{K}$. Consequently, LOB-2021 and LOB-2022 consist of five (i.e., $|\mathcal{K}|$) distinct datasets, each corresponding to one of the five prediction horizons. As for FI-2010 dataset, we chose the labelling threshold θ of Eq. 2 such that the resulting dataset is balanced for $k = 5$. In Table 4 the class balancing of LOB-2021/2022 datasets for each of the horizons and each split is reported. For a fair comparison, the reported balancing is similar with respect to FI-2010 dataset shown in Table 2.

Table 4 Class balancing on LOB-2021 and LOB-2022

Horizon k	LOB-2021			**LOB-2022**		
	Train Set {U, S, D} (%)	Val Set {U, S, D} (%)	Test Set {U, S, D} (%)	Train Set {U, S, D} (%)	Val Set {U, S, D} (%)	Test Set textbf{U, S, D} (%)
1	18 – 63 – 19	19 – 62 – 19	21 – 59 – 20	20 – 60 – 20	18 – 64 – 18	18 – 63 – 19
2	25 – 50 – 25	25 – 50 – 25	27 – 46 – 27	26 – 47 – 27	24 – 51 – 25	25 – 50 – 25
3	28 – 43 – 29	28 – 43 – 29	30 – 40 – 30	30 – 40 – 30	28 – 43 – 29	29 – 42 – 29
5	32 – 35 – 33	32 – 35 – 33	34 – 33 – 33	34 – 31 – 35	33 – 34 – 33	34 – 32 – 34
10	37 – 25 – 38	37 – 25 – 38	38 – 24 – 38	41 – 28 – 41	40 – 20 – 40	41 – 18 – 41

5.3 Data distribution shift

LOB time series are susceptible to distribution shift due to the dynamic nature of the stock markets. This implies that when used in production, the input data seen by the model may deviate from the data it was trained on Bennett and Clarkson (2022). Such shifts can lead to serious consequences that directly translate into misclassification and significant economic losses if the model is not effectively monitored. It is important to note that the primary focus of this work, and the considered studies, is on the trend classification task. Deploying these models into production demands additional effort, particularly in addressing challenges like distribution shift. Being able to observe the market in time, allows for new data collection facilitating continual model retraining, data weighting for past forgetting, and fine tuning. These methods should be adopted in conjugation with the usual techniques to prevent data overfitting, such as early stopping, data augmentation, and regularization.

6 Experiments

We conducted an extensive evaluation to assess the *robustness* and *generalizability* of 15 DL models to solve the SPTP task, as presented in Sect. 3. Among these, 13 were SOTA models, and 2 DL baseline models commonly used in the literature. More details on the models are given in Sect. 4 and in Appendix 9..

In line with many other studies, we adopt the definition of robustness and generalizability introduced by J. Pineau et al. in their work Pineau et al. (2021). Robustness is the ability of a model to replicate its performance when tested on the same data but under different analyses, such as reimplementation of the code, testing of the code on a different computer architecture, and other contextual changes. Generalizability is the ability of a model to replicate its performance when tested on different data and different analytical tools. Robustness is evaluated by testing the proposed models on **FI-2010**, the benchmark dataset employed in all surveyed papers. To evaluate the generalizability, we use **LOB-2021** and **LOB-2022**, retrieved from LOBSTER data provider Berlin xxxx. In some cases, the authors of the considered works have not provided crucial information, such as the code or the hyperparameters of their models, making reimplementation and hyperparameter search

necessary. Our experiments were carried out using **LOBCAST** (LOBCAST xxxx), the open-source framework we developed and made available online. The framework allows the definition of new price trend predictors based on LOB data.

In Sect. 6.1 we describe our framework and its potential applications. In Sect. 6.2 we introduce a complete description of the hyperparameters search. In Sect. 6.3 we discuss the results deriving from the robustness (Sect. 6.3.1) and generalizability (Sect. 6.3.2) studies, and a focus to the performance of ensemble methods (Sect. 6.3.3). In Sect. 6.4 we expand on the performance achieved for the SPTP task when adopting varying labeling strategies (Sect. 6.4.1) and using non-deep models (Sect. 6.4.2). We conclude with a profitability study in Sect. 6.4.3.

6.1 LOBCAST framework for SPTP

We present **LOBCAST** [60], a Python-based framework developed for stock market trend forecasting using LOB data. LOBCAST is an open-source framework that enables users to test DL models for the SPTP task. LOBCAST contains the implementation of the 15 DL models that were used in the experiments. We believe that LOBCAST, along with the advancements in DL models and the utilization of LOB data, has the potential to improve the state of the art on price trend forecasting in the financial domain.

6.1.1 Applications and features

The core application of LOBCAST is *to provide a standardized benchmarking* of DL-based models for the SPTP task. There are several choices to make while implementing a DL model for SPTP. We collected these choices in LOBCAST to ease the development and evaluation of new models. This not only offers a methodological and standardised approach to addressing the problem but also simplifies the comparison between these choices.

LOBCAST features include: (1) LOB data pre-processing utilities dealing with normalization, splitting, and labelling. (2) A training environment for DL models implemented in PyTorch Lightning Paszke et al. (2019). (3) Integrated interfaces with the popular hyperparameter tuning framework WANDB Biewald (2020), which allows users to tune and optimize model performance efficiently. (4) Generation of detailed reports for the trained models, including performance metrics regarding the learning task (F1-Score, Accuracy, Recall, etc.). (5) Generation of reports measuring the complexity of the models in terms of number of parameters, inference and training time. (6) Support for backtesting for profit analysis, utilizing the Backtesting.py[2] library. (7) The PyTorch implementation of the SOTA models used in the experiments.

All these features are implemented through a modular and scalable framework that can be easily expanded with new models and components.

6.2 Hyperparameters search

For evaluating the *robustness* of the surveyed models, we used the hyperparameters reported in the original papers whenever they were available. However, we encountered cases where hyperparameters were not declared at all, such as in LSTM Tsantekidis et al.

[2] https://kernc.github.io/backtesting.py/

(2017a) and CNN1 Tsantekidis et al. (2017b), while in other cases, including CNNLSTM Tsantekidis et al. (2020), AXIALLOB Kisiel and Gorse (2022), ATNBOF Tran et al. (2022) and DAIN Passalis et al. (2019) only partial information was provided. To address these gaps, we performed a grid search exploring different values for the **batch size**, including $\{16, 32, 64, 128, 256\}$ and the **learning rate**, including $\{0.01, 0.001, 0.0001, 0.00001\}$.

Regarding the *generalizability* experiment, we found that the majority of models using the hyperparameters from the robustness experiment performed poorly on the LOB-2021/2022 datasets. So, we conducted a comprehensive hyperparameter search on horizon $k = 5$ (which is the most balanced) using a grid search approach for all 16 models. For this search, we maintained the same number of epochs and optimizer used in the robustness analysis while searching for batch size and learning rate using the same domains mentioned above. F1-Score maximization over the validation set was the chosen criterion for optimizing the hyperparameters. For a complete overview of the hyperparameters utilized in our experiments, refer to Table 5.

6.3 Performance, robustness and generalizability

To test robustness and generalizability, we conducted our experiments for each model using five different seeds to mitigate the impact of random initialization of network weights and training dataset shuffling. The necessity to produce reliable results within reasonable time constraints led us to opt for a set of five seeds. The training process involved training the 15 models for each seed on each of the considered prediction horizons ($\mathcal{K} = \{1, 2, 3, 5, 10\}$). On average, the training process for all the models took approximately 155 h for FI-2010 and 258 h for each LOB dataset, utilizing a cluster comprised of 8 GPUs (1 NVIDIA GeForce RTX 2060, 2 NVIDIA GeForce RTX 3070, and 5 NVIDIA Quadro RTX 6000).

In Table 6, we summarize the results of our experiments. Our choice of F1-Score as the evaluation metric was motivated by the following reasons: (1) it captures both precision and recall in a single value, (2) the datasets are not well balanced and F1-Score is robust to the class imbalance problem that affects the accuracy measure, (3) it is the only metric that is reported in every SOTA paper. The Table compares the claimed performance of each system (column F1 Claim) with those measured in the robustness (FI-2010) and generalizability (LOB-2021 and 2022) experiments. For each dataset, we show the average performance and the standard deviation achieved by each model in all the horizons, along with its rank.

To evaluate the robustness and the generalizability of the models, we compute the **robustness** and the **generalizability scores**, a value ≤ 100 that is computed as $100 - (|A| + S)$, where A and S are defined as follows. A is the average difference between the F1-Score reported in the original paper and the one that we observed in our experiments on FI-2010 for robustness, and on LOB-2021 and LOB-2022 for generalizability. S is the standard deviation of these differences. The score penalizes models that demonstrate higher variability in their performance by subtracting the standard deviation. The average and standard deviation were computed over the declared horizons for each model and considering all five seeds.

Table 6 clearly highlights the following:

1. Except for a few systems, there is a considerable difference between the claimed performances and those measured in both robustness and generalizability experiments. Note that while the performance gap is negative on average and considerably negative

Table 5 Hyperparameters adopted in our experiments

Model	FI-2010 (Robustness)					LOB-2021/2022 (Generalizability)				
	Learning Rate	Optimizer	Batch Size	Epochs	Dropout	Learning Rate	Optimizer	Batch Size	Epochs	Dropout
LSTM	0.001	Adam	32	100	–	0.0001	Adam	64	100	–
MLP	0.001	Adam	64	100	–	0.00001	Adam	64	100	–
CNN1	0.0001	Adam	64	100	–	0.0001	Adam	32	100	–
CTABL	0.01	Adam	256	200	–	0.001	Adam	64	200	–
DAIN	0.0001	RMSprop	32	100	0.5	0.0001	RMSprop	64	100	0.5
DEEPLOB	0.01	Adam	32	100	–	0.01	Adam	32	100	–
CNNLSTM	0.001	RMSprop	32	20	0.1	0.001	RMSprop	128	100	0.1
CNN2	0.001	RMSprop	32	100	–	0.001	RMSprop	128	100	–
TRANSLOB	0.0001	Adam	32	150	–	0.001	Adam	128	100	–
TLONBoF	0.0001	Adam	128	100	–	0.00001	Adam	32	100	–
BINCTABL	0.001	Adam	128	200	–	0.001	Adam	32	200	–
DEEPLOBATT	0.001	Adam	32	100	–	0.0001	Adam	128	100	–
AXIALLOB	0.01	SGD	64	50	–	0.01	SGD	64	50	–
ATNBoF	0.001	Adam	128	80	0.2	0.00001	Adam	32	80	0.2
DLA	0.01	Adam	256	100	–	0.001	Adam	64	100	–
METALOB	0.0001	SGD	64	100	–	0.0001	SGD	64	100	–

Table 6 Robustness, generalizability, and performance scores of the models. Arrows indicate whether the measured F1-Score of a system is higher or lower than stated in the original paper. Colour saturation highlights systems with best (green) and worst (red) robustness and generalizability scores

	FI-2010				LOB-2021			LOB-2022		
Model	F1 Claim	F1 LOBCAST	F1 Rank	Rob. Score (%)	F1 LOBCAST	F1 Rank	General. Score (%)	F1 LOBCAST	F1 Rank	General. Score (%)
MLP	51.8 ± 3.2	↓ 48.0 ± 2.6	14	91.8	↑ 55.5 ± 3.9	14	95.0	↑ 53.1 ± 2.5	13	96.6
LSTM	63.4 ± 2.1	↓ 63.4 ± 3.6	7	95.5	↓ 56.9 ± 4.1	11	85.9	↓ 56.1 ± 2.8	9	88.8
CNN1	57.9 ± 1.9	↑ 58.1 ± 13.1	10	80.9	↓ 57.5 ± 3.0	8	97.0	↓ 57.1 ± 2.7	6	99.3
CTABL	74.3 ± 5.2	↓ 69.6 ± 4.3	5	91.3	↓ 59.7 ± 2.7	3	78.4	↓ 58.1 ± 3.3	5	78.4
DEEPLOB	78.9 ± 4.4	↓ 71.4 ± 5.3	4	87.6	↓ 59.5 ± 3.0	4	73.7	↓ **59.5 ± 2.9**	**1**	74.7
DAIN	66.8 ± 1.5	↓ 55.6 ± 5.9	11	81.4	↓ 55.9 ± 4.4	12	79.5	↓ 54.1 ± 2.1	12	83.9
CNNLSTM	47.0 ± 0.0	↑ 63.2 ± 8.4	8	75.7	↑ 57.0 ± 3.3	10	87.8	↑ 56.8 ± 2.5	7	90.3
CNN2	45.0 ± 0.8	↑ 50.5 ± 17.3	12	70.5	↑ 55.5 ± 3.5	13	86.6	↑ 55.8 ± 3.2	10	88.6
TRANSLOB	**87.3 ± 4.0**	↓ 59.4 ± 2.6	9	69.9	↓ 57.7 ± 2.9	7	64.2	↓ 50.4 ± 6.1	14	56.4
TLONBoF	53.0 ± 0.0	↓ 49.7 ± 10.5	13	81.5	↑ 57.3 ± 2.9	9	**99.1**	↑ 54.2 ± 3.1	11	**99.9**
BINCTABL	80.1 ± 6.9	↑ **82.6±7.0**	**1**	**99.7**	↓ **61.2 ± 2.7**	**1**	73.5	↓ 59.2 ± 3.3	2	72.3
DEEPLOBATT	78.8 ± 3.1	↓ 67.3 ± 9.0	6	81.2	↓ 60.1 ± 3.0	2	75.7	↓ 58.9 ± 2.8	3	74.5
DLA	78.7 ± 0.7	↓ 73.4 ± 12.1	2	93.2	↓ 57.7 ± 3.7	6	74.9	↓ 56.6 ± 2.4	8	76.9
ATNBoF	67.1 ± 5.5	↓ 40.9 ± 7.7	15	66.1	↓ 53.1 ± 3.7	15	80.9	↓ 48.0 ± 6.9	15	81.2
AXIALLOB	82.0 ± 3.7	↓ 73.4 ± 5.7	3	88.2	↓ 59.5 ± 3.3	5	71.3	↓ 58.6 ± 2.6	4	70.7
METALOB	–	82.2 ± 7.3	–	–	55.9 ± 2.6	–	–	53.2 ± 1.5	–	–
MAJORITY	–	60.0 ± 12.7	–	–	55.5 ± 2.3	–	–	47.9 ± 2.0	–	–

in the scenario of LOB-2021 and 2022, a few systems outperform the claimed results, as highlighted by the arrows in Table 6.

2. All models are very sensitive to hyperparameters; in fact, for about half of the runs, they diverged (F1-Score $\leqslant 33\%$) during the hyperparameters search.
3. The ranking of the systems changes considerably if we compare the declared performances with those measured in our experiments. On the other hand, the best six systems in FI-2010 remain the same in LOB-2021 and 2022.
4. The best-ranked systems do not consistently hold the lead in terms of robustness and generalizability - except for BINCTABL. On the contrary, some of them obtained poor generalizability scores, suggesting that they overfitted the FI-2010 dataset.
5. Five of the best six models incorporate attention mechanisms. In particular, the best-performing model is BINCTABL, which enhances the original CTABL model by adding an Adaptive Bilinear Normalization layer, enabling joint normalization of the input time series along both temporal and feature dimensions. On average, BINCTABL improves the F1-Score by up to 9.2% compared to DLA, i.e., the second-best model, and up to 13% compared to CTABL.
6. Regrettably, ensemble models (the last two rows in Table 6) do not exceed the performance of the top-performing models, which is probably due to the relatively high agreement rate among systems, as shown in Fig. 13 and Fig. 14 in Sect. 10. of the supplementary material.

6.3.1 Robustness on FI-2010

As far as the robustness experiments are concerned, it is important to note that some models discussed in the literature incorporate additional market observation features for predictions. This is the case for the models DAIN, CNNLSTM, TLOBOF, and DLA. To ensure a fair comparison among the models, we included them in our study but reduced their feature set to only the 40 raw LOB features. Due to the presence of these additional features, a strict robustness study could not be conducted for these models. However, the reduction

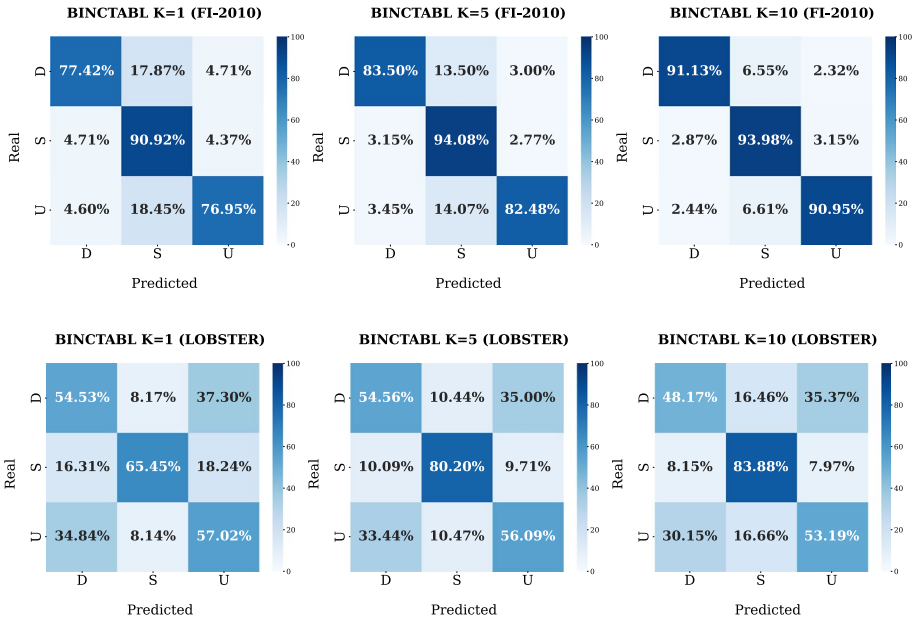


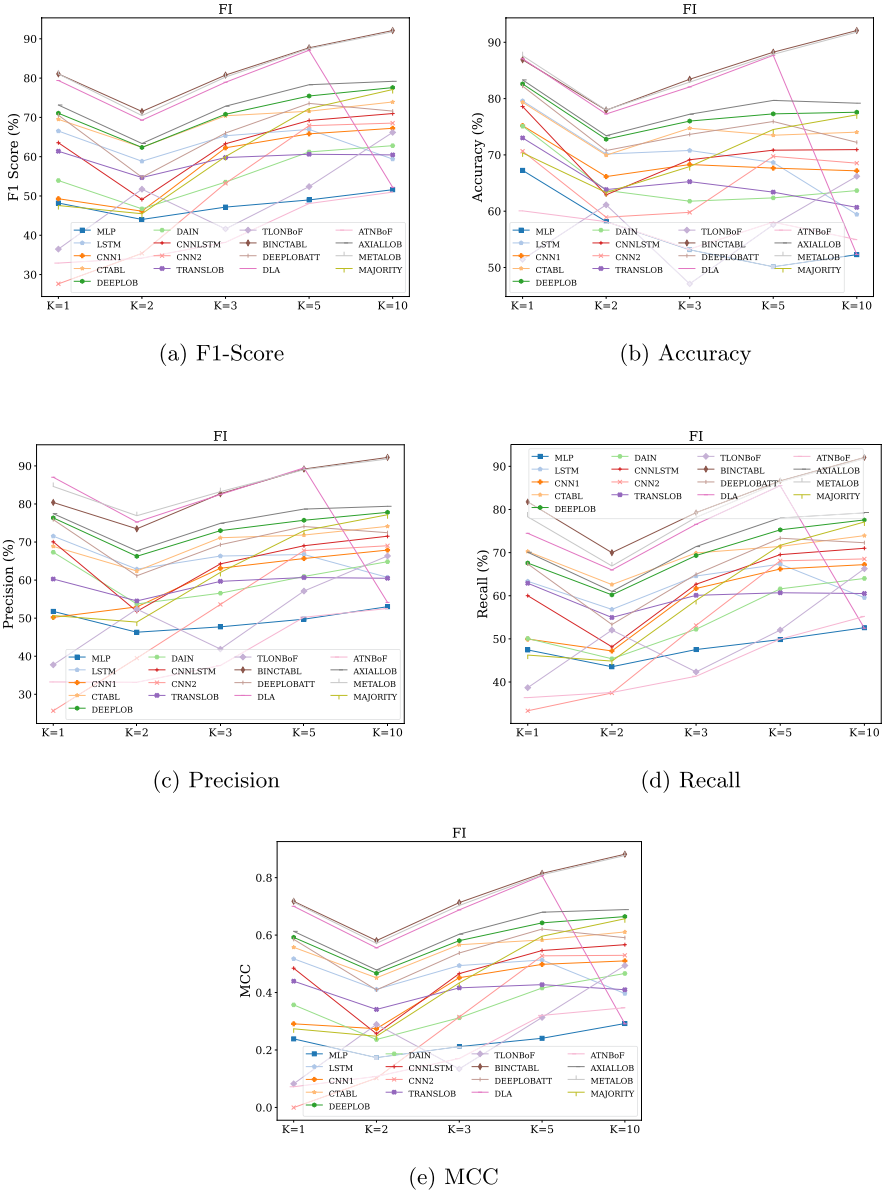
Fig. 4 Confusion matrices for BINCTABL ($k = 1, 5, 10$) on FI-2010 and LOB-2021 datasets

of features did not necessarily cause a deterioration in performance: of particular interest is the case of CNNLSTM, for which the authors used stationary features derived from the LOB, stating that they were better than the raw ones. Impressively, CNNLSTM achieves the greatest average improvement of 20.9% among all the models, proving that, for this model, the raw LOB features are better suited to forecast the mid-price movement than the features proposed by the original authors.

Based on these experiments (summarized in Table 6), the BINCTABL model demonstrates the **highest F1-Score** when averaged over the seeds and prediction horizons, achieving an average of $82.6\% \pm 7.0$. Notably, the BINCTABL model also exhibits the strongest robustness score of 99.7. For a more comprehensive analysis, Fig. 4 provides the confusion matrices of the BINCTABL model's predictions for three horizons ($k = 1$, $k = 5$, and $k = 10$). The confusion matrices demonstrate that the model is slightly biased toward the stationary class. This pattern is consistent across all the models, especially for the first three horizons, reflecting the imbalance of the dataset towards the stationary class, as specified in Sect. 5.

Remarkably, a significant number of models in our study failed to achieve the claimed performance levels. Two possible reasons are the lack of the original code and the missing hyperparameters declaration. Among the models, TRANSLOB and ATNBOF exhibit the largest discrepancies, ranking as the second and first worst performers, respectively. Notably, ATNBoF performs the poorest among all models, both in terms of robustness score and F1-Score.

We observed that CNN1, CNN2, CNNLSTM, TLONBOF, and DLA are the most sensitive models in terms of network weight initialization and dataset shuffling; in fact, these models exhibit a standard deviation over the runs that exceeds 5 basis points, indicating a high degree of variability in their performance. Finally, we highlight that none of the top three models in our study utilize temporal shape $h = 100$ for market observations as input,



(a) F1-Score (b) Accuracy

(c) Precision (d) Recall

(e) MCC

Fig. 5 Evaluation metrics on different horizons $\mathcal{K}$ on FI-2010 dataset

despite it being a common practice in the literature (Zhang et al. 2019; Tsantekidis et al. 2017a, b; Wallbridge 2020; Tran et al. 2022), meaning that they are able to achieve good results without relying on a large historical context. This suggests that the most influential and relevant dynamics impacting their predictions tend to occur within a short time frame.

Figure 5 depicts the F1-Score, Accuracy, Precision, Recall, and MCC of the surveyed models obtained through LOBCAST, for the time horizons $\mathcal{K} = \{1, 2, 3, 5, 10\}$.

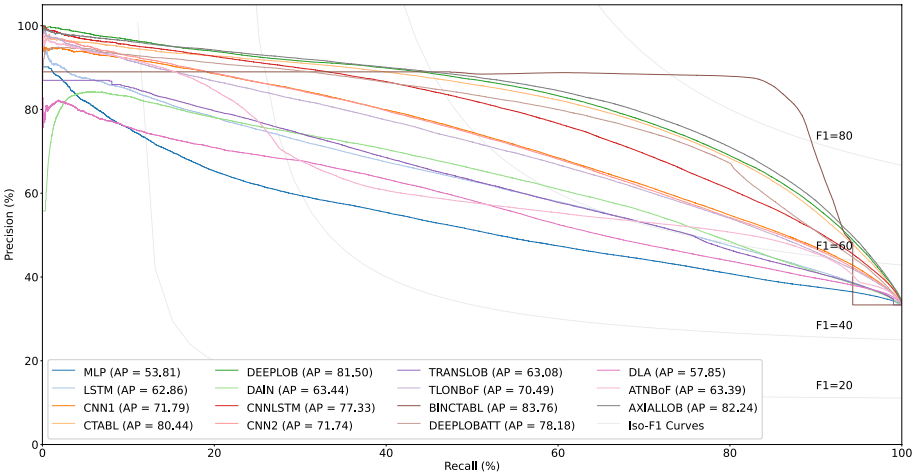


Fig. 6 PR-Curve on FI-2010 for time horizon $k = 10$

Most of the models show similar behaviour with respect to the prediction horizons. Specifically, concerning the F1-Score, the lowest performance is noted when $k = 2$. However, an improvement is evident as the value of k increases, indicating a longer prediction horizon. This may appear counterintuitive, as higher values of k imply forecasting the price trend further into the future. However, for very short horizons, the labelling system adopted may be susceptible to noise, affecting the model's capability to extract relevant patterns. This hypothesis is supported by an experiment in Zhang et al. (2019), in which the authors tried a smoother labelling method and reported a significant decline in the performance of their deep learning model as the prediction horizons increased.

Fig. 6 shows the PR-curves of all models for the time horizon $k = 10$, where classes {`U`, `S`, `D`} are distributed as $\{37\%, 25\%, 38\%\}$. Since the models perform a ternary classification, each curve represents the micro-average precision-recall value and is generated by setting different thresholds for the classification. Thresholds play a role in defining the number of false negatives and false positives, affecting the resulting values of the Precision and the Recall. The best models are the ones with the largest area under the curve, as they are able to make the most accurate predictions (high Precision) while minimizing the false negative rate (high Recall). The figure also shows the iso-F1 Curves on the PR plane. The best-performing model is BINCTABL, with an area under the curve of 8680.

To further compare the performance of the models, we also conducted a T-test for each couple of models reporting the p-values in Table 10 (Appendix 10.). The sample of scores for each model is made up of the F1-Score varying the random seed. We state the null hypothesis h_0 as: *there is no statistical difference between the average performance of the two models*. We highlight in bold the values exceeding a threshold $\alpha = 0.05$, i.e. the couple of models for which h_0 is accepted, that is, the difference in performance has statistical significance.

In Appendix 10.1., we show a different representation of the models' performances at varying prediction horizons and the agreement matrix of their predictions.

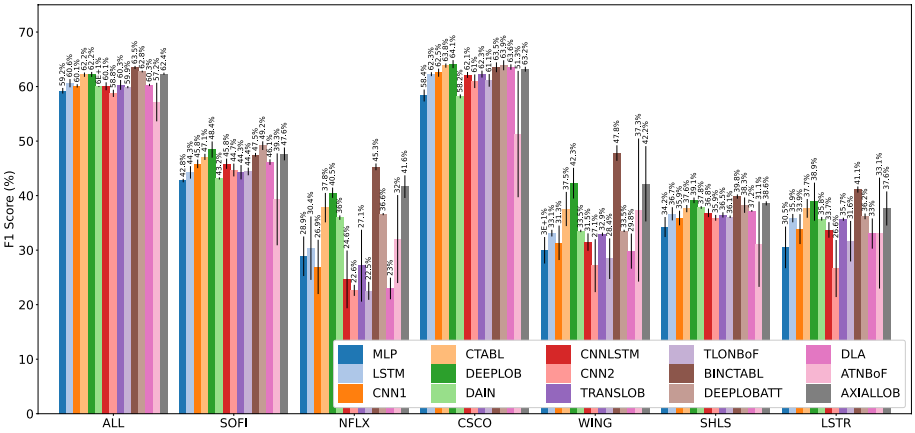


Fig. 7 F1-Score per stock, time horizon $k = 5$, on LOB-2021

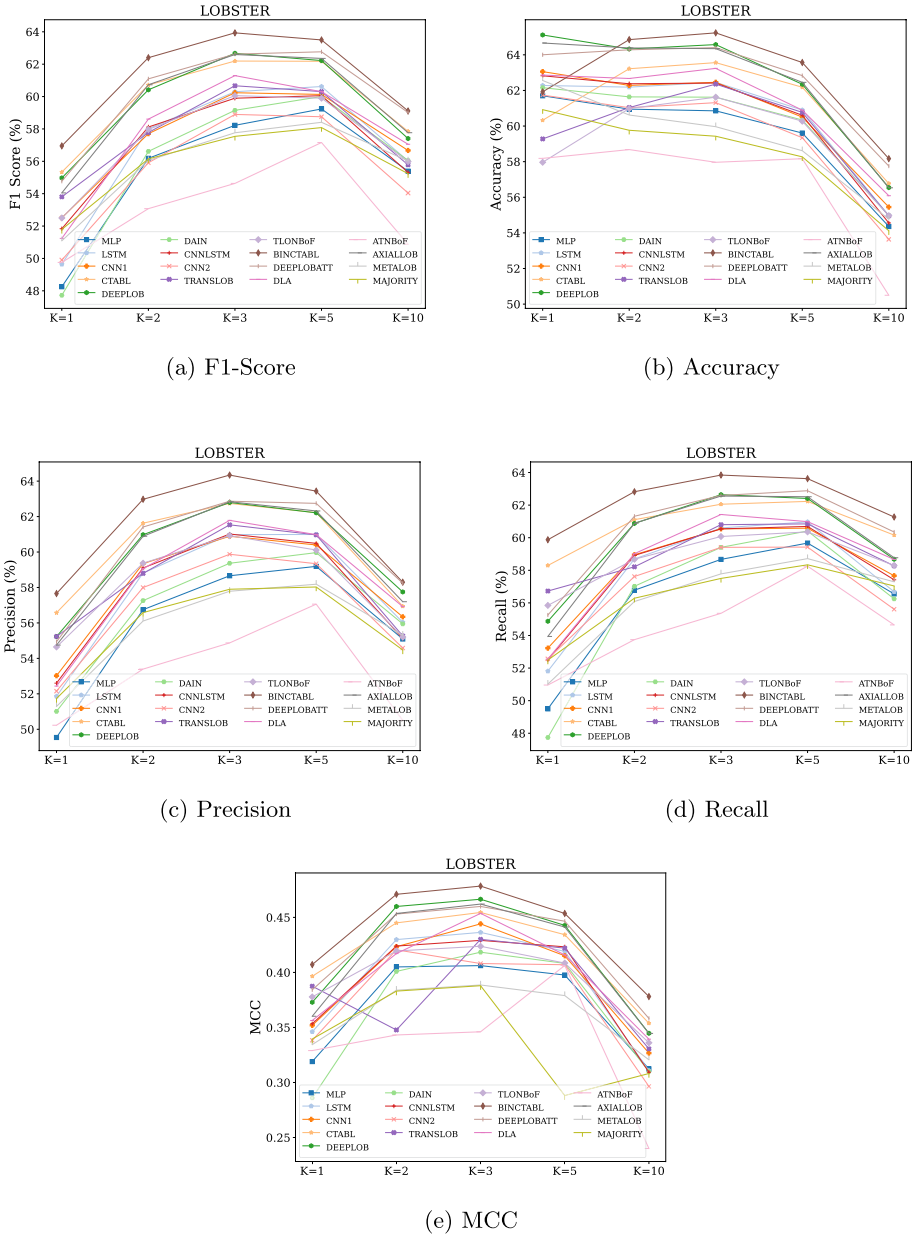
6.3.2 Generalizability on LOB-2021/2022

When comparing the performance of models on the FI-2010 and LOB-2021/2022 datasets, we observe that models showing high performance on the FI-2010 dataset demonstrate a deterioration in performance. Conversely, some of the models that performed poorly on the FI-2010 dataset show an improvement in performance on the LOB-2021/2022 datasets. However, the overall performance of all models on the LOB-2021/2022 dataset is still significantly lower than on the FI-2010 dataset, ranging 48-61% in F1-Score. Furthermore, we conjecture that the overall performance is worse in LOB-2022 than in LOB-2021 due to the higher stocks' volatility. We mention two potential factors contributing to this observed phenomenon. Firstly, the LOB-2021/2022 datasets present a higher level of complexity than the FI-2010 dataset despite having been generated with a similar approach. Indeed, NASDAQ is a more efficient and liquid market than the Finnish one, as evidenced by the fact that LOB-2021/2022 datasets have approximately three times the size of FI-2010 in terms of events for the same period length. Secondly, the best-performing models may overfit the FI-2010 dataset, leading to a decrease in their performance when applied to LOB-2021/2022 datasets. In particular, BINCTABL experiences an average decrease of approximately 19.6% in F1-Score across all horizons, resulting in a generalizability score of 73.5%.

In Fig. 7, we present the results of our tests for the time horizon $k = 5$ on each individual stock from LOB-2021 dataset. Among the tested models, CSCO stands out as yielding the highest performance. This may be attributed to the high stationarity of CSCO (balance 18-65-17% in the train set, see Table 3), indicating more stable and predictable behaviour.

This hypothesis is supported by the confusion matrices in Fig. 4, which consistently show the best performance in the stationary class across all models; we reported only those of BINCTABL since all other models show similar patterns.

We highlight that extracting the per-stock information on the FI-2010 dataset was impossible because it was already assembled, and the authors did not provide information on that procedure. We also show the performance of the models on LOB-2021, including F1-Score, Accuracy, Precision, Recall, and MCC which are displayed in Fig. 8.



(a) F1-Score

(b) Accuracy

(c) Precision

(d) Recall

(e) MCC

Fig. 8 Evaluation metrics on different horizons $\mathcal{K}$ on LOB-2021

Fig. 9 shows the PR-Curves for the time horizon $k = 10$ on LOB-2021. With respect to the same plot on FI-2010 in Fig. 6, the performance of all methods is more similar to one another, which is in line with the findings reported in Fig. 8. The best-performing method is CTABL, with an area under the curve of 5379.5, which only slightly differs from the other top-performing models. On average, the integral of the curves is 5279.4. This result

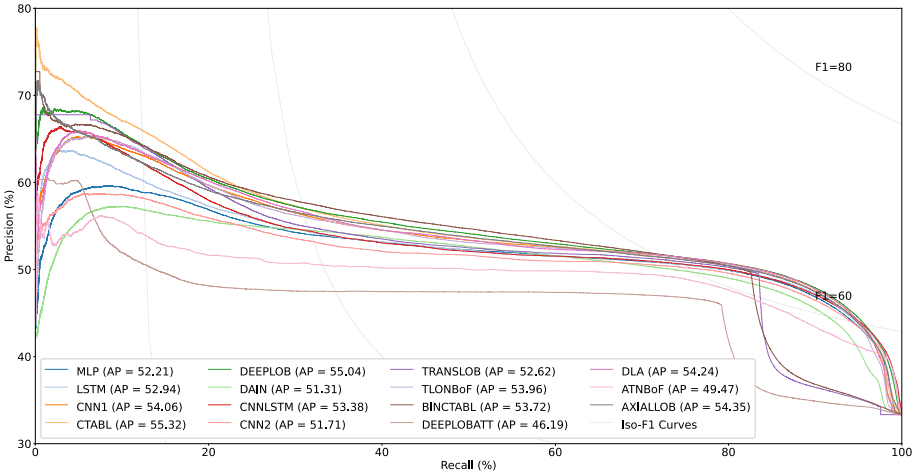


Fig. 9 PR-Curve on LOB-2021 for time horizon $k = 10$

highlights that the models are less reliable on the LOB-2021 dataset and fall afoul of misclassifying price trends.

Table 11 shows the p-values of the T-test on LOB-2021 for the horizon $k = 5$. We performed this test following the same approach used for the results shown in Table 10. Bold p-values correspond to high accordance among the models. Notice that, while on the FI-2010 dataset, there are only nine couples of models with a p-value exceeding the threshold $\alpha = 0.05$, there are as many as 41 pairs of models that are not considerably statistically different on LOB-2021. These results confirm the results depicted in Figs. 8 and 9.

The plots for LOB-2022 are omitted since they show similar properties; indeed, we observe that most models exhibit a similar trend in both LOB-2021 and LOB-2022 datasets. However, the performance curves in these generalizability tests differ from the results obtained on the FI-2010 dataset, shown in Fig. 5. Specifically, for the LOB-2021/2022 datasets, the F1-Score of most models shows an increasing trend as the prediction horizon increases up to $k = 3$, after which it starts to decrease.

To ease readability, in Table 7 we report the F1-Score of all the models, horizons, and periods.

The performance of the models, as reported by the authors of the selected paper, exhibits changes when evaluated on the LOB-2021 and LOB-2022 datasets. These changes show varying degrees of generalizability among the models.

Notably, the ATNBoF model demonstrates the most substantial improvement with respect to the declared performances, showing an average increase of 12.2% across all prediction horizons. A similar improvement is exhibited by MLP and TLONBoF.

Despite this improvement, ATNBoF still exhibits the lowest overall performance, with an average score of 53.1%. It is worth mentioning that ATNBoF is the most sensitive to random initialization.

In contrast, the other models experience a significant decline in performance when evaluated on LOB-2021 and LOB-2022 datasets. For example, the previously best-performing model on the FI-2010 dataset, BINCTABL, shows an average decrease in F1-Score of approximately 19.6% across all prediction horizons. This decline results in a generalizability score of 73.5% (as mentioned in Table 6). However, despite this decline, BINCTABL

Table 7 F1-Score on LOB-2021 and LOB-2022. Columns FI 2010, FI^r 2010, LOB 2021, and LOB 2022, respectively, represent the claimed performance of the models in the respective papers, the performance reproduced with LOBCAST on FI, LOB-2021, LOB-2022

Model	$k = 1$				$k = 2$				$k = 3$				$k = 5$				$k = 10$			
	FI 2010	FI^r **2010**	LOB 2021	LOB 2022	FI 2010	FI^r **2010**	LOB 2021	LOB 2022	FI 2010	FI^r **2010**	LOB 2021	LOB 2022	FI 2010	FI^r **2010**	LOB 2021	LOB 2022	FI 2010	FI^r **2010**	LOB 2021	LOB 2022
MLP	48.3	48.2	48.3	51.1	51.1	44.0	56.2	54.1	–	47.2	58.2	55.9	56.0	49.0	59.2	55.0	–	51.6	55.4	49.3
LSTM	66.3	66.5	49.6	53.7	62.4	58.8	58.0	57.4	–	65.3	60.3	60.6	61.4	66.9	60.6	56.2	–	59.4	56.0	52.6
CNN1	55.2	49.3	52.5	55.3	59.2	46.1	57.7	59.8	–	62.3	60.2	59.3	59.4	65.8	60.1	58.5	–	67.2	56.7	52.6
CTABL	77.6	69.5	55.3	57.8	66.9	62.4	60.7	60.9	–	70.4	62.2	60.8	78.4	71.6	62.2	58.8	–	73.9	57.8	52.0
DEEPLOB	83.4	71.1	55.0	57.0	72.8	62.4	60.4	**62.0**	–	70.8	62.7	**62.4**	80.4	75.4	62.2	60.8	–	77.6	57.4	55.2
DAIN	68.3	53.9	47.7	52.2	65.3	46.7	56.6	54.9	–	53.5	59.1	55.8	–	61.2	60.0	56.5	–	62.8	56.1	51.2
CNNLSTM	47.0	63.5	51.8	55.0	–	49.1	58.1	59.8	–	63.3	59.9	59.2	47.0	69.2	60.1	57.1	47.0	71.0	55.3	53.1
CNN2	46.0	27.6	49.9	51.9	–	35.4	55.9	59.0	–	53.2	58.9	58.7	45.0	67.9	58.8	57.3	44.0	68.5	54.0	52.0
TRANSLOB	**88.7**	61.4	53.8	43.7	**80.6**	54.7	57.8	43.0	–	59.8	60.7	57.5	**88.2**	60.6	60.3	56.6	**91.6**	60.5	55.8	51.0
TLONBoF	53.0	36.5	52.5	53.1	–	51.7	58.0	56.5	–	41.6	60.1	57.1	–	52.4	59.9	55.7	–	66.2	56.0	48.5
BINCTABL	81.0	**81.1**	**57.0**	**58.4**	71.2	**71.5**	**62.4**	**62.0**	–	**80.8**	**63.9**	62.2	88.1	**87.7**	**63.5**	60.4	–	**92.1**	**59.1**	53.2
DEEPLO-BATT	82.4	70.6	54.8	55.8	73.7	54.8	61.1	60.5	76.9	66.0	62.6	62.1	79.4	73.6	62.8	**60.9**	81.5	71.6	59.0	**55.3**
DLA	77.8	79.4	51.2	54.4	–	69.3	58.6	58.0	79.4	78.9	61.3	60.0	79.0	87.1	60.3	57.3	–	52.2	57.1	53.4
ATNBoF	67.9	32.9	49.8	47.8	60.0	34.2	53.1	50.3	–	38.2	54.6	41.3	73.4	48.1	57.2	59.8	–	51.0	50.9	40.9
AXIALLOB	85.1	73.2	54.0	56.9	75.8	63.4	60.7	60.1	**80.1**	72.8	62.6	62.0	83.3	78.3	62.4	59.6	85.9	79.2	57.8	54.6
METALOB	–	81.1	51.1	52.3	–	70.5	56.1	53.3	–	80.3	57.8	55.3	–	87.5	58.4	54.5	–	91.8	56.0	50.9
MAJORITY	–	47.1	51.8	50.6	–	44.9	56.2	49.2	–	59.7	57.5	48.1	–	71.8	56.9	46.7	–	76.3	55.2	44.7

remains the top-performing model when evaluated on the LOB-2021 dataset on almost all the prediction horizons. On these datasets, it exhibits similar performance to DEEPLOB and DEEPLOBATT models.

In Appendix 10.2., we show the agreement matrix of the models' predictions.

6.3.3 Ensemble method discussion

To train *METALOB* without falling into overfitting, we divided the test set of LOB-2021/2022 into three distinct subsets. We allocated 70% of the data for training, 15% for validating, and the remaining 15% for testing the meta-classifier. By implementing these ensemble methods, our objective was to leverage the collective intelligence of ensemble models and potentially achieve performance that surpasses that of individual models. Unfortunately, the ensemble models did not achieve the expected level of performance, as they failed to surpass the performance of the best individual models. A plausible explanation for this phenomenon is the relatively high degree of consensus among the systems, as evidenced by Fig. 13 and Fig. 14 in Sect. 10. of the supplementary material. Moreover, it is likely that the methods converge on cases that are easy to classify and diverge on cases that are difficult to classify.

6.4 Additional experiments: labeling, non-DL models & profit

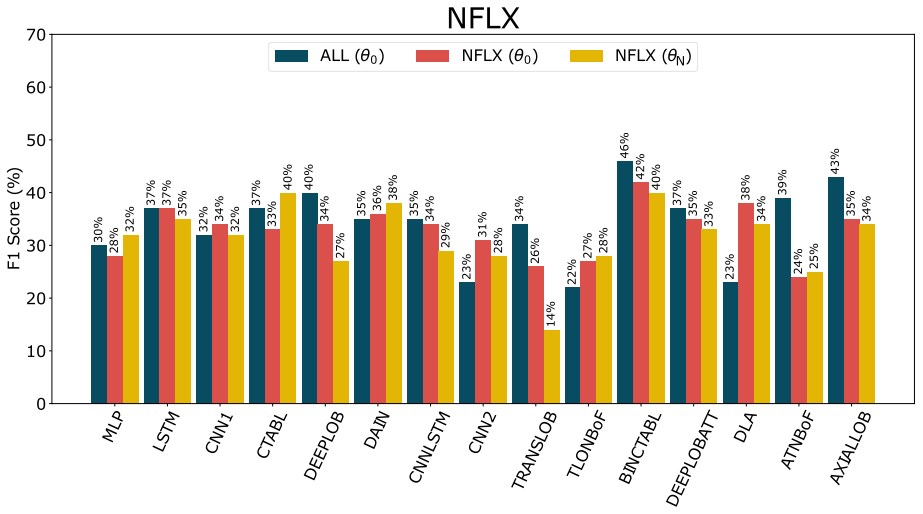
In this section, we delve into additional experiments. In Sect. 6.4.1, we measure the impact of labeling parameters on the quality of the SPTP task. In Sect. 6.4.2, we go beyond deep learning models and explore how tree-based methods perform on the SPTP task using the same experimental setting presented in the previous section. In Sect. 6.4.3, we incorporate profit considerations through backtesting.

6.4.1 Labelling

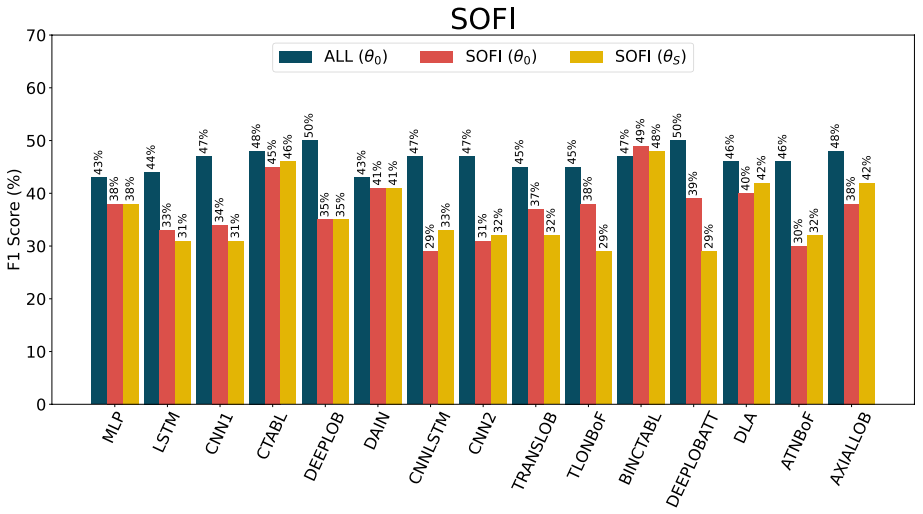
The experiments analyzed in Sect. 6 highlight that the models' performance does not exhibit a clear trend with respect to the prediction horizon. The labelling method is probably the cause of this phenomenon; in fact, classifying trends based on the mid-price tends to embody noise on the nearest horizons. This hypothesis is supported by the work of Zhang et al. (2019); specifically, they generated a dataset using an alternative labelling method that relies on the mean of the previous and next k mid-prices to identify trends. Interestingly, they show an inverse trend in performance with respect to the horizons: the best performance is achieved with the shortest horizon and deteriorates when it increases. While exploring various labelling techniques is beyond the scope of this benchmark, we provide an initial investigation in this direction. Specifically, focusing on $k = 5$ in LOB-21, we select two stocks, NFLX and SOFI.

Based on Eqs. 1 and 2, we can define θ_N and θ_S as the thresholds that balance the occurrences of the classes for the stocks NFLX and SOFI, respectively. Similarly, we can define θ_0 as the threshold that balances the occurrences of the classes for the ensemble of six stocks within the dataset.

Figure 10 shows the results of three different training settings: (i.) **ALL** ($\boldsymbol{\theta_0}$) represents the training of the models over the ensemble of all the six stocks using the threshold θ_0; (ii.) **NFLX** ($\boldsymbol{\theta_0}$) (**SOFI** ($\boldsymbol{\theta_0}$)) represents the training of the models over NFLX (SOFI) stock



(a) NFLX



(b) SOFI

Fig. 10 Different labelling strategies on NFLX and SOFI stocks for $k = 5$

using the threshold θ_0. (iii.) **NFLX** (θ_N) (**SOFI** (θ_S)) represents the training of the models over NFLX (SOFI) stock using the threshold θ_N.

In the case of SOFI, all methods, except for BINCTABL, achieve the highest performance in the **ALL** (θ_0) setting. This indicates that these models are able to extract useful signals from other stocks, reducing overfitting and improving overall performance. On the other hand, comparing the **SOFI** (θ_0) and **SOFI** (θ_S) settings does not provide significant insights. This suggests that the balancing of the three classes is not crucial for achieving higher performance. This is even more the case for NFLX in Fig. 10a, considering that the imbalance due to θ_0 is much higher (see Table 3).

Table 8 Random Forest parameters

Hyper Parameter	Values
n_estimators	[200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]
max_depth	[10, 25, 50, 75, 100]
min_samples_leaf	[1, 2, 4]
min_samples_split	[2, 5, 10]
temporal shape (h)	[5, 10, 15, 50, 100, 300]

Table 9 XGBoost parameters

Hyper Parameter	Values
n_estimators	[100, 250, 500, 750, 1000, **1250**]
max_depth	[3, 4, 5, **6**, 7, 8, 9, 10]
booster	**gbtree**
eta	[0.01, 0.1, 0.2, 0.3, 0.4]
min_child_weight	[0, 2, 4, 6, 8]
colsample_bytree	[0.5, 0.75]
colsample_bylevel	[0.5, 0.75]
temporal shape (h)	[5, 10, 15, 50, 100, 300]

These results indicate that the labelling mechanism should be revised from its current definition and be agnostic with respect to the balancing involved. Therefore, trend definition should not solely depend on the magnitude of the future price shift relative to the current price. Other factors, such as persistence over time and volume considerations, should also be taken into account. A more comprehensive discussion of the limitations and challenges associated with the labelling mechanisms can be found in the final discussion and conclusions section.

6.4.2 Random forest & XGBoost for SPTP

To test the quality of the predictions of DL over non-DL models, we conducted an empirical investigation focusing on the predictive capabilities of two of the most popular non-DL models: Random Forests and XGBoost. These experiments are motivated by the results presented in some previous works (e.g., Grinsztajn et al. 2022; Shwartz-Ziv and Armon 2022), which show that some tree-based models outperform recently proposed DL models on tabular data. Employing the standard experimental setup detailed in Sect. 6 and carefully tuning hyperparameters (refer to Table 8 and Table 9) our analysis revealed an F1-Score of **51%** for the Random Forest model and **65%** for XGBoost on the FI-2010 dataset with horizon $k = 5$. We obtained these results using class weights and the hyperparameter values in bold in the Tables. We acknowledge that for Tree-based algorithms, normalization might lead to worse performance, but as said before, the FI-2010 dataset is released already normalized, and to guarantee a fair comparison, we decided to apply standardization to the LOB 2021/22 dataset.

As illustrated in Fig. 5a and Fig. 12, our results indicate a competitive performance of these non-DL models when compared to several DL models, including ATNBoF, MLP, TLONBoF, TRANSLOB, and DAIN. However, the hypothesis that non-DL models outperform DL counterparts in this specific task does not seem to hold. While non-DL models exhibit notable performance, DL methods show a substantial advantage in predicting price trends. We recall that the F1-Score of the best-performing model on the FI-2010 with $k = 5$ was 87.7%, obtained by BINCTABL. This is in line with the results of the experiments in Nti et al. (2020). The tabular nature of LOB data, with its geometric properties, such as local dependence Sirignano (2019), and visual indicators embedded in column positions of the LOB, seems to align better with the strengths of DL models, especially those based on convolution. Furthermore, we remark that LOB data are multivariate time series, and SOTA forecasting papers in this domain are primarily dominated by deep learning models (Mahmoud and Mohammed 2021; Torres et al. 2021; Lim and Zohren 2021).

6.4.3 Profit analysis

As a final benchmark test, we conducted a trading simulation using our framework, relying on Backtesting.py Python library.[3] As highlighted by Olorunnimbe and Viktor (2023), most of the existing literature in the SPTP field neglects backtesting, even though it is essential for evaluating the performance of algorithmic trading strategies and for potential real-world use.

We performed backtesting using the same period as the test set of the LOB-2021 dataset, i.e., from 2021-07-13 to 2021-07-15. To perform backtesting, we generated an Open High Low Close (OHLC) time series with a 10 events period. The OHLC is an aggregation technique to summarize periods of a time series, e.g., minutes, hours, days, or a number of events (10 in this case). Each data point of the series represents four aggregates of the considered period. The *Open* represents the first price of the period; *High* is the highest price of the period; *Low* is the lowest price of the period; *Close* is the last price of the period.

We underline that the use of LOB data is most often associated with High-Frequency Trading (HFT), i.e., strategies that analyze this data in real time to make split-second decisions about trade executions. We remark that a trading action (buy/sell/hold) is taken every ten events, so at the end of the backtesting simulation, for each stock, hundreds of thousands of orders are placed and filled.

We base our trading simulation on the methodology of the seminal paper Zhang et al. (2019) in this field, in which the authors conducted a similar experiment. We established certain parameters for our simulation. Firstly, we set the number of shares per trade to a fixed value of 1, simplifying our analysis and assuming a negligible market impact. Furthermore, our simulated trader begins with an initial capital of $10.000, and we make the assumption of no transaction fees.

The *trading strategy* relies on the models and operates by generating signals every 10 events to predict subsequent price movements. These signals, categorized as *up*, *stationary*, or *down*, determine the trading action. When the signal is *up*, the simulated trader places a buy order. Conversely, if the signal is *down* and the trader currently holds a long position, he places a sell order. In cases where the signal is *stationary*, the trader takes no action. The orders are filled at the next open price.

[3] https://kernc.github.io/backtesting.py/

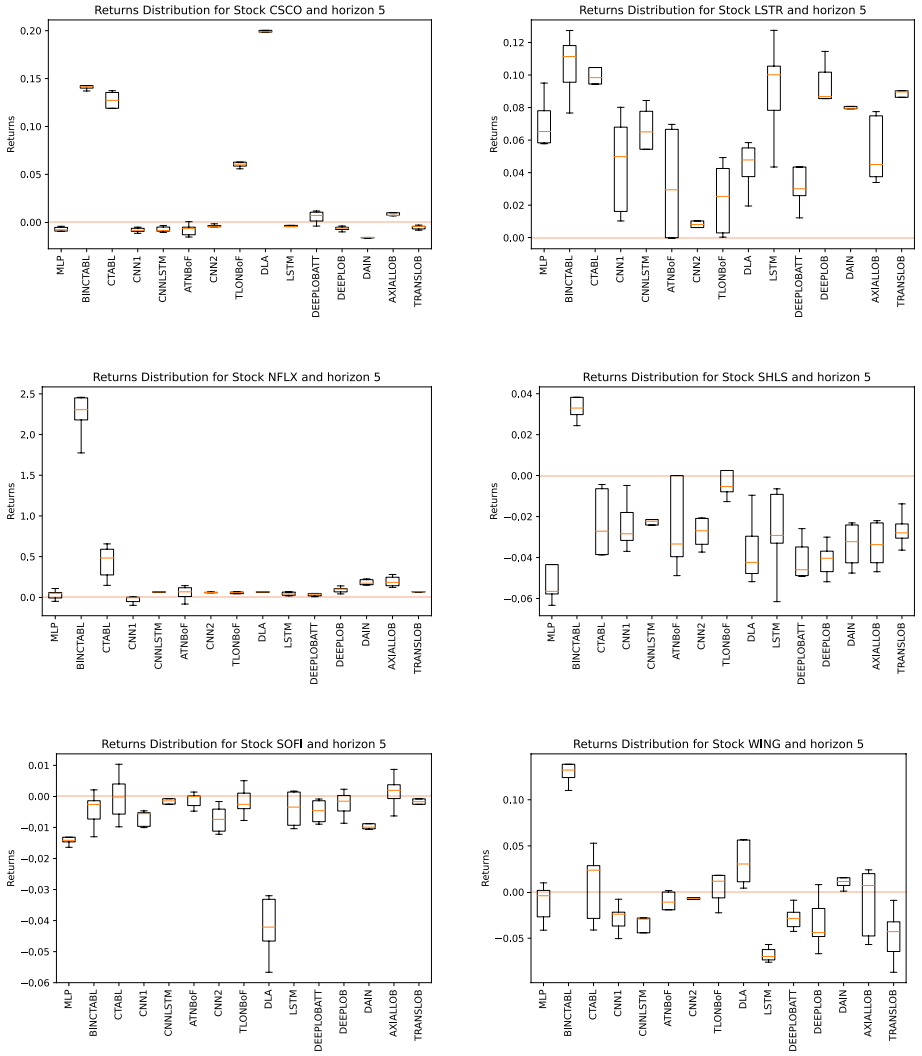


Fig. 11 Distribution of returns on five seeds

The results of the trading simulation for each stock are presented in Fig. 11. The strongest correlation observed is between the daily returns of the stocks, as shown in Table 3, and the returns of the strategy described above. In fact, the two stocks with the highest positive daily returns (namely LSTR and NFLX) are the only ones for which the strategy is profitable. On the other hand, the two stocks with the highest negative daily returns (SOFI and SHLS) are the ones for which most models show a negative return. Another correlation, albeit less strong, is between the volatility of the stocks and the return of the models. Specifically, lower volatility is associated with higher model returns.

We recognize the limitations of this simulation. For instance, we do not perform portfolio optimization or position sizing, we assume the trades execution at the mid-price, and we

ignore transaction costs, but a realistic and sophisticated algorithmic trading simulation is beyond the scope of this study and remains an interesting aspect for future research.

7 Discussion and conclusions

Our findings highlight that price trend predictors based on DNNs using LOB data are not consistently reliable as they often exhibit non-robust and non-generalizable performance. Our experiments demonstrate that the existing models are very susceptible to hyperparameter selection, randomization, and experimental context (stocks, volatility, market, historical period). In addition, the experimental setup fails to capture the intricacies of the real-world scenario. This lack of robustness and generalizability makes them inadequate for practical applications in real-world settings.

7.1 Models

Our results lead to a crucial observation: on the LOBSTER dataset, SOTA DL models for LOB data exhibit low generalizability. We suggest that this phenomenon is due to two factors: the higher complexity of the LOBSTER dataset compared to the FI-2010 dataset and the overfitting of the best-performing models to the FI-2010 dataset, which lowers their performance on the LOBSTER dataset. In fact, in the original papers, all the considered models (except for DEEPLOB, DEEPLOBAT and DLA) were trained, validated, and tested only on FI-2010, which is a smaller dataset with less frequent and voluminous orders than those contained in LOBSTER-derived datasets. Our conjecture is supported by insightful findings reported in the existing literature: several works (e.g., Orimoloye et al. 2020; Ruff et al. 2021; Najafabadi et al. 2015) have shown that while some datasets exhibit simple patterns that can be effectively captured by a shallow model, others may require deeper architectures to model complex relationships.

Another key finding of this study is that the top models with the highest performance on both datasets employ attention mechanisms. This suggests that the attention technique enhances the extraction of informative features and the discovery of patterns in LOB data. However, in general, it appears that current models cannot cope with the complexity of financial forecasting with LOB data.

7.2 Dataset

Financial trends can be influenced by both local and international political events, in fact, political actions and decisions can significantly impact economic conditions, market sentiment, and investor confidence Engle et al. (2013). These factors are not captured by LOB data alone. For this reason, we believe that price predictors may benefit from integrating LOB data with additional information, for example, sentiment analysis relying on social media and press data, representing an easily accessible source of exogenous factors impacting the market Ren et al. (2018). This is particularly true for mid- and long-term price trend prediction, whereas it might not hold for HFT strategies Bouchaud et al. (2018). We remark that micro and macroscopic market trends are fundamentally different, and the microscopic behaviour of the market is very much driven by HFT algorithms, making it almost exclusively dependent on financial movements rather than external factors. In this scenario, granular and raw LOBs may suffice to provide data for price trend prediction. Several works have used sentiment analysis for price

trend prediction. The work in Jin et al. (2020) combines comments made by investors on the online platform StockTwits with Apple stock price time series and uses LSTM for stock closing price prediction. By creating a dataset composed of tweets and historical stock prices, the work in Xu and Cohen (2018) proposes a new architecture for stock price prediction. Similar approaches are used in Li et al. (2014), Nguyen et al. (2015), where the authors show that the performance of their predictors improves when their dataset is enriched with sentiment data. Despite the wide research in this field, to the best of our knowledge, no one has ever used LOB data together with sentiment analysis for price trend prediction. The existing literature suggests that this could be a valuable research direction.

Another weakness in dataset generation is the potential for training, validation, and test splits to have dissimilar distributions. This occurs due to the distinct characteristics of the historical periods covered by the stock time series. This can negatively affect the model's ability to generalize effectively and make reliable predictions on unseen data. A last limitation regarding the dataset is the representation of the limit order book which has been shown to be sensitive to permutations by Wu et al. (2021). In Wu et al. (2022), the same authors proposed robust alternative representations.

7.3 Labelling

As we discussed in Sections 2, 5 and 6.3, the choice of the threshold for class definition in Eq. 2 plays a crucial role in determining the trend associated to a market observation. We believe that current solutions present room for improvement. As discussed in Sect. 5, in FI-2010, the parameter θ was chosen to obtain a balanced dataset in the number of classes for the horizon $k = 5$ (which is the mean value of the considered interval in the set $\mathcal{K}$). Thus, θ is not chosen in accordance with its financial implication but rather serves the purpose of improving model performance. We recall that the dataset is made of different stocks. With such a labelling system, with a fixed θ, stocks with low volatility become associated with stable trends, as their behaviour is overshadowed by stocks exhibiting higher volatility. Good practices that could be investigated are to use a weighted look-behind moving average to absorb data noise instead of mid-prices as in Eq. 2 or to define a dynamically adapting θ which accounts for changing trends of a stock's mid-price. Moreover, the labelling approach of Eq. 2, used by all surveyed models, fails to leverage important aspects available in LOB data, so another possible improvement is the definition and use of other insightful features that can be extrapolated from the LOB in addition to the mid-price. Such values could encapsulate other peculiar and informative features, such as stocks' spread and volumes which directly influence stock volatility and so the returns.

7.4 Profit

In the context of stock prediction tasks, it is of utmost importance to go beyond standard statistical performance metrics such as accuracy and F1-Score and incorporate trading simulations to assess the practical value of algorithms. SPTP predictors' ultimate measure of success lies in their ability to generate profits under real market conditions. It is essential to conduct trading simulations using real simulators that go beyond testing on historical data. Recent progress has been made in the context of reactive simulators (Coletta et al. 2022a, b; Mizuta 2016; Shi and Cartlidge 2023).

7.5 Limitations and risks

We acknowledge that our study is subject to some limitations, which should be considered when interpreting our findings. First, we conducted a grid hyperparameter search for the models which did not specify them. Since hyperparameter search is not exhaustive, our chosen best hyperparameters could potentially undermine the quality of the original systems. Secondly, due to computational resource limitations, we could not train the benchmarked models on LOB datasets spanning longer periods, e.g., years rather than weeks. We recognize that doing so could have led to different results.

We also highlight that using DL or, more generally, AI models for solving SPTP and exploiting them for trading can have a number of risks. Some of them are inherently technical. This is the case for data biases, that is, incomplete or unrepresentative data, which can cause predictive algorithms to favor groups that are better represented in the training data Boukherouaa et al. (2021). Lack of explainability is another risk that could expose organizations to vulnerabilities (such as biased data, unsuitable modeling techniques, or incorrect decision-making) and potentially undermine the trust in their robustness Silberg and Manyika (2019). AI models for trading are also vulnerable to cyber-attacks. Malicious users can exploit AI model vulnerabilities to evade detection and prompt the models to make the wrong decisions or to extract information by manipulating data at some stage of the model lifecycle Comiter (2019). Other risks have an ethical nature and can impact financial stability. One of them is inequality among investors. As training and predicting are expensive in terms of hardware equipment and energy, and because of the challenges in model interpretation and prediction, AI trading can lead to a concentration of information among those who can afford the required technology, exacerbating income inequality and asymmetry in the market, with an uncertain impact on financial stability (Robledo Costales 2023; Boukherouaa et al. 2021). Because of limited regulation of the AI trading systems, there is a lack of transparency, making it difficult to detect possible unfair strategies Robledo Costales (2023). Governing legal and regulatory framework regulators should welcome the advancements of AI in finance and undertake the necessary preparations to harness its potential advantages and address its associated risks.

8 Disclaimer

This paper was prepared for informational purposes in part by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates (“JP Morgan”), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

Models description

Tsantekidis et al. (2017a) (2017) use a LSTM to predict price directions considering moving averages of the mid-price over the past and the future k steps. In the same year, the same authors proposed in Tsantekidis et al. (2017b) a model based on a Convolutional Neural Network (CNN) (CNN1) for future mid-price movement predictions from large-scale high-frequency limit order data. The proposed architecture is composed of a series of convolutional and pooling layers followed by a set of fully connected layers that are used to classify the input. The parameters of the model are learned by minimizing the categorical cross-entropy. In Tsantekidis et al. (2020) (2020), the same research group proposed two new architectures. The first one (CNN2) uses a series of convolutional layers for capturing the temporal dynamics of time series extracted from a LOB and for correlating temporally distant features. In the last convolutional layer, CNN2 retains the temporal ordering by flattening only the dimensions of the convolution. The authors then propose an architecture that merges the described CNN with an LSTM, that we call CNNLSTM. Initially, the CNN is used for feature extraction for the LOB time series. It produces a new time series of features with the same length as the original one, which is then passed to the LSTM module for classification.

Tran et al. (2018) in 2018 introduced a new NN architecture for multi-variate time series that incorporates an attention mechanism in the temporal mode. The authors call this architecture Temporal Attention-Augmented Bilinear (TABL), as it applies a bilinear transformation to the input, which consists of a set of samples at different time stamps. The Bilinear Layer (BL) is able to detect feature and time dependencies within the same sample and is augmented with a temporal attention mechanism to capture interactions between different time instances. The authors define three different network configurations, called A(TABL), B(TABL), and C(TABL), with 0, 1, and 2 hidden layers, respectively. In our experiments, we consider C(TABL), which outperforms the others. In Tran et al. (2021) (2021), the same authors extended the solutions implemented in Tran et al. (2018) by integrating a data-driven normalization strategy that takes into account statistics from both temporal and feature dimensions to tackle potential problems posed by non-stationarity and multimodalities of the input series. The new model is called BINCTABL.

Passalis et al. (2019) introduce the DAIN (Deep Adaptive Input Normalization) three-step layer that adaptively normalizes data depending on the task at hand, instead of using some fixed statistics calculated beforehand as in traditional normalization approaches. DAIN works as follows: in the first layer, called the adaptive shifting layer, the mean of the current time series is scaled by the weight matrix of the first neural layer. The resulting vector is passed to the adaptive scaling layer, which first computes the standard deviation of the original feature vector with respect to the shifted one and then scales this result using the weight matrix of the scaling layer. The last layer, called *adaptive gating layer*, is meant to suppress features that are not relevant by applying a sigmoid function in order to neglect features with excessive variance, which could hinder network generalization. The authors integrate DAIN in three different architectures, a MLP proposed in Nousi et al. (2019), a CNN as in Tsantekidis et al. (2017b) and RNN Cho et al. (2014). In our experiments, we consider the architecture with the highest performance, namely the MLP.

Zhang et al. (2019) (2019) propose DEEPLOB. The authors propose a smooth data labelling approach based on mid-prices to limit noise and discard small oscillations. They propose a 3-block architecture composed of standard convolutional layers, an Inception

Module, and a LSTM layer. The first two elements are used for feature extraction, whereas the LSTM layer captures time dependencies among the extracted features.

Wallbridge (2020) (2020) introduce TransLOB, a new DL architecture for mid-price movement prediction, composed of two main components: a convolutional module made up of five dilated causal convolutional layers and a transformer module, composed by two transformer encoder layers, each made up of a combination of multi-head self-attention, residual connections, normalization, and feedforward layers. Between the convolutions and the transformer module, the tensor is passed to a normalization layer and concatenated to a positional encoding.

Passalis et al. (2020) (2020) propose a model for high-frequency limit order book data based on Temporal Logistic Neural Bag-of-Features formulation (TLoNBoF). Given a collection of time series, TLoNBoF extracts features with a 1-D convolutional layer to capture the temporal relationships between succeeding feature vectors. Then the features are transformed into vectors of constant length, i.e., their length must be invariant to the length of the input time series. To cope with this, the authors define a Temporal Logistic Neural Bag-of-Features formulation to aggregate the extracted feature vectors. A fine-grained temporal segmentation scheme is also proposed to capture the temporal dynamics of the time series. To this end, the transformed feature vectors are segmented into three temporal regions to capture the short-term, mid-term, and long-term behaviour of the time series.

In 2021, Zhang and Zohren (2021) adopt Sequence-to-Sequence (Seq2Seq) (Sutskever et al. 2014; Cho et al. 2014) and Attention Luong et al. (2015) to recursively generate multi-horizon forecasts and build a predictor called DEEPLOBATT. A typical Seq2Seq model consists of an encoder that analyses the input time steps to extract meaningful features. Then, only the last hidden state from the encoder is used to make estimations, which penalizes the processing of long sequence input. To overcome this limitation, the Attention module accesses hidden states of the encoder and assigns a proper weight to each hidden state. Each input contains the most recent 50 updates, and each update includes information for both the ask and bid of a LOB. Therefore, a single input has the dimension (50, 40), and each output consists of a multi-horizon prediction of all 5 points of the FI-2010 dataset. As an encoder, they adapt a previous model, namely DeepLob Zhang et al. (2019), to extract representative features from raw LOB data while they experiment with both Seq2Seq and Attention models for the decoder.

Guo and Chen (2022) (2022) propose a novel architecture for price trend prediction named Deep Learning Architecture (DLA). Firstly, the dataset is preprocessed and aggregated at different time windows. Once extracted, the features are given as input to the three-phase proposed architecture. The first phase uses Temporal Attention to adaptively assign attention weights to each moment of the sliding window. The processed data is passed to a stacked Gated Recurrent Unit (GRU) architecture to obtain an accurate representation of the analysed trends, which is complex and nonlinear. The GRU architecture consists of two hidden GRU layers to generate as output the hidden state at each time period. This is given to the second temporal attention stage, which is used to generate more accurate attention weights. The proposed solution is compared to several other models in the literature, including C(TABL) Tran et al. (2018), DeepLOB Zhang et al. (2019) and TLo-NBoF Passalis et al. (2020). The proposed solution achieves very high performance on the FI-2010 dataset outperforms the other models. The authors analyse the performance of their model by varying several parameters, including label thresholds and the choice of the time step.

Tran et al. (2022) extend the solution proposed in Passalis et al. (2017), which introduces a neural bag-of-features (N-BoF)-based method for building a *codeword* that is eventually fed to a classifier. In Tran et al. (2022), the neural bag-of-feature model was enhanced

by incorporating a 2D-Attention (2DA) module that highlights important elements in the matrix data while discarding irrelevant ones by zeroing them out. The 2D-Attention function performs a linear interpolation between the input data matrix and input data matrix filtered by an attention mask matrix that encodes the importance of the columns of the original input. The proposed 2DA block can be applied to the features to highlight or discard the outputs of certain quantization neurons, whose results are considered equally important in the NBoF model for every input sequence (Codeword Attention). The resulting model is called ATNBoF. The 2DA function can also be applied to lend weight to salient temporal information, which is otherwise aggregated and equally contributing to the quantized features in the NBoF model (Temporal Attention).

Kisiel and Gorse (2022) (2022) propose Axial-LOB, a model based on axial attention for price trend prediction. Unlike the naive attention mechanism, axial attention factorizes 2D attention into two 1D attention modules, one along the width (feature) axis, and a second one along the height (time) dimension. Raw values of the LOB are preprocessed and passed to the axial attention block: Each layer of the attention block is preceded and followed by a module composed of 1×1 convolutions, batch normalization, and ReLu activation to adjust the number of channels in the intermediate layers of the network. For training the axial attention module, the authors use mini-batch Stochastic Gradient Descent (SGD) by minimizing the cross-entropy loss between the predicted class probabilities and the ground truth label. The authors compare the performance of the proposed model against the solutions adopted in Tsantekidis et al. (2017b), Tran et al. (2018), Zhang et al. (2019) in terms of precision, recall, and F1-Score on the FI-2010 dataset. Axial-LOB proves to have improved performance with respect to these works while being simpler in terms of the number of parameters.

Additional experimental results

Robustness

Figure 12 shows a bar chart representing the F1-Score of the 17 models reproduced using the LOBCAST framework for the five prediction horizons $\mathcal{K}$. The plot shows black empty bars representing the declared performance in the corresponding paper, when applicable. In the figure, the number on the bar represents the obtained performance in LOBCAST on the FI-2010 dataset, and the value in brackets indicates the difference between the obtained performance and the originally declared performance in the respective paper. We highlight that not all papers declare their performance for all the horizons. The figure clearly highlights how robust the considered models are. Surprisingly, for CNN2 and CNNLSTM, our experiments achieved noticeably higher performance than the one declared in the original paper.

The largest discrepancy is observed for TRANSLOB and ATNBoF (or TNBoF-TA), whose average performances differ by 28% from the original results. On average, ATNBoF achieves an F1-Score of only 40.9%. This substantial deviation from the claimed performance highlights the challenges and limitations associated with this particular model.

Figure 13 shows the agreement matrix of the models for the horizon $k = 5$. As expected, the highest agreement (≈80%) is among the best-performing models, namely BINCTABL,

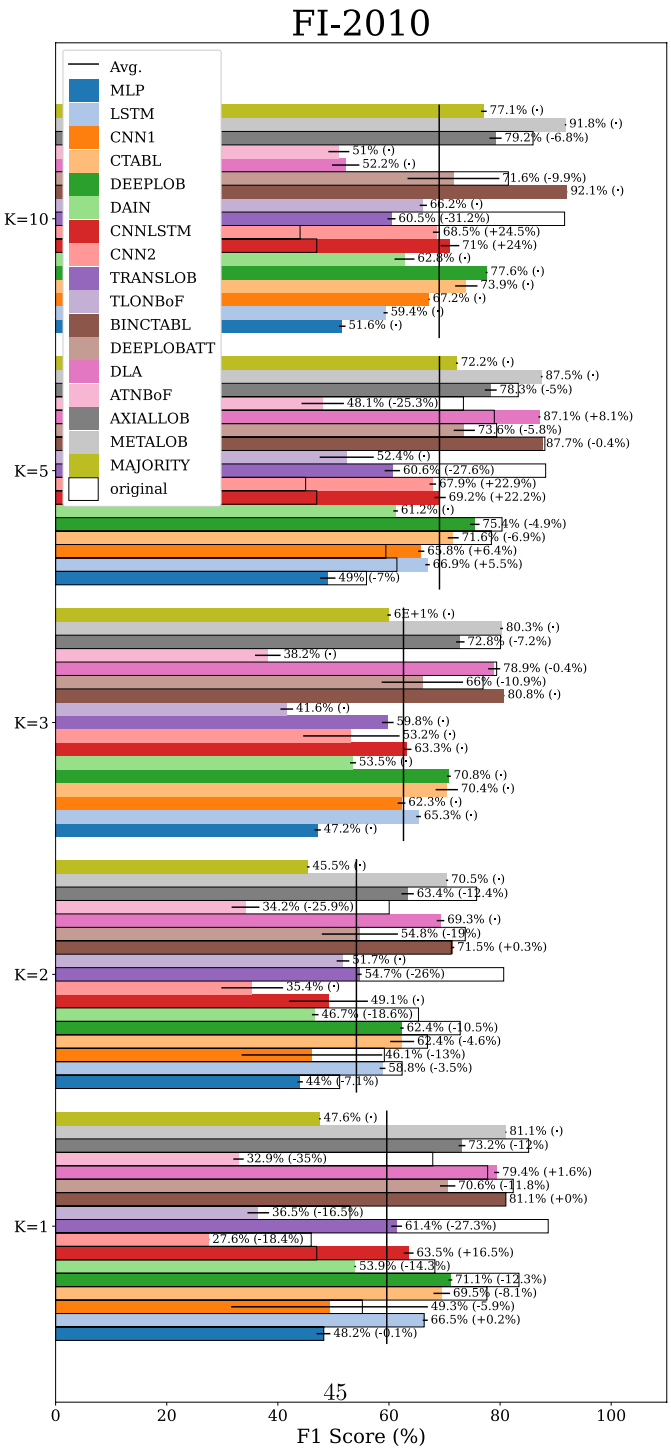


Fig. 12 F1-Score on FI-2010

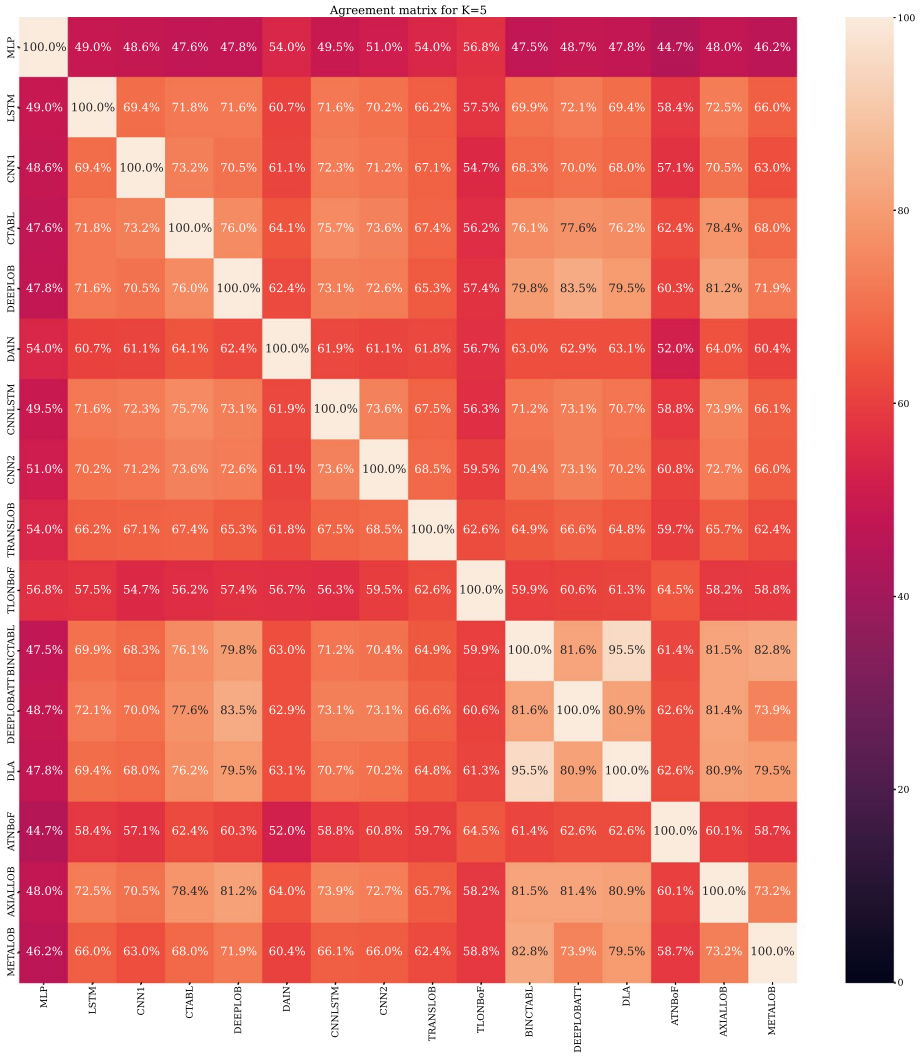


Fig. 13 Agreement matrix FI-2010 in the horizon $k = 5$

AXIALLOB, DEEPLOB, CTABL, DLA and DEEPLOBATT. The model that exhibits less correlation with the other models is MLP.

The best-performing model in our benchmark is BINCTABL, reaching 92.1% of F1-Score on time horizon $k = 10$. Moreover, it notably closely aligns with the performance reported in the paper presenting it. Specifically, BINCTABL introduces an Adaptive Bilinear Normalization layer to CTABL, enabling joint normalization of the input time series along both temporal and feature dimensions. This enhancement yields a remarkable improvement, with an average increase of 9.2% in the F1-Score compared to the second-best model (DLA). Interestingly BINCTABL is composed only of 11.446 parameters, which makes it very fast at inference time (0.0005s).

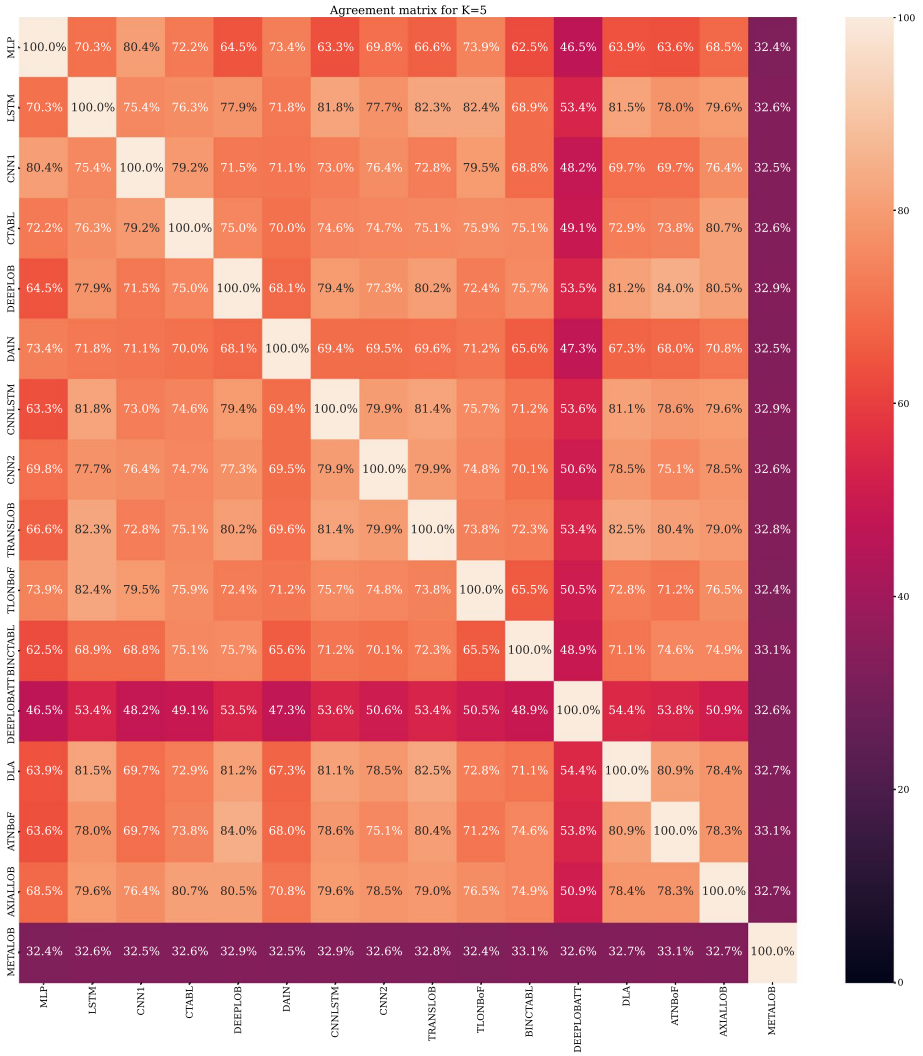


Fig. 14 Agreement matrix on LOB-2022

Generalizability

Figure 14 shows the agreement matrix on LOB-2022. Considering the more flattened performances of the models on the LOB-2021/2022 dataset compared to the FI-2010 dataset, the agreement percentages among the models are consistently high, and no distinct patterns are observed. Unlike FI-2010, where METALOB predicted as BINCTABL 82.8% of the time, on LOB-2022 (and also LOB-2021) METALOB showed no preference for any model, resulting in a balanced agreement rate ($\approx 33\%$) among all models. We decided not to include the agreement matrix of LOB-2021 because it was similar to LOB-2022.

Table 10 p-values of the T-test on FI-2010 for horizon $k = 5$

	LSTM	CNN1	CTABL	DEEPLOB	DAIN	CNNLSTM	CNN2	TRANSLOB	TLONBoF	BINCTABL	DEEPLO-BATT	DLA	ATNBoF	AXIALLOB	METALOB	MAJORITY
MLP	4e−06	2e−06	2e−08	1.3e−08	2.1e−05	3e−08	1e−06	2e−06	**.240675**	6e−07	9e−08	5e−07	**.671884**	2e−09	3.5e−06	4e−06
LSTM	–	.01194	.000253	3e−06	3.9e−08	.008082	.032604	.000457	.003819	4.6e−08	.001799	1e−09	.000536	5e−06	2e−09	1e−06
CNN1	–	–	3.8e−05	1.9e−06	2e−06	.000953	.000898	.000777	.005021	1.4e−07	.000751	1.8e−08	.000634	8.4e−07	3.2e−08	5e−06
CTABL	–	–	–	.000373	4e−06	.010175	.000479	3e−06	.001094	5e−06	**.113016**	4e−06	.000135	1.6e…05	4e−06	**.256306**
DEEPLOB	–	–	–	–	1.6e−07	1.6e−05	2e−06	1e−06	.000551	9e−06	**.125099**	6e−06	8.1e−05	.003406	6e−06	.001316
DAIN	–	–	–	–	–	1.8e−05	2.7e−07	**.452821**	.022074	2e−08	.00013	0.0	.002186	5.8e−07	1e−09	2.1e−08
CNNLSTM	–	–	–	–	–	–	**.062585**	1.6e−05	.001841	3e−06	.006636	2e−06	.000209	2e−06	2e−06	.003497
CNN2	–	–	–	–	–	–	–	.000131	.002906	2.8e−07	.002747	5.2e−08	.000397	2e−06	8.3e−08	5.1e−05
TRANSLOB	–	–	–	–	–	–	–	–	.025307	3e−06	8e−06	2e−06	.001566	8e−08	2e−06	6.2e−05
TLONBoF	–	–	–	–	–	–	–	–	–	.000131	.000388	.000139	**.202507**	.000296	.000133	.001225
BINCTABL	–	–	–	–	–	–	–	–	–	–	.000119	.00176	3.2e−05	6.2e−05	.037839	4e−09
DEEPLOBATT	–	–	–	–	–	–	–	–	–	–	–	.000131	2.4e−05	.004296	.000119	**.234532**
DLA	–	–	–	–	–	–	–	–	–	–	–	–	3.3e−05	6e−05	.007015	0.0
ATNBoF	–	–	–	–	–	–	–	–	–	–	–	–	–	3.9e−05	3.2e−05	.000219
AXIALLOB	–	–	–	–	–	–	–	–	–	–	–	–	–	–	5.5e−05	.000271
METALOB	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0.0

Table 11 p-values of the T-test on LOB-2021 for horizon $k = 5$

	LSTM	CNN1	CTABL	DEEPLOB	DAIN	CNNLSTM	CNN2	TRANSLOB	TLONBoF	BINCTABL	DEEPLO-BATT	DLA	ATNBoF	AXIALLOB	METALOB	MAJORITY
MLP	.018607	.021589	2.6e−05	2.6e−05	.038899	**.10573**	**.283349**	**.08208**	**.05829**	5e–06	6.4e−05	.010697	**.308099**	7.7e−05	.029677	.009123
LSTM	–	**.279997**	.010331	.00918	**.176818**	**.310756**	.00593	**.609023**	**.129668**	.00074	.003929	**.453778**	**.122387**	.008292	.002487	.002129
CNN1	–	–	5.7e−05	.000129	**.460271**	**.85318**	.010043	**.728193**	**.250598**	2.4e−07	5e–06	**.394171**	**.168575**	7e–06	3.9e−05	3.2e−05
CTABL	–	–	–	**.887803**	.000265	.001914	6.5e−05	.010641	5.7e−05	.000942	.039844	.000154	.046493	**.455996**	1e–06	8e–06
DEEPLOB	–	–	–	–	.00047	.001623	4.9e−05	.009341	.000143	.00241	**.076299**	.000354	.045168	**.612903**	3e-06	1.9e−05
DAIN	–	–	–	–	–	**.893312**	.01842	**.539226**	**.398116**	4e-06	4e-08	.046864	**.183409**	1e-06	.000187	7.9e−08
CNNLSTM	–	–	–	–	–	–	.028972	**.675565**	**.700262**	.000243	.001296	**.545926**	**.178718**	.002261	.007487	.004825
CNN2	–	–	–	–	–	–	–	.026103	.022311	2.3e−05	.000153	.007106	**.423282**	.000192	**.381685**	**.111617**
TRANSLOB	–	–	–	–	–	–	–	–	**.42584**	.00128	.004928	**.989729**	**.15165**	.009266	.011145	.007343
TLONBoF	–	–	–	–	–	–	–	–	–	1.2e−07	3.8e−08	.026764	**.196436**	1.2e−07	6.1e−05	1e-06
BINCTABL	–	–	–	–	–	–	–	–	–	–	.00391	2.5e−07	.022831	.000308	8e-09	7.8e−08
DEEPLOBATT	–	–	–	–	–	–	–	–	–	–	–	1.8e−07	.033816	.012705	1.5e−07	0.0
DLA	–	–	–	–	–	–	–	–	–	–	–	–	**.150606**	1e-06	1.2e−05	1e-06
ATNBoF	–	–	–	–	–	–	–	–	–	–	–	–	–	.042348	**.519783**	**.629823**
AXIALLOB	–	–	–	–	–	–	–	–	–	–	–	–	–	–	8.9e−08	2e-09
METALOB	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	**.091866**

Acknowledgements This research was funded by JPMorgan Chase AI Research Faculty award "*Understanding interdependent market dynamics: vulnerabilities and opportunities*". We also thank Poste Italiane for funding a Ph.D. scholarship on Financial applications of Artificial Intelligence.

Author Contributions M.P., G.M., and L.B. were responsible for coding the framework for the experiments, writing the main manuscript text, and preparing figures. V.A., A.C., and I.C. were responsible for writing the main manuscript text, and preparing figures. All authors collaborated on enhancements and reviewed the manuscript.

Funding Open access funding provided by Università degli Studi di Roma La Sapienza within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors declare no competing interests.

References

Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M et al (2016) Tensorflow: a system for large-scale machine learning. In: OSDI, vol. 16, pp. 265–283. Savannah, GA, USA

Al-Alawi AI, Alaali YA (2023) Stock market prediction using machine learning techniques: Literature review analysis. In: 2023 International Conference On Cyber Management And Engineering (CyMaEn), pp. 153–157. https://doi.org/10.1109/CyMaEn57228.2023.10050933

Alsulmi M (2022) From ranking search results to managing investment portfolios: exploring rank-based approaches for portfolio stock selection. Electronics 11(23):4019

Baker M (2016) Reproducibility crisis. Nature 533(26):353–66

Bennett S, Clarkson J (2022) Time series prediction under distribution shift using differentiable forgetting. arXiv preprint arXiv:2207.11486

Berlin G, LOBSTER: Limit Order Book System. https://lobsterdata.com/

Biewald L (2020) Experiment Tracking with Weights and Biases. Software available from wandb.com. https://www.wandb.com/

Bouchaud J-P, Bonart J, Donier J, Gould M (2018) Trades, quotes and prices: financial markets under the microscope. Cambridge University Press, Cambridge

Boukherouaa EB, Shabsigh MG, AlAjmi K, Deodoro J, Farias A, Iskender ES, Mirestean MAT, Ravikumar R (2021) Powering the Digital Economy: Opportunities and Risks of Artificial Intelligence in Finance. International Monetary Fund, Washington, D.C.

Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler DM, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020) Language models are few-shot learners. arXiv:2005.14165

Cao L (2022) Ai in finance: challenges, techniques, and opportunities. ACM Comput Surv (CSUR) 55(3):1–38

Cao C, Hansch O, Wang X (2008) Order placement strategies in a pure limit order book market. J Financ Res 31(2):113–140

Cao C, Hansch O, Wang X (2009) The information content of an open limit-order book. J Futur Mark: Futur, Opt, Other Deriv Prod 29(1):16–41. Accessed 6 Mar 2024

Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078

Coletta A, Moulin A, Vyetrenko S, Balch T (2022) Learning to simulate realistic limit order book markets from data as a world agent. In: Proceedings of the Third ACM International Conference on AI in Finance, pp. 428–436

Coletta A, Prata M, Conti M, Mercanti E, Bartolini N, Moulin A, Vyetrenko S, Balch T (2022) Towards realistic market simulations: A generative adversarial networks approach. In: Proceedings of the Second ACM International Conference on AI in Finance (ICAIF), New York, NY, USA. https://doi.org/10.1145/3490354.3494411

Comiter M (2019) Attacking artificial intelligence. Belfer Center Paper 8:2019–08

Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805

Duong HN, Kalev PS (2014) Anonymity and the information content of the limit order book. J Int Financ Mark, Inst Money 30:205–219

Engle RF, Ghysels E, Sohn B (2013) Stock market volatility and macroeconomic fundamentals. Rev Econ Stat 95(3):776–797

Fei Y, Zhou Y (2023) Intelligent prediction model of shanghai composite index based on technical indicators and big data analysis. Highlights Bus, Econ Manag 17:370–389

Gomber P, Haferkorn M (2015) High frequency trading. Encyclopedia of Information Science and Technology, Third Edition, 1–9

Grinsztajn L, Oyallon E, Varoquaux G (2022) Why do tree-based models still outperform deep learning on typical tabular data? In: Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track

Gundersen OE, Kjensmo S (2018) State of the art: Reproducibility in artificial intelligence. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32

Guo Y, Chen X (2022) Forecasting the mid-price movements with high-frequency lob: a dual-stage temporal attention-based deep learning architecture. Arab J Sci Eng 48:9597–9618

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778

Hu Z, Zhao Y, Khushi M (2021) A survey of forex and stock price prediction using deep learning. Appl Syst Innov 4(1):9

Huang RD, Stoll HR (1994) Market microstructure and stock return predictions. Rev Financ Stud 7(1):179–213

Jiang W (2021) Applications of deep learning in stock market prediction: recent progress. Expert Syst Appl 184:115537

Jin Z, Yang Y, Liu Y (2020) Stock closing price prediction based on sentiment analysis and lstm. Neural Comput Appl 32:9713–9729

Kisiel D, Gorse D (2022) Axial-lob: High-frequency trading with axial attention. arXiv preprint arXiv:2212.01807

Kumbure MM, Lohrmann C, Luukka P, Porras J (2022) Machine learning techniques and data for stock market forecasting: a literature review. Expert Sys Appl 197:116659

LAI CY, CHEN R-C, Caraka R (2019) Prediction average stock price market using lstm. ir. lib. cyut. edu. tw

Li X, Xie H, Chen L, Wang J, Deng X (2014) News impact on stock price return via sentiment analysis. Knowl-Based Syst 69:14–23

Lim B, Zohren S (2021) Time-series forecasting with deep learning: a survey. Philos Trans R Soc A 379(2194):20200209

Liu X-Y, Xia Z, Rui J, Gao J, Yang H, Zhu M, Wang C, Wang Z, Guo J (2022) Finrl-meta: market environments and benchmarks for data-driven financial reinforcement learning. Adv Neural Inf Process Syst 35:1835–1849

LOBCAST. https://github.com/matteoprata/LOBCAST

Lucchese L, Pakkanen M, Veraart A (2022) The short-term predictability of returns in order book markets: a deep learning perspective. arXiv preprint arXiv:2211.13777

Luong M-T, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025

Mahmoud A, Mohammed A (2021) A survey on deep learning for time-series forecasting. Mach Learn Big Data Anal Paradig: Anal, Appl Chall. https://doi.org/10.1007/978-3-030-59338-4_19

Mintarya LN, Halim JN, Angie C, Achmad S, Kurniawan A (2023) Machine learning approaches in stock market prediction: a systematic literature review. Procedia Comput Sci 216:96–102

Mizuta T (2016) A brief review of recent artificial market simulation (agent-based model) studies for financial market regulations and/or rules. Available at SSRN 2710495

Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E (2015) Deep learning applications and challenges in big data analytics. J Big Data 2(1):1–21

Nasdaq: Stock Screener. https://www.nasdaq.com/market-activity/stocks/screener
Nguyen TH, Shirai K, Velcin J (2015) Sentiment analysis on social media for stock movement prediction. Expert Syst Appl 42(24):9603–9611
Nousi P, Tsantekidis A, Passalis N, Ntakaris A, Kanniainen J, Tefas A, Gabbouj M, Iosifidis A (2019) Machine learning for forecasting mid-price movements using limit order book data. IEEE Access 7:64722–64736. https://doi.org/10.1109/ACCESS.2019.2916793
Ntakaris A, Magris M, Kanniainen J, Gabbouj M, Iosifidis A (2018) Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. J Forecast 37(8):852–866
Nti IK, Adekoya AF, Weyori BA (2020) A systematic review of fundamental and technical analysis of stock market predictions. Artif Intel Rev 53(4):3007–3057
Olorunnimbe K, Viktor H (2023) Deep learning in the stock market-a systematic survey of practice, backtesting, and applications. Artif Intell Rev 56(3):2057–2109
Orimoloye LO, Sung M-C, Ma T, Johnson JE (2020) Comparing the effectiveness of deep feedforward neural networks and shallow architectures for predicting stock price indices. Expert Syst Appl 139:112828
Ozbayoglu AM, Gudelek MU, Sezer OB (2020) Deep learning for financial applications: a survey. Appl Soft Comput 93:106384
Pascual R, Veredas D (2003) What pieces of limit order book information do are informative? an empirical analysis of a pure order-driven market
Passalis N, Tefas A, Kanniainen J, Gabbouj M, Iosifidis A (2019) Deep adaptive input normalization for time series forecasting. IEEE Trans Neural Netw Learn Syst 31(9):3760–3765
Passalis N, Tefas A, Kanniainen J, Gabbouj M, Iosifidis A (2020) Temporal logistic neural bag-of-features for financial time series forecasting leveraging limit order book data. Pattern Recognit Lett 136:183–189
Passalis N, Tsantekidis A, Tefas A, Kanniainen J, Gabbouj M, Iosifidis A (2017) Time-series classification using neural bag-of-features. In: 2017 25th European Signal Processing Conference (EUSIPCO), pp. 301–305. IEEE
Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al (2019) Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems 32
Pineau J, Vincent-Lamarre P, Sinha K, Larivière V, Beygelzimer A, d'Alché-Buc F, Fox E, Larochelle H (2021) Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program). J Mach Learn Res 22(1):7459–7478
Ratto AP, Merello S, Oneto L, Ma Y, Malandri L, Cambria E (2018) Ensemble of technical analysis and machine learning for market trend prediction. In: 2018 IEEE Symposium Series on Computational Intelligence (ssci), pp. 2090–2096. IEEE. Accessed 6 Mar 2024
Ren R, Wu DD, Liu T (2018) Forecasting stock market movement direction using sentiment analysis and support vector machine. IEEE Syst J 13(1):760–770
Robledo Costales I (2023) Benefits and risks of using AI in trading. https://www.cityindex.com/en-uk/news-and-analysis/benefits-and-risks-of-ai/. Accessed 6 Mar 2024
Ruff L, Kauffmann JR, Vandermeulen RA, Montavon G, Samek W, Kloft M, Dietterich TG, Müller K-R (2021) A unifying review of deep and shallow anomaly detection. Proceedings of the IEEE 109(5):756–795
Rundo F, Trenta F, Stallo AL, Battiato S (2019) Machine learning for quantitative finance applications: a survey. Appl Sci 9(24):5574
Saha S, Gao J, Gerlach R (2021) Stock ranking prediction using list-wise approach and node embedding technique. IEEE Access 9:88981–88996
Sezer OB, Gudelek MU, Ozbayoglu AM (2020) Financial time series forecasting with deep learning: a systematic literature review: 2005–2019. Appl Soft Comput 90:106181
Shah D, Isah H, Zulkernine F (2019) Stock market analysis: a review and taxonomy of prediction techniques. Int J Financ Stud 7(2):26
Shah J, Vaidya D, Shah M (2022) A comprehensive review on multiple hybrid deep learning approaches for stock prediction. Intel Syst Appl 16:200111
Shi Z, Cartlidge J (2023) Neural stochastic agent-based limit order book simulation: A hybrid methodology. arXiv preprint arXiv:2303.00080
Shwartz-Ziv R, Armon A (2022) Tabular data: deep learning is not all you need. Inf Fusion 81:84–90
Silberg J, Manyika J (2019) Notes from the ai frontier: Tackling bias in ai (and in humans). McKinsey Global Institute 1(6)
Sirignano JA (2019) Deep learning for limit order books. Quant Finance 19(4):549–570

Song Q, Liu A, Yang SY (2017) Stock portfolio selection using learning-to-rank algorithms with news sentiment. Neurocomputing 264:20–28
Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. Advances in neural information processing systems 27
Torres JF, Hadjout D, Sebaa A, Martínez-Álvarez F, Troncoso A (2021) Deep learning for time series forecasting: a survey. Big Data 9(1):3–21
Tran DT, Iosifidis A, Kanniainen J, Gabbouj M (2018) Temporal attention-augmented bilinear network for financial time-series data analysis. IEEE Trans Neural Netw Learn Syst 30(5):1407–1418
Tran DT, Kanniainen J, Gabbouj M, Iosifidis A (2021) Data normalization for bilinear structures in high-frequency financial time-series. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 7287–7292. IEEE
Tran DT, Kanniainen J, Iosifidis A (2022) How informative is the order book beyond the best levels? machine learning perspective. arXiv preprint arXiv:2203.07922
Tran DT, Passalis N, Tefas A, Gabbouj M, Iosifidis A (2022) Attention-based neural bag-of-features learning for sequence data. IEEE Access 10:45542–45552
Tsantekidis A, Passalis N, Tefas A, Kanniainen J, Gabbouj M, Iosifidis A (2017) Using deep learning to detect price change indications in financial markets. In: 2017 25th European Signal Processing Conference (EUSIPCO), pp. 2511–2515. https://doi.org/10.23919/EUSIPCO.2017.8081663
Tsantekidis A, Passalis N, Tefas A, Kanniainen J, Gabbouj M, Iosifidis A (2017) Forecasting stock prices from the limit order book using convolutional neural networks. In: 2017 IEEE 19th Conference on Business Informatics (CBI), vol. 1, pp. 7–12. IEEE
Tsantekidis A, Passalis N, Tefas A, Kanniainen J, Gabbouj M, Iosifidis A (2020) Using deep learning for price prediction by exploiting stationary limit order book features. Appl Soft Comput 93:106401
Wallbridge J (2020) Transformers for limit order books. arXiv preprint arXiv:2003.00130
Wu Y, Mahfouz M, Magazzeni D, Veloso M (2021) How robust are limit order book representations under data perturbation? arXiv preprint arXiv:2110.04752
Wu Y, Mahfouz M, Magazzeni D, Veloso M (2022) Towards robust representation of limit orders books for deep learning models. arXiv preprint arXiv:2110.05479
Xu Y, Cohen SB (2018) Stock movement prediction from tweets and historical prices. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1970–1979
Zaznov I, Kunkel J, Dufour A, Badii A (2022) Predicting stock price changes based on the limit order book: a survey. Mathematics 10(8):1234
Zhang Z, Lim B, Zohren S (2021) Deep learning for market by order data. Appl Math Financ 28(1):79–95
Zhang Z, Zohren S, Roberts S (2019) Deeplob: deep convolutional neural networks for limit order books. IEEE Trans Signal Proc 67(11):3001–3012
Zhang Z, Zohren S (2021) Multi-horizon forecasting for limit order books: Novel deep learning approaches and hardware acceleration using intelligent processing units. arXiv preprint arXiv:2105.10430