Prediction of US 30-years-treasury-bonds mouvement and trading entry point using Robust 1DCNN-BiLSTM-XGBoost algorithm

Abdellah EL ZAAR[1], Nabil BENAYA[1], Toufik BAKIR[2], Amine MANSOURI[2], and Abderrahim EL ALLATI[1]

[1]Universite Abdelmalek Essaadi Departement de Physique
[2]Universite de Bourgogne

April 6, 2023

Abstract

This paper proposes a novel algorithm that accurately predicts market trends and trading entry points for US 30-year-treasury bonds using a hybrid approach of 1-Dimensional Convolutional Neural Network (1DCNN), Long-Short Term Memory (LSTM), and XGBoost algorithms. We compared the performance of various strategies using 1DCNN and LSTM and found that existing state-of-the-art methods based on LSTM have excellent results in market movement prediction tasks, but the effectiveness of 1DCNN and LSTM in terms of trading entry point and market perturbations has not been studied thoroughly. We demonstrate, through experiments that our proposed 1DCNN-BiLSTM-XGBoost algorithm combined with moving averages crossover effectively mitigates noise and market perturbations, leading to high accuracy in spotting trading entry points and trend signals for US 30-year-treasury-bonds. Our experimental study shows that the proposed approach achieves an average of 0.0001% Root Mean Squared Error and 100% R-Square, making it a promising method for predicting the market trends and trading entry points.

Prediction of US 30-years-treasury-bonds mouvement and trading entry point using Robust 1DCNN-BiLSTM-XGBoost algorithm

Abdellah EL ZAAR[a,*], Nabil BENAYA[a], Toufik BAKIR[b], Amine MANSOURI[b], Abderrahim EL ALLATI[a]

[a]*Laboratory of R&D in Engineering Sciences, FST Al-Hoceima, Abdelmalek Essaadi University, Tetouan 93000, Morocco.*
[b]*IMVIA Laboratory, University of Burgundy, Dijon 21078, France.*

Abstract

This paper proposes a novel algorithm that accurately predicts market trends and trading entry points for US 30-year-treasury bonds using a hybrid approach of 1-Dimensional Convolutional Neural Network (1DCNN), Long-Short Term Memory (LSTM), and XGBoost algorithms. We compared the performance of various strategies using 1DCNN and LSTM and found that existing state-of-the-art methods based on LSTM have excellent results in market movement prediction tasks, but the effectiveness of 1DCNN and LSTM in terms of trading entry point and market perturbations has not been studied thoroughly. We demonstrate, through experiments that our proposed 1DCNN-BiLSTM-XGBoost algorithm combined with moving averages crossover effectively mitigates noise and market perturbations, leading to high accuracy in spotting trading entry points and trend signals for US 30-year-treasury-bonds. Our experimental study shows that the proposed approach achieves an average of 0.0001% Root Mean Squared Error and 100% R-Square, making it a promising method for predicting the market trends and trading entry points.

Keywords: 1-Dimensional Convolutional Neural Network (1DCNN), time series forecasting, Long-Short Term Memory (LSTM), Signal trend prediction.

1. Introduction

Financial time series forecasting using deep learning techniques has attained great success. These techniques refer to using historical data to predict trends and signals across various financial markets, including bonds, forex, indices, and cryptocurrencies. The effectiveness of predictive models can significantly impact profitability by generating high cumulative returns. Presently, state-of-the-art methods can be classified into two categories: traditional machine learning techniques, as seen in the works of P Arora *et al*. [1], MN Ashtiani *et al*. [2], and deep learning-based methods presented in recent research such as work done by GI Kim *et al*. [3], HV Dudukcu *et al*. [4], and C Zhong *et al*. [5]. The Long-Short Term Memory (LSTM) algorithm has been developed and utilized to forecast financial market movements with remarkable success. The primary contrast between recurrent neural networks (RNNs) and LSTMs is that LSTMs are an advancement of RNNs that enhances their memory. LSTMs exhibit superior performance in retaining long-term dependencies in the data compared to RNNs, accomplished by utilizing special memory cells and gate functions [6].

Recent studies have investigated the robustness of LSTMs by modifying memory cell structures to enhance prediction efficiency [7]. The Bidirectional LSTM (BiLSTM) comprises two LSTMs: the first processes data in a left-to-right sequence, and the second processes in a right-to-left sequence. The difference between BiLSTM and LSTM lies in the fact that the output of an LSTM network relies on the data processed in all previous iterations, whereas the BiLSTM processes current data in a sequence while also considering previous and upcoming data.

1-dimensional Convolutional Neural Networks (1DCNNs) are utilized to predict financial market movements and demonstrate their robustness in handling one-dimensional data. In current research, Convolutional Neural Networks exhibit their ability to learn dependencies accurately for three-dimensional data using convolution and pooling functions. Additionally, this powerful algorithm can effectively learn dependencies for one-dimensional data, which is the case for time series.

Prevailing volatility and noise in financial markets are two significant factors that hinder the effectiveness of the prediction process, ultimately affecting the cumulative profits and returns. The current financial time series prediction methods are highly sensitive to these perturbations. Traditional LSTMs and BiLSTMs may fail in predicting the financial market trend under such conditions. However, successful predictions require strong feature and data structure engineering, optimized architecture, and in-depth knowledge of financial market technical analysis. Technical analysis involves studying and anticipating financial market movements based on historical charts, statistical and visual indicators, mathematical functions, and more. A good technical analysis approach can lead to highly accurate market movement predictions.

Apart from technical analysis, traders and investors must

*Corresponding author
Email addresses: abdellah.elzaar@etu.uae.ac.ma (Abdellah EL ZAAR), nabil.benaya@gmail.com (Nabil BENAYA), toufik.bakir@u-bourgogne.fr (Toufik BAKIR), amine.Mansouri@u-bourgogne.fr (Amine MANSOURI), eabderrahim@uae.ac.ma (Abderrahim EL ALLATI)

have their own trading strategy. Through backtesting, traders can analyze the performance of their trading strategy and make necessary improvements. The basic reason why investors and traders lose a significant amount of capital is due to their inability to control their emotions. Discipline and concentration are crucial while trading financial markets. This is where machine learning and deep learning algorithms can offer a significant advantage in predicting financial market movements and identifying optimal trading entry points.

US 30-year Treasury bonds are issued by the US government and have a maturity period of 30 years from the date of issue. These bonds provide a fixed interest rate until maturity, after which the investor receives the full value of the bond. Due to the increased risk associated with a longer maturity, 30-year Treasury bonds tend to offer higher interest rates than shorter-term bonds. For our study, we utilized data from Contracts for Difference (CFDs) on US 30-year Treasury bonds. CFDs are derivative financial instruments that allow buyers and sellers to exchange the price difference of an underlying asset, such as a stock, bond or commodity.

In this paper, we present a novel hybrid approach for predicting the movement of US 30-year-treasury-bonds CFDs and identifying optimal trading entry points. Our proposed method involves combining 1DCNN, Bidirectional LSTMs, and Xgboost algorithms. To enhance the effectiveness of our model, we utilize a combination of statistical indicators and previous close prices as features for our dataset. Moreover, we introduce a new algorithm that can accurately predict the ideal trading entry point. Our experimental results demonstrate the robustness and accuracy of our approach, with stable predicted results for both market trends and trading entry points. Additionally, our approach shows promising profitability, with the predicted entry points yielding significant cumulative returns.

The rest of this paper is organized as follows: section 2 presents works done recently in the field of financial time series forecasting using Deep Learning. Section **??** describes the the process we used the prepare the financial time series data and features engeneering. In section 4 we explain our proposed method. Experiments and achieved results are discussed in section 5.1. We give an overview about profitability using our proposed approach in section 5.2, Section 6 is dedicated to conclusion and future works.

2. Related work

Most works done in the field of financial time series forecasting uses historical data to predict only the market trend. Historical data contains the Open, High, Low and Close prices of a specific time period. This data is used to perform the technical analysis and train machine learning models. Financial time series forecasting requires a strong and careful features engineering to predict with high accuracy the market trend. Moreover, it is necessary to study the market behavior and implement the appropriate algorithm to obtain an accurate and effective results. The existing state-of-the-art works provide satisfactory results concerning the prediction of the market movement. However, market trend prediction with respect to the trading entry point and market perturbations have not been studied yet. Several machine learning and deep learning methods have been used to predict financial markets movement. For example, S Alonso-Monsalve *et al*. [8], implemented LSTM and CNN architectures to predict the trend of Bitcoin, Dash, Ether, Litecoin, Monero and Ripple cryptocurrency exchange. The proposed approaches were able to provide good results specially for Bitcoin, Etherium and Litecoin cryptocurrencies. In [9], M Poongodi *et al*. implemented Support Vector Machine (SVM) algorithm to predict the price of Ethereum blockchain cryptocurrency in an industrial finance system. When applying their proposed model, the Support Vector Machine (SVM) method produced significantly higher accuracy (96.06%) compared to Linear Regression, which achieved an accuracy of only 85.46%. This can be explained by the SVM ability to classify 1-Dimentional data. T Shintate *et al*. [10] provided a trend prediction classification framework named the Random Sampling Method (RSM) for cryptocurrency time series that are non-stationary. Their proposed approach shows strong results and outperforms those based on LSTM. Recurrent Neural Networks still also used to time predict financial time series movement. In [11], A Lazcano *et al*. proposed a combined model based on Recurrent Neural Networks and Graph Convolutional Networks for crude oil prices prediction. Their proposed approach uses two pre-trained models on crude oil prices and a third model which receives the input and make the predictions. In term of accuracy their proposed method achieved better results compared with ARIMA and PROPHET models. The work done by S Zaheer *et al*. [12] develops a hybrid deep-learning forecasting model. The model takes the input stock data and forecasts two stock parameters close price and high price for the next day. Their proposed method based on CNN-LSTM-RNN has the lowest Mean Absolute Error (MAE) Compared with CNN-LSTM. The main problem of financial time series movement prediction models is that they don't provide effictive performance in the long term. When using these models in predicting trading entry point, the investors risks to lose their capital.

3. Data preparation and feature engineering

Financial market data is a critical source of information for traders, researchers, and institutions, providing live variations of trade data, including financial information such as price, bid-ask quotes, and market volume. Trading platforms distribute reports on various financial assets and instruments to assist traders and firms in making informed decisions. Historical market data is used to analyze the market behavior and movement over a specific time interval. Trading platforms offer visual graph charts with several indicators that can be used to interpret market price action accurately. A successful market interpretation and analysis can lead to accurate entry and exit trading positions. Several graph charts represent market movement and price action, such as bar charts, line charts, point and figure, market profile, and candlestick charts. This article will focus on candlestick charts, as they provide interesting visual indicators compared to other chart types, as shown in Fig 1.

Figure 1: candlestick chart.

The candlestick bars are characterized by two main colors, red and green, as well as different patterns. These colors and patterns are used to analyze the market movement during specific time periods. As shown in Fig 2, each candlestick provides four price points: Open, High, Low, and Close. The red color indicates that the open price was higher than the close price, and the price is moving down. Conversely, the green color indicates that the open price was lower than the close price, and the price is moving up.

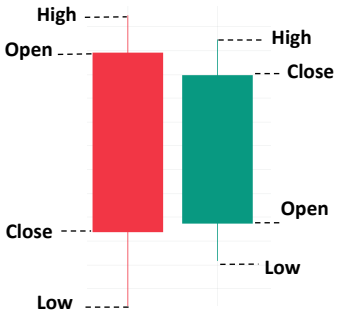


Figure 2: candlestick price levels.

To predict price movement, we exclusively utilize the close prices. To accurately identify the entry point for trading, we combine two moving averages, the Simple Moving Average (SMA) and Hull Moving Average (HMA). Through extensive analysis of historical data for US 30-year-treasury-bonds, we determined that the market responds to the crossover of HMA (with a period of 171 candlestick bars) and SMA (with a period of 19 candlestick bars). The Simple Moving Average is calculated by averaging the prices over a specified period, as follows:

$$SMA_p = \frac{1}{p} \sum_{n-p+1}^{n} (C_i) \tag{1}$$

Where C_i is the current period close price and P the number of calculation periods.

Hull Moving Average (HMA) is calculated using Weighted Moving Average (WMA) periods. Weighted Moving Average (WMA) is calculated by multiplying each one of the closing prices C_i within the considered series, by a certain weight coefficient W_i:

$$WMA_p = \frac{\sum_{i=1}^{P}(W_i \times C_i)}{\sum_{i=1}^{P} W_i} \tag{2}$$

Finally, we can obtain the Hull Moving Average (HMA) by the following formula:

$$\begin{aligned} \mathrm{HMA} = \mathrm{WMA}\Bigg(2 \times \mathrm{WMA}\left(\frac{\mathrm{close}}{2}, \frac{\mathrm{period}}{2}\right) \\ -\mathrm{WMA}(\mathrm{close}, \mathrm{period}), \sqrt{\mathrm{period}}\Bigg) \end{aligned} \tag{3}$$

The direction of the market changes when the Simple Moving Average (SMA) crosses the Hull Moving Average (HMA), as shown in Fig 3. A bullish crossover occurs when the SMA crosses up the HMA, indicating a potential buy entry point, while a bearish crossover occurs when the SMA crosses down the HMA, indicating a potential sell entry point.

Figure 3: bullish and bearish crossover.

To train our proposed model, we used the close prices shown in Fig 4 for 15-minute time-frame candlesticks from October 2022 to February 2023. The final dataset is presented in Table 1.

Figure 4: UST30Y 15min close prices from 10/2022 to 02/2023.

4. 1DCNN-BiLSTM-Xgboost algorithm

The proposed algorithm for predicting trends in US 30-year-treasury-bonds is a fusion of three powerful deep learning techniques: 1DCNN, bidirectional Long-Short Term Memory, and Xgboost algorithms. Our dataset comprises close prices of US

Table 1: Dataset.

	time	close	close_HMA_171	Close_SMA_19
182	2022-10-12 23:30:00	124.97	124.512603	124.853158
183	2022-10-12 23:45:00	125.03	124.531110	124.874211
184	2022-10-13 01:00:00	125.09	124.549994	124.883158
185	2022-10-13 01:15:00	125.03	124.568751	124.892105
186	2022-10-13 01:30:00	124.97	124.587116	124.906316
187	2022-10-13 01:45:00	125.00	124.605247	124.929474
188	2022-10-13 02:00:00	124.95	124.622900	124.952105
189	2022-10-13 02:15:00	124.94	124.640028	124.970526
190	2022-10-13 02:30:00	124.97	124.656732	124.990000
191	2022-10-13 02:45:00	124.94	124.672856	124.979474
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
8476	2023-02-20 16:45:00	125.78	125.782036	125.638421
8477	2023-02-20 17:00:00	125.78	125.779820	125.648421

30-year treasury bonds at 15-minute intervals, which is first preprocessed by the BiLSTM algorithm. The output vector generated by the BiLSTM is then fed into a 1-dimensional convolutional layer for further feature extraction. Finally, the XGBoost algorithm takes the resulting feature map as input to achieve accurate trend prediction. The following equation represents the computation in a bidirectional LSTM (biLSTM) neural network.

$$\begin{aligned} \overrightarrow{y_t} &= \text{LSTM}(x_t, \overrightarrow{y_{t-1}}) \\ \overleftarrow{y_t} &= \text{LSTM}(x_t, \overleftarrow{y_{t+1}}) \\ y_t &= [\overrightarrow{y_t}, \overleftarrow{y_t}] \end{aligned} \tag{4}$$

At each time step t, the input x_t is processed by two separate LSTM cells (as indecated in Fig 5), one in the forward direction (represented by the $\overrightarrow{y_t}$ vector) and one in the backward direction (represented by the $\overleftarrow{y_t}$ vector). The final hidden state at time t is the concatenation of the forward and backward hidden states, represented by the y_t vector. We decided to combine the 1DCNN with the bidirectional Long-Short Term Memory (biLSTM) algorithm due to its exceptional ability to capture both local patterns and reduce noise. Traditional linear models often struggle to capture the non-linear relationships present in financial time series data, making it difficult to generate accurate predictions. However, the 1DCNN with its parameter sharing and convolution functions proves to be highly effective in handling the noise typically present in financial time series. The 1-dimensional convolutional layer $\text{Conv}(y_t)$ for the resulting y_t vector can be modeled as follows:

$$\text{Conv}(y_t) = f\left(\sum_{i=1}^{Z} w_i y_{t+i-1} + b\right) \tag{5}$$

Where f is the activation function (e.g. ReLU), w_i is the weight of the i-th filter, Z is the filter size, and b is the bias term.

Finally the Xgboost algorithm is used for the prediction. XGBoost (short for eXtreme Gradient Boosting) is a supervised learning algorithm that belongs to the family of boosting algorithms. It is used for classification and regression tasks. The mathematical equation for XGBoost can be written as follows:

$$v_i = \sum_{k=1}^{K} f_k(x_i) = \sum_{k=1}^{K} w_{q(x_i,k)} \tag{6}$$

where v_i is the predicted target variable for the i^{th} data point, K is the number of trees, $f_k(x_i)$ is the prediction of the k^{th} tree for the i^{th} data point, w_j is the predicted weight for the j^{th} leaf, and $q(x_i,k)$ is the index of the leaf in which the i^{th} data point falls for the k^{th} tree. The objective function for XGBoost is given by:

$$Obj = \sum_{i=1}^{n} \mathscr{L}(v_i, \hat{v_i}) + \sum_{k=1}^{K} \Omega(f_k) \tag{7}$$

Where $\mathscr{L}$ is the loss function, $\hat{v_i}$ is the predicted target variable for the i^{th} data point, $\Omega(f_k)$ is the regularization term that penalizes the complexity of the k^{th} tree, and n is the number of data points. The output of the XGBoost model that takes the convolutional output $\text{Conv}(y_t)$ as input, can be expressed by the following equation:

$$\hat{y}t = g\left(\sum_{j=1}^{M} \gamma_j h_j + \sum_{k=1}^{K'} \beta_k \text{Conv}(y_{t-k})\right) \tag{8}$$

Where $\hat{y}t$ is the predicted output for time step t, h_j are the input features, M is the number of input features, γ_j are the corresponding feature weights, g is the sigmoid function, K' is the number of time steps used as input for the XGBoost model, β_k are the corresponding time-dependent weights, and $\text{Conv}(y_{t-k})$ is the convolutional output for time step $t-k$.

This equation combines the input features with the convolutional output to make a prediction for the output at each time step. The overall system is illustrated in Fig 5.

5. Experiments, results and profitability

5.1. Experiments and results

The experiments were carried out on a system with a six-core processor and 56GB of RAM. Python was the programming language used for implementing the algorithms that processed the collected dataset (as described in Section **??**). After constructing and training the model architecture, we evaluated its effectiveness and accuracy in predicting the closing prices and trends of US 30-year-treasury bonds. We analyzed its performance using the Root Mean Square Error (RMSE) and R-squared metrics. These metrics were used to compare the predicted movement of the market with the actual movement. The RMSE is a measure of the average difference between the predicted close prices ($\hat{C_i}$) and the reel close prices (C_i) of the US 30-year-treasury-bonds. To calculate the RMSE, we first obtain the squared differences between the predicted and reel values,

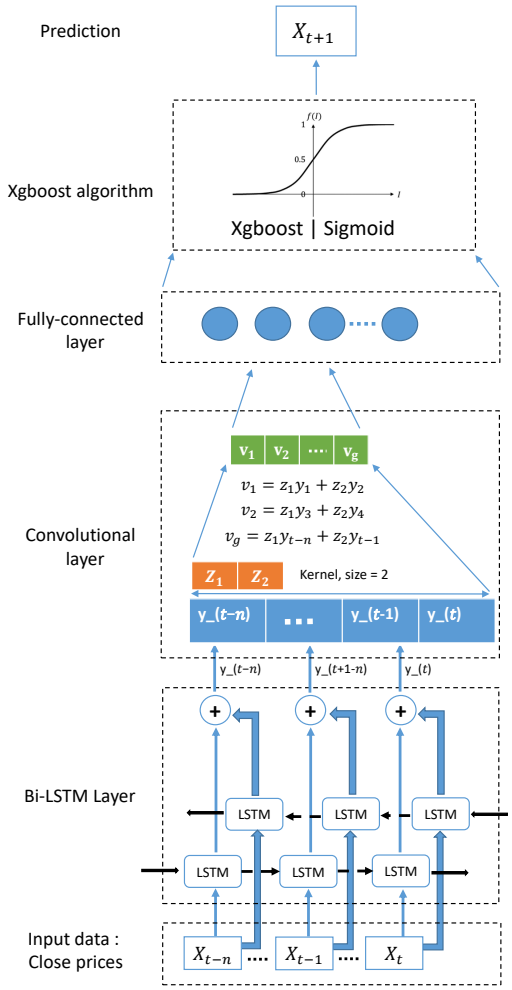


Figure 5: 1DCNN-BiLSTM-Xgboost algorithm

and then take the square root of their average. This is shown in equation 9. The RMSE provides a single value that represents the overall accuracy of the model, with lower values indicating better performance. In other side, the R-squared measures the proportion of the variance in the actual close prices that is explained by the predicted close prices. It is calculated as the ratio of the explained variance to the total variance. When the Rsquared value is higher, it means that the model is capable of explaining a larger proportion of the variability in the dependent variable. This indicates that the model fits the data better, providing a more accurate representation of the relationship between the dependent variable and the independent variables.

$$RMSE = \sqrt{\sum_{i=1}^{n}\left(\frac{(\hat{C}_i - C_i)^2}{n}\right)} \tag{9}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{C}_i - C_i)^2}{\sum_{i=1}^{n}(C_i - \bar{C}_i)^2} \tag{10}$$

To conduct a comprehensive comparative study, we evaluated the performance of several models, including the 1DCNN, BiLSTM, combined LSTM-1DCNN, and combined BiLSTM-1DCNN. In our study, we utilized the 1DCNN and the LSTM as baseline models.

The figures presented in Figs. 6 and 7 clearly show that the LSTM model outperforms the 1DCNN model in terms of performance. This can be attributed to the LSTM's ability to learn both short-term and long-term dependencies, thanks to its memory cells and gates. Although the 1DCNN model also demonstrates its ability to predict the US 30-year-treasury bonds, it is not as effective when dealing with noisy data. When comparing BiLSTM to LSTM, both algorithms demonstrate their effectiveness in predicting the market trend, as evidenced in Figs. 7 and 8. However, when it comes to RMSE and R-square, the BiLSTM outperforms the LSTM by providing a lower RMSE. This improvement can be attributed to the increased number of cells and gates in the BiLSTM architecture. To enhance the prediction process, we decided to combine the BiLSTM algorithm with the 1-DCNN. The combination of 1DCNN-BiLSTM has been found to be more accurate and effective, as shown in Figs 9 and 10 with an RMSE of 0.0004. To minimize errors and reduce noise, we incorporated the XGBoost algorithm as a classifier.

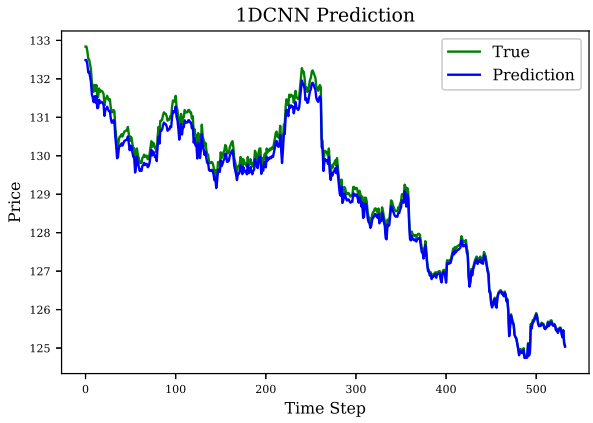
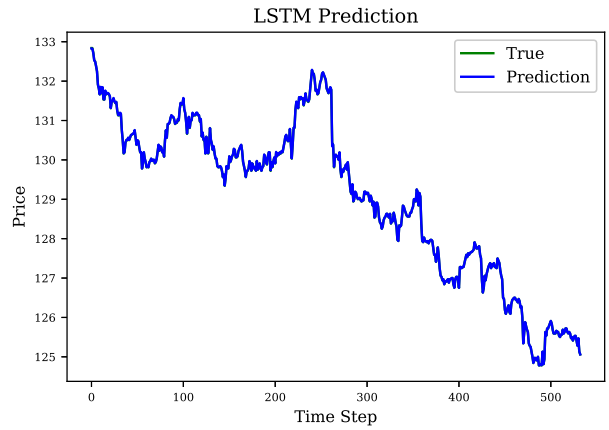


Figure 6: 1DCNN.

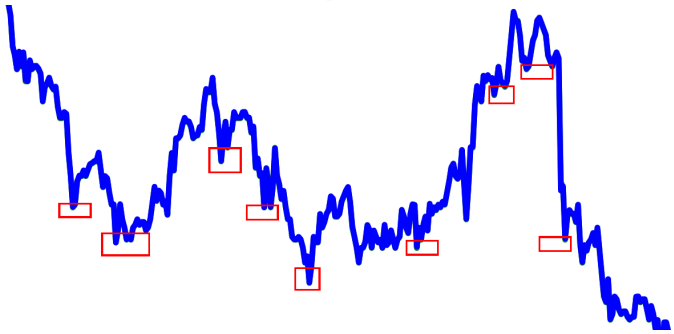
As demonstrated in Fig 5, the combination of these three algorithms yielded strong results in predicting the US 30-year-treasury-bonds market. The comparison in Table 2 reveals that the combination of the key characteristics of these algorithms provides a lower RMSE and a higher R-square than all other tested strategies. This technique has not been previously utilized to predict the US 30-year-treasury-bonds market and to identify trading entry points, making this work a novel contribution to the field.

5.2. *Trading entry point and profitability*

In this section, we present our proposed entry point trading algorithm and analyze the profit results based on our approach

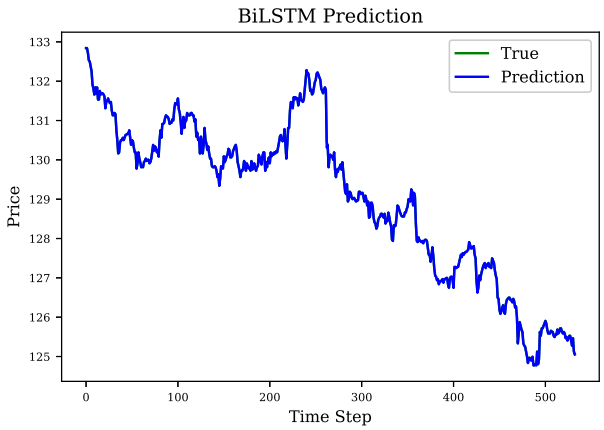


(a) LSTM prediction.

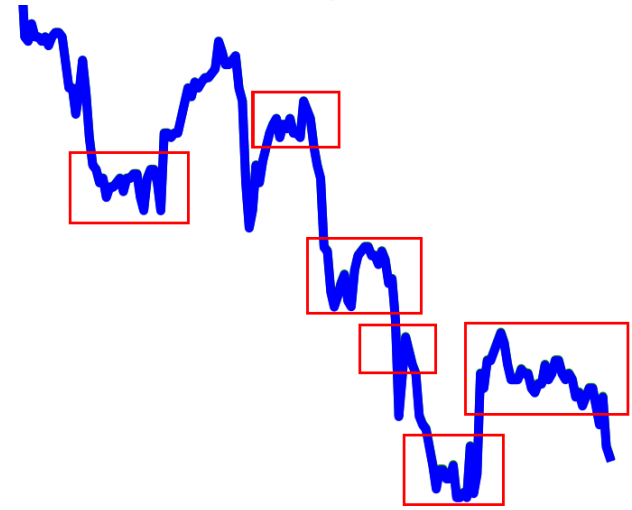


(b) LSTM error.

Figure 7: LSTM.



(a) BiLSTM prediction.



(b) BiLSTM error.

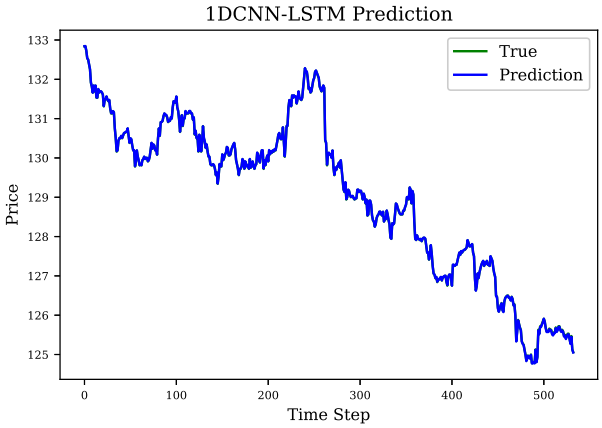
Figure 8: BiLSTM.

Table 2: Model performance

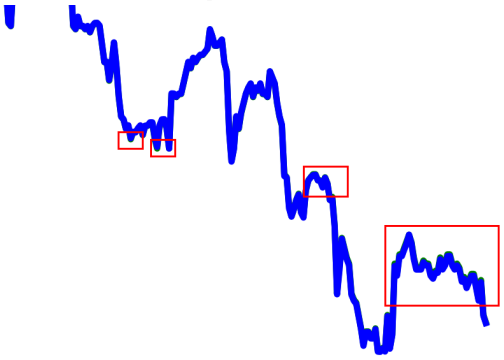
Approach	RMSE	R-square
1DCNN-BiLSTM-Xgboost (proposed method)	**0.0001%**	**100%**
1DCNN-BiLSTM	0.0004%	100%
1DCNN-LSTM	0.0008%	99.89%
BiLSTM	0.003%	99.60%
LSTM	0.01%	99.45%
1DCNN	0.19%	99.10%

predictions. As shown in Fig 12, our trading entry point algorithm combines data provided by the moving averages and predicted close prices. The algorithm can determine the beginning of the market trend using bullish and bearish crossovers with key periods (19 and 171) of the moving averages (HMA and SMA) as described in section **??**. Based on our approach's predicted prices and moving averages data, the trading entry point algorithm determines the type of trading order. A buy order is placed only if we have a bullish crossover and the predicted close price C_{n+1} is higher than the current close price C_n. Conversely, a sell order is placed only if we have a bearish crossover of the moving averages and the predicted close price C_{n+1} is lower than the current close price C_n. The method we use aims to capitalize on bullish and bearish trends while minimizing false breakouts to increase the accuracy of our approach. False breakouts refer to market movements that appear to signal a trend but ultimately fail to confirm it. By avoiding these false signals, our method can provide more reliable trading positions, which may result in higher profits. Therefore, the strategy involves carefully analyzing market trends and identifying genuine signals to make informed trading decisions.

To evaluate the effectiveness of our proposed approach, we assess the profitability of trading entry points. We compare our method with other prediction approaches discussed in section 5.1. To study our trading entry point algorithm, we conduct backtesting on a demo account with a capital of 1kdollar. We implement the algorithm (shown in Fig 12) using Python language and Metatrader5 libraries. The figure below (Fig 13) displays the cumulative profits generated by our proposed trading entry points algorithm, which utilizes the 1DCNN-BiLSTM-Xgboost approach, from February 1st, 2023 to March 17th, 2023. Additionally, we have included four other figures to present the profits generated by our algorithm using various deep learning models, including 1DCNN-BiLSTM, 1DCNN-LSTM, BiLSTM, LSTM, and 1DCNN, as shown in Figs 14, 15, 16, 17, and 18, respectively.

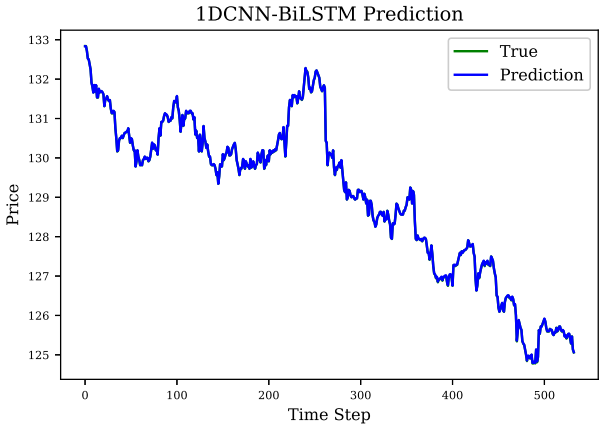


(a) 1DCNN-LSTM prediction.

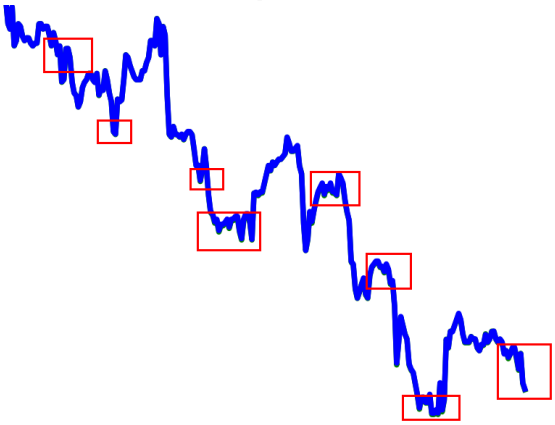


(b) 1DCNN-LSTM error.

Figure 9: 1DCNN-LSTM.



(a) 1DCNN-BiLSTM prediction.



(b) 1DCNN-BiLSTM error.

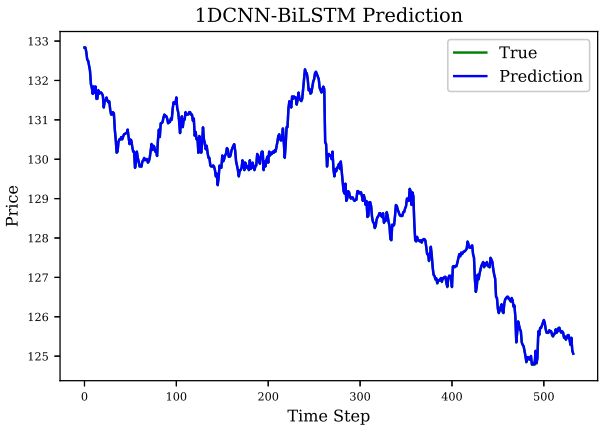
Figure 10: 1DCNN-BiLSTM.

Theses figures illustrate the cumulative returns of each proposed trading algorithm over a 45-day period. The 1DCNN-BiLSTM-Xgboost algorithm achieved a profit of 400 dollars, which is 40% of the initial capital. The 1DCNN-BiLSTM algorithm produced a profit of 375 dollars, equivalent to a return of 37.5% of the initial capital, while the 1DCNN-LSTM algorithm returned 340 dollars. On the other hand, the BiLSTM trained from scratch generated a profit of 319 dollars, and the LSTM algorithm provided satisfactory returns of 25% of the initial capital. In contrast, the 1DCNN did not produce effective results in terms of cumulative profits and resulted in a loss of -15%.

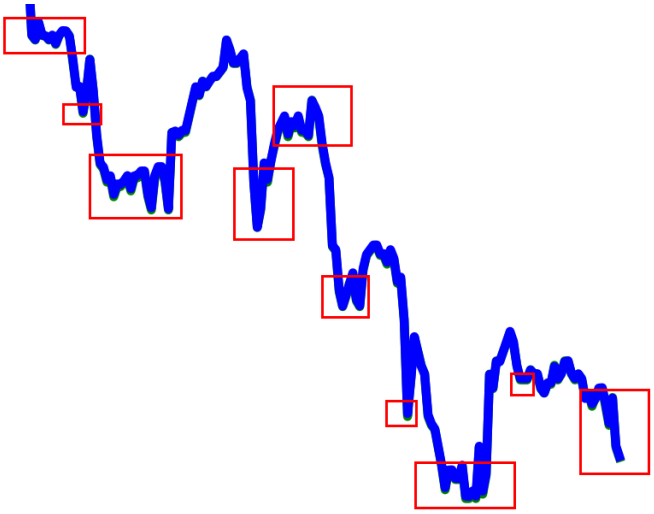
Table 3: Profitability Based on Predicted Entry Points using Proposed Approaches: 45-Day Backtesting Results with 1,000 dollars Account Balance

Proposed approach	Profits (USD)	Ppurcentage (%)
1DCNN-BiLSTM-Xgboost	**400**	**40%**
1DCNN-BiLSTM	375	37.5%
1DCNN-LSTM	340	34%
BiLSTM	319	31.9%
LSTM	250	25%
1DCNN	−150	−15%

Based on our analysis, we have concluded that training a 1DCNN from scratch does not produce satisfactory results on its own. However, when combined with the BiLSTM and Xgboost algorithm, the 1DCNN provides excellent results. Table 3 provides a summary of the profitability percentages achieved by each of the proposed algorithms.



(a) 1DCNN-BiLSTM-Xgboost prediction.



(b) 1DCNN-BiLSTM-Xgboost error.

Figure 11: 1DCNN-BiLSTM-Xgboost.

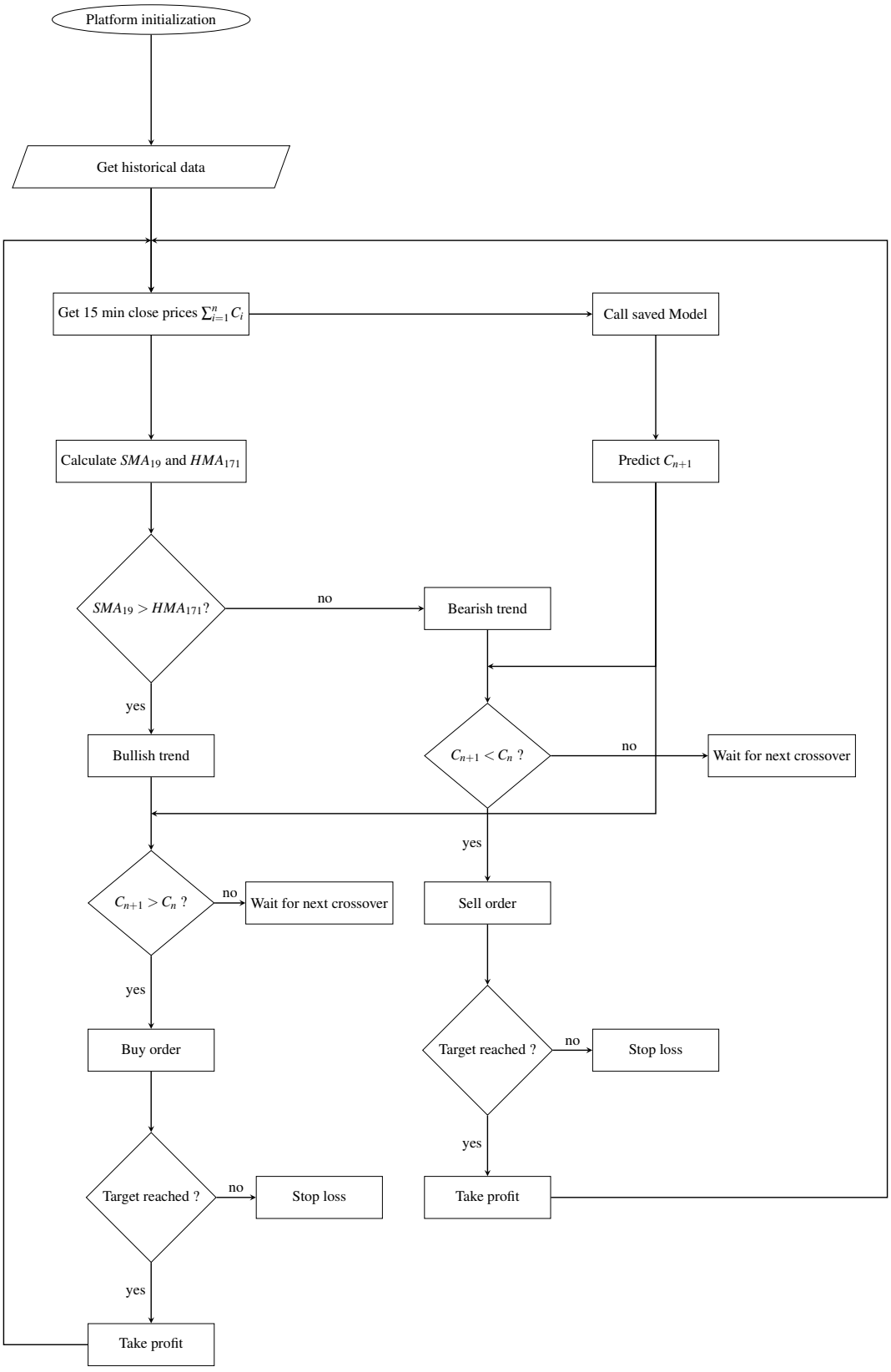


Figure 12: Trading entry point algorithm

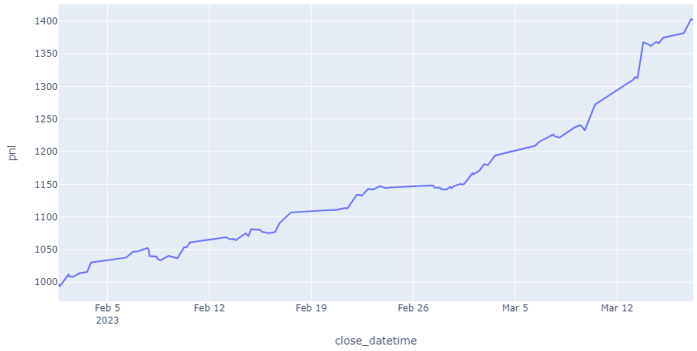


Figure 13: 1DCNN-BiLSTM-Xgboost profitability.

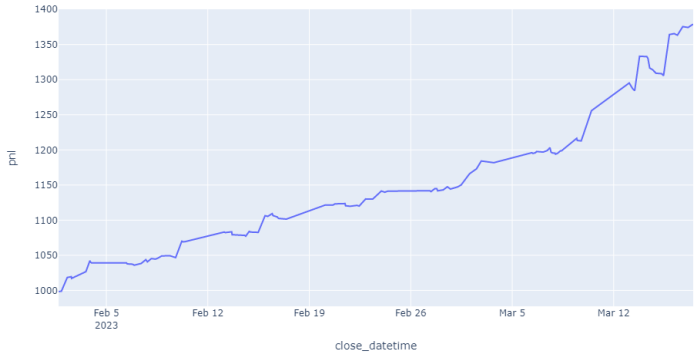


Figure 14: 1DCNN-BiLSTM profitability.

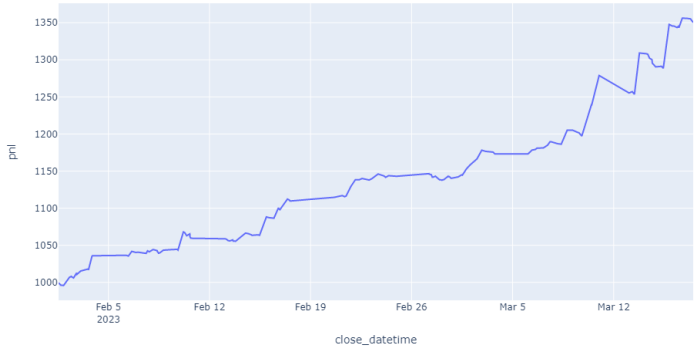


Figure 15: 1DCNN-LSTM profitability.

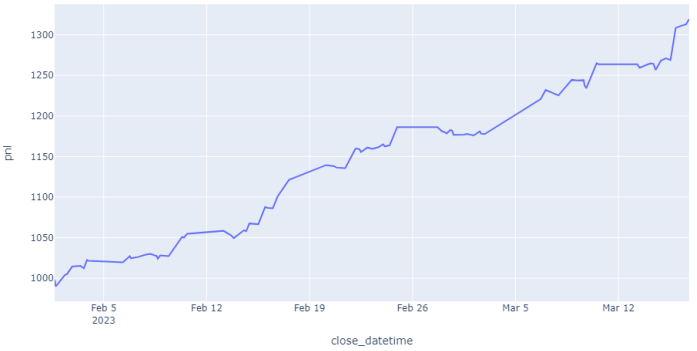


Figure 16: BiLSTM profitability.

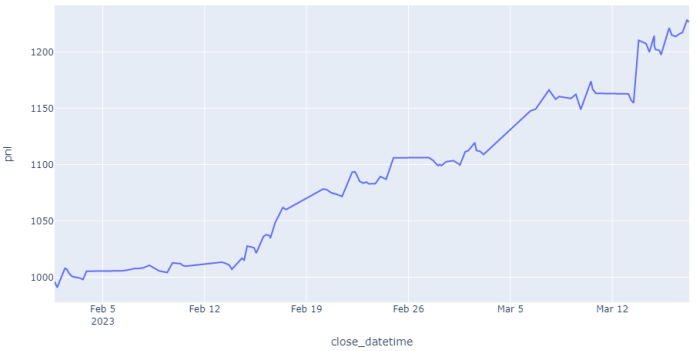


Figure 17: LSTM profitability.

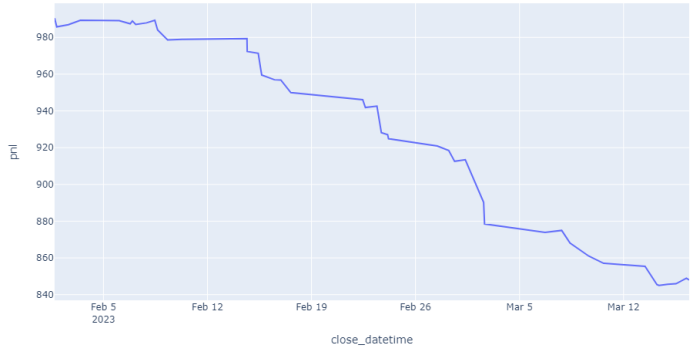


Figure 18: 1DCNN profitability.

6. Conclusion

In conclusion, the use of deep learning algorithms in predicting market movement and entry points has the potential to increase traders' profitability and to automate their trades. Emotions such as fear, doubt, and impulsiveness are common causes of capital loss in trading, but using a sound technical analysis strategy combined with deep learning algorithms can help reduce these risks. Our proposed approach combines the 1DCNN-BiLSTM-Xgboost algorithms with HMA(171) and SMA(19) moving averages to predict the movement and entry points of the US 30-year-treasury-bonds market, resulting in highly accurate results. While other algorithms, such as 1DCNN, have higher Root Mean Square Error (RMSE) and cannot deal with noisy datasets, the 1DCNN-BiLSTM-Xgboost algorithm provides a lower RMSE and a higher profitability of about 4% of the initial balance.

Furthermore, the proposed approach can be extended to find relationships between different financial markets and predict the most profitable positions during specific time intervals. It can also help identify real trends and avoid fake breakouts during trading. Overall, the combination of deep learning algorithms and technical analysis strategies has the potential to revolutionize trading and to increase profitability for traders.

7. Declaration of Competing Interest

I declare that I have no Conflict of Interest.

References

[1] P. Arora, A. Balyan, Comparative analysis of lstm, encoder-decoder and gru models for stock price prediction, in: Computational Intelligence: Select Proceedings of InCITe 2022, Springer, 2023, pp. 399–410.

[2] M. N. Ashtiani, B. Raahmei, News-based intelligent prediction of financial markets using text mining and machine learning: A systematic literature review, Expert Systems with Applications (2023) 119509.

[3] G. I. Kim, B. Jang, Petroleum price prediction with cnn-lstm and cnn-gru using skip-connection, Mathematics 11 (3) (2023) 547.

[4] H. V. Dudukcu, M. Taskiran, Z. G. C. Taskiran, T. Yildirim, Temporal convolutional networks with rnn approach for chaotic time series prediction, Applied Soft Computing 133 (2023) 109945.

[5] C. Zhong, W. Du, W. Xu, Q. Huang, Y. Zhao, M. Wang, Lstm-regat: A network-centric approach for cryptocurrency price trend prediction, Decision Support Systems (2023) 113955.

[6] N. I. Ab Kader, A review of long short-term memory approach for time series analysis and forecasting, in: Proceedings of the 2nd International Conference on Emerging Technologies and Intelligent Systems: ICETIS 2022, Volume 2, Vol. 573, Springer Nature, 2023, p. 12.

[7] S. M. Islam, N. Thakur, K. Garg, A. Gupta, A recent survey on lstm techniques for time-series data forecasting: Present state and future directions, Applications of Artificial Intelligence, Big Data and Internet of Things in Sustainable Development (2023) 123–132.

[8] S. Alonso-Monsalve, A. L. Suárez-Cetrulo, A. Cervantes, D. Quintana, Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators, Expert Systems with Applications 149 (2020) 113250.

[9] M. Poongodi, A. Sharma, V. Vijayakumar, V. Bhardwaj, A. P. Sharma, R. Iqbal, R. Kumar, Prediction of the price of ethereum blockchain cryptocurrency in an industrial finance system, Computers & Electrical Engineering 81 (2020) 106527.

[10] T. Shintate, L. Pichl, Trend prediction classification for high frequency bitcoin time series with deep learning, Journal of Risk and Financial Management 12 (1) (2019) 17.

[11] A. Lazcano, P. J. Herrera, M. Monge, A combined model based on recurrent neural networks and graph convolutional networks for financial time series forecasting, Mathematics 11 (1) (2023) 224.

[12] S. Zaheer, N. Anjum, S. Hussain, A. D. Algarni, J. Iqbal, S. Bourouis, S. S. Ullah, A multi parameter forecasting for stock time series data using lstm and deep learning model, Mathematics 11 (3) (2023) 590.