

Transformer Encoder and Multi-features Time2Vec for Financial Prediction

Nguyen Kim Hai Bui*, Nguyen Duy Chien*, Péter Kovács[†], and Gergő Bognár[†]

^{*}ELTE Eötvös Loránd University

[†]Department of Numerical Analysis, ELTE Eötvös Loránd University

{qmibhu, r0iftm, kovika}@inf.elte.hu, bognargergo@staff.elte.hu

Abstract—Financial prediction is a complex and challenging task of time series analysis and signal processing, expected to model both short-term fluctuations and long-term temporal dependencies. Transformers have remarkable success mostly in natural language processing using attention mechanism, which also influenced the time series community. The ability to capture both short and long-range dependencies helps to understand the financial market and to recognize price patterns, leading to successful applications of Transformers in stock prediction. Although, the previous research predominantly focuses on individual features and singular predictions, that limits the model's ability to understand broader market trends. In reality, within sectors such as finance and technology, companies belonging to the same industry often exhibit correlated stock price movements.

In this paper, we develop a novel neural network architecture by integrating Time2Vec with the Encoder of the Transformer model. Based on the study of different markets, we propose a novel correlation feature selection method. Through a comprehensive fine-tuning of multiple hyperparameters, we conduct a comparative analysis of our results against benchmark models. We conclude that our method outperforms other state-of-the-art encoding methods such as positional encoding, and we also conclude that selecting correlation features enhance the accuracy of predicting multiple stock prices.

Index Terms—time series analysis, financial prediction, neural networks, feature engineering

I. INTRODUCTION

Time series forecasting, especially in the financial sector, is a challenging, yet critically demanded and safety critical problem. Market prices are influenced by a multitude of factors, leading to complex investment and trading strategies that are required to model and manage multiple aspects [1]. In this paper, we focus on the particular problem of stock price prediction. The stock market can be viewed as a dynamic system, where stocks and further financial commodities (e.g. currencies, equities) are being traded. Traders generally prefer to purchase stocks of companies that show potential for future growth and avoid those expected to decline in value [2]. Companies within the same sector often exhibit similar correlations, as they tend to be affected similarly by macroeconomic events such as economic downturns or political policies. The most commonly used methods for predicting stock returns are broadly categorized into fundamental analysis and technical analysis. Fundamental analysis aims to determine the intrinsic

value of stocks based on internal and external economic indicators, that is particularly essential for investors focused on long-term trading. Technical analysis, being investigated in this paper, focuses on techniques to maximize the capital gain and decrease the loss, based on statistical analysis that is demanded to forecast the stock price [3]. Fast and accurate stock price forecasting is difficult, and this field is an active area of research where the best ways are yet to be found [2].

In principle, time series forecasting are expected to understand and model complex linear and non-linear interactions within historical data to predict the future. Traditional statistical approaches commonly adopt linear regression, exponential smoothing [4], and auto regression model [5], see e.g. the well-known Auto Regressive Integrated Moving Average (ARIMA) or Seasonal ARIMA (SARIMA) methods. With the advances in deep learning, recent works are heavily invested in ensemble models and sequence-to-sequence modeling such as recurrent neural networks (RNNs), and long short-term memory (LSTM) [6], in particular. However, the primary drawback of these methods is that the RNN family struggles to capture extremely long-term dependencies [7]. The recently popularized sequence-to-sequence model called Transformer [8] has achieved great success in natural language processing (NLP), especially in machine translation and large language models (LLMs), but also influenced other fields like signal processing and computer vision. Different from RNN-based models, transformers employ self-attention mechanism to learn the relationship among different positions globally. Motivated by the success of deep learning in image classification, natural language processing, and time series analysis, deep learning has become an emerging approach in the field of finance as well. Deep neural networks has the ability to learn low-dimensional representation extracted from originally high-dimensional input data. Machine and deep learning techniques have been found to be efficient in price forecasting and market trend prediction, see e.g. [9], that support investors to anticipate future trends and optimize stock investments.

This work investigates stock price forecasting with deep learning, offering multiple novel contributions. First, we suggest a multi-feature selection method based on analysis of multiple correlating stock prices, then we propose a novel neural network architecture involving Time2Vec [10] as encoding, combined with the Encoder of a Transformer. Time2Vec is a model-agnostic temporal encoding, leading towards model-

Supported by the University Excellence Fund of Eötvös Loránd University, Budapest, Hungary (ELTE). Nguyen Kim Hai Bui and Nguyen Duy Chien contributed equally to this work.

based deep learning [11], possibly offering transparent and interpretable decision-making.

II. BACKGROUND

A. Transformer

Transformers [8] are state-of-the-art sequence-to-sequence deep learning models that leverage the so-called self-attention mechanism, that allows the model to weigh the importance of each token in a sequence relative to others. By processing all tokens simultaneously, transformers can better understand context across the entire input, and capture long-range dependencies without the sequential limitations of RNNs. Furthermore, transformers show great ability to model interactions in sequential data, and thus they are appealing to time series modeling. Many variants of Transformer have been proposed to address special challenges in time series modeling and have been successfully applied to various time series tasks, such as forecasting, anomaly detection, and classification [12]. Usual transformer architectures involves Encoder and Decoder parts. Here, we focus on the Encoder only, that consists of multiple identical layers, each containing a multi-head self-attention mechanism and a feed-forward network, designed to process and understand input sequences by focusing on relationships within the data.

B. Time2Vec

Time2Vec [10] is an approach providing a model-agnostic vector representation for time. The input signal is decomposed in the time domain to $k + 1$ components of the form

$$\mathbf{t2v}(\tau)[i] = \begin{cases} \omega_i \tau + \varphi_i, & \text{if } i = 0, \\ \mathcal{F}(\omega_i \tau + \varphi_i), & \text{if } 1 \leq i \leq k, \end{cases}$$

where τ denotes the time instance, $\mathcal{F}$ is a periodic activation function, and ω_i and φ_i are learnable parameters. Note that if $\mathcal{F}$ is a sinusoid, then Time2Vec provides a time-invariant decomposition that encodes a temporal signal into a set of frequencies. The model was inspired by parts of positional encoding from [8], therefore it can be directly combined with the transformer architecture. In this sense, Time2Vec serves as an alternative to positional encoding that can also capture periodic behavior not supported by positional encoding.

C. Correlation Analysis

In order to identify repeating patterns, trends, and temporal dependencies within and between time series data, we examined their autocorrelation and cross-correlation. We considered the normalized cross-correlation of time series x and y as a function of lag k of the form

$$\rho_{xy}(k) = \frac{1}{\sigma_x \sigma_y} \sum_t (x_t - \bar{x})(y_{t+k} - \bar{y}),$$

where $\bar{x}$, $\bar{y}$, σ_x and σ_y denotes the means and the standard deviations of x and y , respectively. Autocorrelation of x is computed as ρ_{xx} , i.e. the cross-correlation with itself. In principle, correlation quantifies the linear relationship between the inputs. The values range between -1 and 1 due to the

normalization, where 1 , 0 , and -1 indicate perfect correlation, no correlation, and anticorrelation, respectively.

D. Feature aggregation

We investigated statistical aggregation methods to combine features from multiple time series to a single feature vector, which provides a simplified data representation that still benefits from the variations of multiple sources. Although the input time series are sampled at the same frequency, there are missing data points (treated as NaN), due to the size differences and missing information in the database. In order to resolve this obstacle, we used Geometric Mean Not NaN (GMNN), where we computed the geometric mean of different sources at each time point, ignoring NaN values.

III. METHODOLOGY

A. Motivation

The performance of a stock market predictor heavily depends on the correlations between historical data for training and the current input for prediction. Motivated by [13], we investigated autocorrelation and cross-correlation of multiple groups of stock indices to reveal short-term temporal dependencies and relations. As an example, Fig. 1 depicts the daily correlations using Exxon Mobil as base market. Similar to [13], we have found that the autocorrelation of the base market is mostly zero outside the origin, that suggests possible Markovian property. Consequently, we can presume that historical data offers limited insight into its future movements. On the other hand, different data sources such as Chevron provide a promising supporting feature due to the high correlation, and possible actual causation. Motivated by the strong correlation observed between the selected stock pairs, we propose a novel method to combine multiple stocks into a single input representation. By leveraging cross-correlation relationships, our approach aims to capture the joint dynamics of these correlated stocks, enabling the model to better understand their collective behavior and improve predictive accuracy.

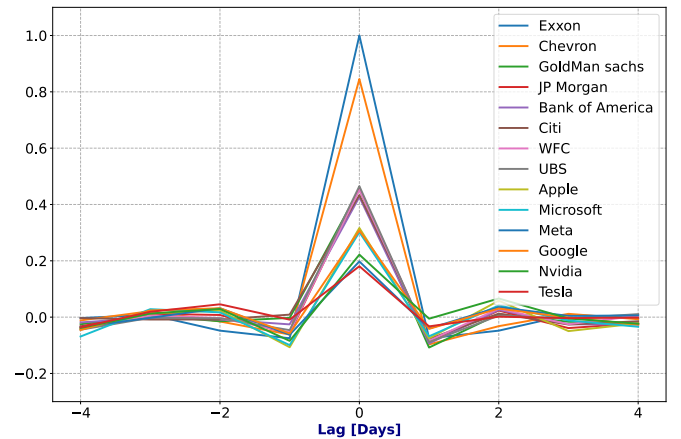


Fig. 1. Correlation analysis of stock indices: auto-correlation and cross-correlation of market trends compared to Exxon Mobil, inspired by [13].

B. Data Acquisition

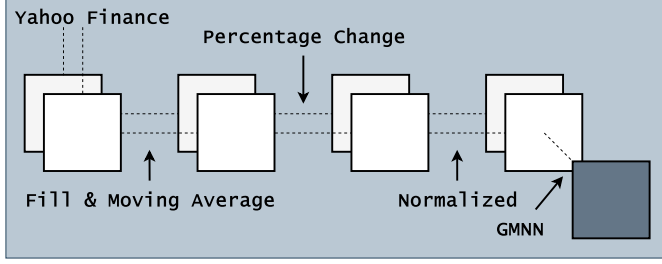


Fig. 2. The preprocessing pipeline. Steps: Fill and Moving Average, Percentage Change Computing, Normalization, Aggregation using GMNN.

Daily stock datasets are collected from their release date up to the 30th of September, 2024, using Yahoo Finance [14]. Based on the above motivation, we collected three groupings of two equities whose members significantly correlated to each other, such as **Group A:** NASDAQ (^IXIC), S&P500 (^GSPC); **Group B:** Exxon Mobil (XOM), Chevron (CVX); and **Group C:** Morgan Stanley (MS), Goldman Sachs (GS).

We propose the preprocessing and feature selection pipeline depicted on Fig. 2. First, missing data points are filled to create consistent and continuous inputs. Then, moving average is applied with a 14-day period in order to compensate for short-term fluctuations and produce smoother local trends. Next, percentage changes are computed for the next day, normalized between 0 and 1. Finally, once these steps are applied to each stock indices in the group, the processed data of the whole group is aggregated using GMNN to a single feature, that will serve as the final input for the model.

After the predictions, we applied postprocessing to retrieve back the close prices, as of Fig. 3. The output from the model is clipped in the range from 0 to 1 during the denormalization step. Then, the reverse percentage change and reverse moving average are computed to get desired close prices.

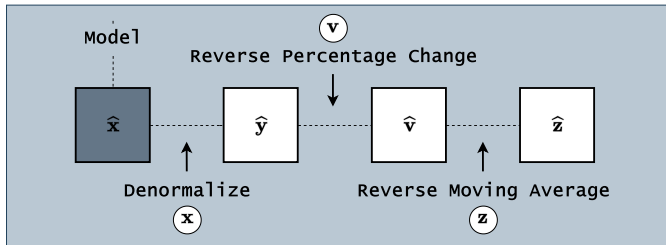


Fig. 3. The postprocessing pipeline. Steps: Denormalization, Reverse Percentage Change, Reverse Moving Average. x is the original normalized values, $\hat{x}$ is the predicted normalized values, $\hat{y}$ is the predicted percentage change values, v is the original moving average values, $\hat{v}$ is the predicted moving average values, z is the original close prices, $\hat{z}$ is the predicted close prices.

C. Proposed Model

Inspired by the inherent limitations of complex deep learning models, which are frequently trained on historically inconsistent and unreliable datasets, as well as the common data challenges across diverse industries such as technology,

finance, energy, and capital markets, this research aims to tackle critical methodological constraints in predictive analytics. We introduce a novel network architecture, presented in Fig. 4, that combines the Time2Vec and the Encoder of the Transformer. In detail, Time2Vec encodes positions and captures both linear and periodic behaviors of prices over time; the multiple Encoders perform multi-head self-attention, layer normalization, as well as positional-wise feed-forward; Residual connections are used to improve the behavior of the network which leads to a more valuable and powerful model [15]; Pooling reduces the spatial dimensions of the output from the Encoder, in terms of the number of features, while retaining the most important information; Dropout is an indispensable layer to avoid overfitting problem.

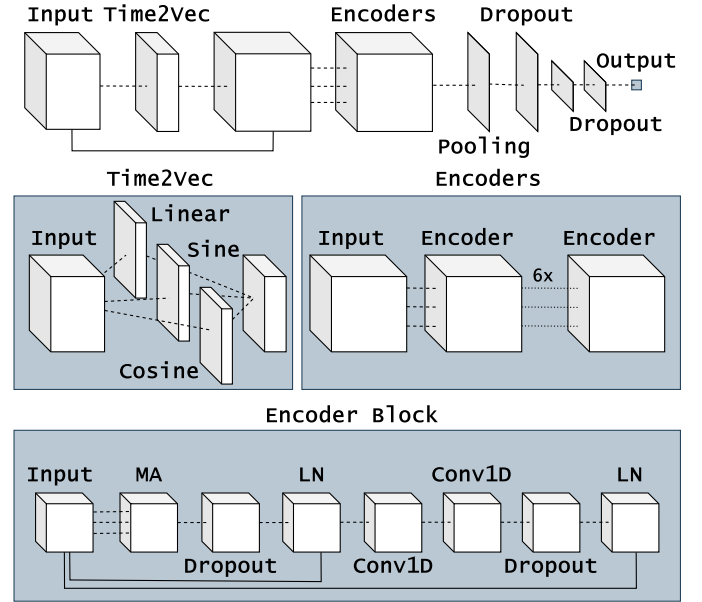


Fig. 4. The proposed model and architectures. Proposed model (top), Time2Vec module (left mid), Transformer Encoders module (right mid), Encoder Block (bottom). MA stands for Multi-head Attention, LN stands for Layer Normalization. The solid lines are residual connections.

Our design concepts were the to develop a robust architecture that can utilize the expression and generalization ability of the attention mechanism through Encoders, and to involve Time2Vec as encoding. Our investigations proved that Time2Vec is a fitting encoding scheme for the target application, and provides better ability to capture temporal patterns compared to the traditional approaches, as discussed in Section IV.

IV. RESULTS AND EVALUATION

We evaluated multiple models for comparison purposes, where in each case we trained the models on the first 80% of the data, validated on the next 10%, and the performance evaluations are made using the last 10%. Common evaluation metrics are used, such as root mean square error (RMSE), mean square error (MSE), mean absolute percentage error (MAPE), mean absolute error (MAE), R-square (R2).

In Table I, we compared the performance of single-feature models and the proposed multi-feature approach. Here, we used the following notations: models such as NAS, SP, EM, C, MS, and GS are single-feature models, meaning that we only use their target for training; on the other hand, NAS_SP, EM_C, and MS_GS are multi-feature models; they are trained by using the combined dataset (e.g., NASDAQ and S&P500 for NAS_SP). To evaluate models in each group, we rely on their target, for instance: NAS is a single-feature model with NASDAQ as a target, NAS_SP is a multi-feature model with NASDAQ as a target as well. Then, NAS_SP is compared to NS with respect to the same target, which is NASDAQ in this case. By this evaluation, we expect to figure out how well the cross-correlation affects the models' performance. We can conclude that the multi-feature models outperform single-feature models in 2 out of 3 cases, i.e. the feature selection benefits from the proposed feature aggregation. The only exception is the circumstance where the stock ages are young (i.e., their release date is about 10 to 20 years ago) and the cross-correlations are not tight enough. In addition, we can point out that the cross-correlation of NASDAQ and S&P500 is approximately 91%. Between Exxon Mobil and Chevron it is nearly 84% (see Fig. 1), and between Morgan Stanley and Goldman Sachs it is about 81%. Based on those clues, we conclude that the better the link between stocks, the better the multi-feature prediction.

TABLE I
EVALUATIONS BY TARGET OF EACH GROUP

Target	Model	RMSE	MSE	MAPE	MAE	R2
Group A						
^IXIC	NAS	0.0291	0.0008	3.1392	0.0187	0.8264
	NAS_SP	0.0248	0.0006	2.6785	0.0161	0.8743
^GSPC	SP	0.0164	0.0003	1.9422	0.0098	0.8766
	EM	0.0180	0.0003	2.0511	0.0104	0.8517
Group B						
XOM	EM	0.0350	0.0012	4.3509	0.0204	0.8163
	EM_C	0.0333	0.0011	4.0724	0.0200	0.8330
CVX	C	0.0275	0.0008	3.3767	0.0149	0.8241
	MS	0.0342	0.0012	3.7204	0.0172	0.7293
Group C						
MS	MS	0.0148	0.0002	1.7162	0.0105	0.8655
	MS_GS	0.0166	0.0003	2.0747	0.0128	0.8325
GS	GS	0.0221	0.0005	2.8797	0.0156	0.8545
	GS	0.0201	0.0004	2.6159	0.0144	0.8797

In Table II, we compared our model with different encoding techniques, and state-of-the-art deep learning predictor models, such as RNN and LSTM. Here, *TT2VFin* refers to our proposed multi-features model, *TT2VFin+p* adds positional encoding layer, *TT2VFin+M* uses masking in the attention mechanism, *TT2VFin+p+M* uses both positional encoding and masking. The results show that *TT2VFin* outperforms its variations, and *Transformer+p* is a Transformer Encoder using positional encoding instead of Time2Vec. As a more detailed analysis, we can see that applying a mask does not significantly affect *TT2VFin*; however, the incorporation of positional encoding to the proposed model reduces its performance. The

TABLE II
COMPARING TO STATE-OF-THE-ART MODELS

Target	Model	RMSE	MSE	MAPE	MAE	R2
Group A						
^IXIC	TT2VFin	0.0248	0.0006	2.6785	0.0161	0.8743
	TT2VFin+M	0.0293	0.0009	3.3222	0.0205	0.8240
	TT2VFin+p	0.1773	0.0314	27.500	0.1691	- 5.427
	TT2VFin+p+M	0.0337	0.0011	3.9642	0.0241	0.7685
	Transformer+p	0.0382	0.0015	4.4677	0.0279	0.7008
	RNN	0.0316	0.0010	3.6215	0.0224	0.7959
^GSPC	LSTM	0.0272	0.0007	2.9289	0.0176	0.8490
	TT2VFin	0.0164	0.0003	1.9422	0.0098	0.8766
	TT2VFin+M	0.0180	0.0003	2.2002	0.0109	0.8515
	TT2VFin+p	0.2265	0.0513	42.337	0.2253	- 22.51
	TT2VFin+p+M	0.0247	0.0006	3.6617	0.0189	0.7204
	Transformer+p	0.0207	0.0004	2.5934	0.0132	0.8024
XOM	RNN	0.0195	0.0004	2.3061	0.0116	0.8262
	LSTM	0.0184	0.0003	2.3468	0.0119	0.8448
Group B						
XOM	TT2VFin	0.0333	0.0011	4.0724	0.0200	0.8330
	TT2VFin+M	0.0460	0.0021	5.8378	0.0303	0.6824
	TT2VFin+p	0.0859	0.0074	11.169	0.0619	- 0.108
	TT2VFin+p+M	0.0870	0.0076	11.315	0.0632	- 0.137
	Transformer+p	0.0408	0.0017	5.4162	0.0292	0.7495
	RNN	0.0351	0.0012	4.5526	0.0239	0.8150
CVX	LSTM	0.0342	0.0012	4.3279	0.0224	0.8248
	TT2VFin	0.0275	0.0008	3.3767	0.0149	0.8241
	TT2VFin+M	0.0380	0.0014	5.2774	0.0237	0.6661
	TT2VFin+p	0.0713	0.0051	10.285	0.0500	- 0.176
	TT2VFin+p+M	0.0700	0.0049	10.007	0.0486	- 0.134
	Transformer+p	0.0382	0.0015	4.4677	0.0279	0.7008
MS	RNN	0.0287	0.0008	3.6525	0.0172	0.8093
	LSTM	0.0274	0.0008	3.4605	0.0158	0.8261
Group C						
MS	TT2VFin	0.0166	0.0003	2.0747	0.0128	0.8325
	TT2VFin+M	0.0174	0.0003	2.1018	0.0130	0.8152
	TT2VFin+p	0.0495	0.0024	6.4577	0.0409	- 0.494
	TT2VFin+p+M	0.0484	0.0023	6.3163	0.0399	- 0.430
	Transformer+p	0.0177	0.0003	2.0849	0.0130	0.8097
	RNN	0.0258	0.0007	3.6253	0.0227	0.5933
GS	LSTM	0.0186	0.0003	2.4826	0.0149	0.7891
	TT2VFin	0.0221	0.0005	2.8797	0.0156	0.8545
	TT2VFin+M	0.0236	0.0006	3.2474	0.0176	0.8343
	TT2VFin+p	0.0661	0.0044	10.086	0.0521	- 0.304
	TT2VFin+p+M	0.0671	0.0045	10.265	0.0529	- 0.343
	Transformer+p	0.0214	0.0005	2.8383	0.0154	0.8621
GS	RNN	0.0246	0.0006	3.4026	0.0192	0.8200
	LSTM	0.0280	0.0008	4.3016	0.0229	0.7669

In this experiment, *TT2VFin* and its variations use 143K parameters, *Transformer+p* uses 173K parameters, RNN uses 559K parameters, and LSTM uses 355K parameters.

likely reason behind this behavior is that Time2Vec itself acts as an encoding scheme, and the combination of the two encodings confuses the model and leads to suboptimal performance. We also see that *TT2VFin* outperforms *Transformer+p* in 5 out of 6 cases and has comparable performance in one case, that proves that Time2Vec is more efficient for the time-series domain than the original positional encoding. Regarding the baseline deep learning models, the proposed model outperforms RNN despite utilizing about three times fewer parameters as RNN. LSTM demonstrates its power in financial prediction; nonetheless, there is still a performance difference between *TT2VFin* and LSTM. Statistically, *TT2VFin* outperforms LSTM by around $9.25\% \pm 0.63\%$ in Group A (averaging the metrics). We achieved the same results in other groups, where our proposed model is dominating over others. *TT2VFin+p* and *TT2VFin+M* have no change in performance,

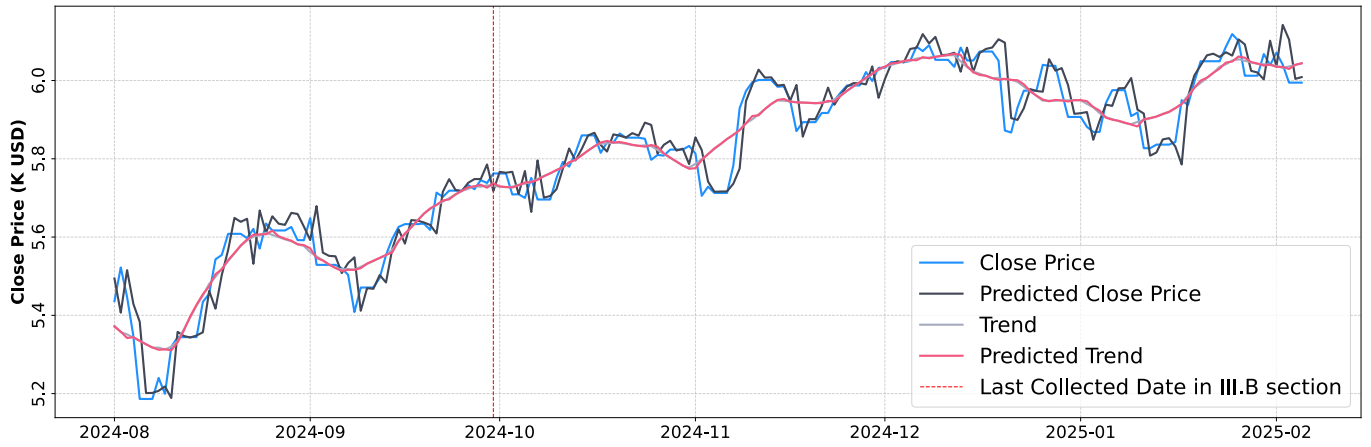


Fig. 5. The predicted close prices. S&P500 predicted close prices and original close prices are displayed in the same interval of 6-month (from 2024-08-01 to 2025-02-05) together with its trend and predicted trend. The red line demonstrates the last collected date we did in Section III-B. Note that a similar pattern of close prices and trends are attained by the NASDAQ stock.

but $TT2VFin+p+M$ cannot maintain its previous performance. In general, our proposed model uses 2 to 3 times fewer parameters than others but has roughly 14.23% greater performance (by averaging all the metrics differences between base and RNN and LSTM).

Additionally, we also investigated the performance of real close price predictions, visualized on Fig. 5. The model preserves its performance when facing with new data, which proves its generalization ability, specially in these days when the market is extremely unstable due to multiple factors.

V. CONCLUSION

In this paper, we investigated the application of deep learning for stock price prediction, which is a challenging time series forecasting problem in the financial industry. Notably, we represent a novel methodology for combining correlation features, aiming to better capture inter-stock relationships and temporal dynamics. We studied different markets by selecting correlation features to enhance the accuracy in predicting multiple stock prices and developed a novel neural network architecture that combines the Encoder of the Transformer with the Time2Vec model. Through careful testing and analysis, we have found that our model performs exceptionally well, better than standard models such as LSTM and RNN, and encoding methods like positional encoding from Transformer. In most cases, our multi-feature proposed model is better than a single-feature model, that verifies the importance and potential of feature selection and aggregation. In conclusion, our study demonstrated that techniques like Time2Vec and Encoder are promising future approaches combined with multi-source feature selection. We also believe that our approach can be successfully adopted and generalized to related financial and other time series prediction problems, and can achieve even better results for more refined data sources (e.g., having the intra-day historical data).

ACKNOWLEDGMENT

We thank our friend, Ho Minh Tri, for his resources assistance during the model training process.

REFERENCES

- [1] L. H. Pedersen, *Efficiently Inefficient: How Smart Money Invests and Market Prices Are Determined*. Princeton University Press, 2015.
- [2] A. J. Himanshu Gupta, "A study on stock forecasting using deep learning and statistical models," 2024. [Online]. Available: <https://arxiv.org/html/2402.06689v1>
- [3] D. Banerjee, "Forecasting of Indian stock market using time-series ARIMA model," in *2014 2nd International Conference on Business and Information Management (ICBIM)*, 2014, pp. 131–135.
- [4] J. Gardner, Everette S. and E. McKenzie, "Forecasting trends in time series," *Management Science*, vol. 31, no. 10, pp. 1237–1246, 10 1985.
- [5] A.-A. Semenoglou, E. Spiliotis, S. Makridakis, and V. Assimakopoulos, "Investigating the accuracy of cross-learning time series forecasting methods," *International Journal of Forecasting*, vol. 37, no. 3, pp. 1072–1084, 2021.
- [6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [7] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," 2019. [Online]. Available: <http://arxiv.org/abs/1907.00235>
- [8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017. [Online]. Available: <http://arxiv.org/abs/1706.03762>
- [9] W. C. Risheng Qiao and Y. Qiao, "Prediction of stock return by LSTM neural network," *Applied Artificial Intelligence*, vol. 36, no. 1, p. 2151159, 2022.
- [10] S. M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupart, and M. A. Brubaker, "Time2Vec: Learning a vector representation of time," 2019. [Online]. Available: <http://arxiv.org/abs/1907.05321>
- [11] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," 2022. [Online]. Available: <https://arxiv.org/abs/2012.08405>
- [12] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," 2023.
- [13] S. Shen, H. Jiang, and T. Zhang, "Stock market forecasting using machine learning algorithms," 2012. [Online]. Available: <https://api.semanticscholar.org/CorpusID:16643114>
- [14] "Yahoo finance," accessed: 2024-05-05. [Online]. Available: <https://www.finance.yahoo.com>
- [15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: <https://arxiv.org/abs/1512.03385>