International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V73I1P101

Volume 73 Issue 1, 1-18, January 2025

Original Article

Enhancing Stock Market Predictions with Multi-Feature Time2Vec-Transformer Models

Prakhar Srivastava

University of Illinois at Chicago, Illinois, USA.

Corresponding Author : prakharsrivsatava002@gmail.com

Received: 15 November 2024 Revised: 23 December 2024 Accepted: 10 January 2025 Published: 30 January 2025

Abstract *- Accurate financial forecasting is inherently challenging due to the complex and dynamic nature of stock price movements, driven by diverse and interrelated factors. Traditional models often fail to capture both short-term volatility and long-term dependencies adequately. This paper introduces a novel financial prediction model that combines the Transformer architecture with Time2Vec, extending existing methodologies by integrating multiple correlated market features. The approach enhances prediction accuracy and reduces errors in extreme market conditions. Extensive experiments conducted on stock indices, including NASDAQ, S&P 500, and Exxon Mobil, demonstrate that the proposed model significantly outperforms traditional methods such as LSTM and RNN. By leveraging correlated market features, the model effectively captures intricate relationships, leading to improved forecasting performance. The study highlights the advantages of incorporating correlation-aware features in financial models and discusses potential applications in broader financial markets. These findings pave the way for developing more robust prediction systems, offering valuable insights for investors, analysts, and policymakers in managing financial risks and opportunities.*

Keywords *- Financial forecasting, Transformer, Time2Vec, Stock price prediction, NASDAQ, S&P 500, Exxon mobil, LSTM, RNN, Correlated market features.*

1. Introduction

Time series determining is testing, particularly in the monetary business [1]. It includes measurably understanding complex direct and non-straight cooperations inside verifiable information to foresee what's in store. Customary measurable methodologies generally embrace straight relapse, outstanding smoothing [2], and auto relapse model [3]. With the advances in profound learning, late works vigorously put resources into outfit models and succession to arrangement displaying like RNN (repetitive brain organizations) and Long Momentary Memory [4]. Notwithstanding, the essential disadvantage of these strategies is that the RNN family battles to catch very long haul conditions [5].

A notable grouping-to-grouping model called Transformer [6] has made extraordinary progress in NLP, particularly LLM like ChatGPT Gemini. Not the same as RNN-based modes, Transformer utilizes a multi-head self-consideration component to familiarize people worldwide with the relationship among various positions. For instance, as far as the tech business, for example, the cutback of representatives in 2023 initially started with Meta, then moved to research, Microsoft, and Apple. Then, the stock cost of those organizations is practically down or up simultaneously. On the other hand, as far as informal community organizations are concerned, In different ventures like oil, individuals suspect the ongoing business sector and sell all their stocks.

1.

2. Background And Related Work

2.1. Neural Network

A Neural Network (also Artificial Neural Network or Neural Net, abbreviated ANN or NN) is a model inspired by the structure and function of biological neural networks in animal and human brains. Neural networks consist of neurons organized. into layers (see 1), where the layers usually perform a linear transformation followed by a non-linear activation function. This allows networks to learn complex patterns and relationships in the data.

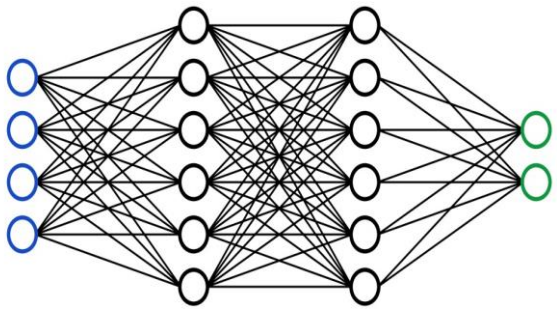

Fig. 1 An example Neural Network (from [7])

There are numerous different activation functions. We can put them into 2 categories: Non-Linear Activate Functions and others.

2.2. Non-Linear Activation Functions

In this section, we will walk through the most important activation functions and their properties.

Sigmoid:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

It's especially useful for classification or probability prediction tasks because it can be implemented in the training of computer vision and deep learning networks. However, vanishing gradients can make these problematic when used in hidden layers, and this can cause issues when training a model. That is why nowadays, ReLU (see 3) is more widely used in computer vision and deep learning than Sigmoid.

Tanh:

$$f(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})} \tag{2}$$

It is a steeper gradient and also encounters the same vanishing gradient challenge as sigmoid.

ReLU:

$$f(x) = \max(0, x) \tag{3}$$

ReLU does not activate every neuron in sequence at the same time, making it more efficient than the tanh or sigmoid/logistic activation functions. Unfortunately, the downside of this is that some weights and biases for neurons in the network might not get updated or activated.

LeakyReLU:

$$f(x) = \max(0.1 * x, x) \tag{4}$$

The advantages of Leaky ReLU are the same as that of ReLU, in addition to the fact that it does enable backpropagation, even for negative input values.

ParametricReLU:

$$f(x) = \max(a * x, x) \tag{5}$$

Parametric ReLU is one more variation of ReLU that is expected to tackle the issue of inclination becoming zero for the left 50% of the pivot.

Softmax:

$$\text{softmax}\,(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \tag{6}$$

It is most usually utilized as an enactment capability for the last layer of the brain network on account of multi-class characterization.

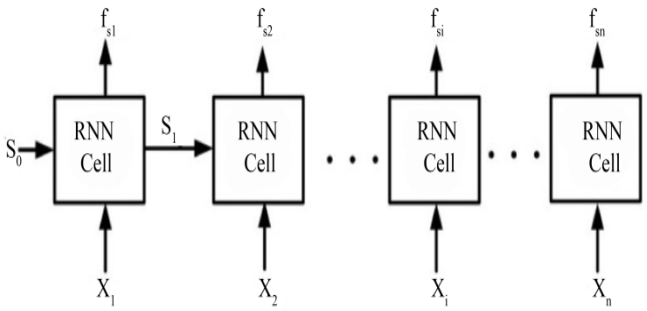

Fig. 2 Recurrent Neural Network model (from [11])

2.3. ARIMA

Ordinary estimating strategies in writing use measurable devices, such as remarkable smoothing (ETS) [2] and auto backwards coordinated moving normal (ARIMA) [3], on mathematical time series information for making one-stride ahead expectations. These forecasts are then recursively taken into account in terms of what in-store contributions are needed to acquire multi-step gauges. Multi-skyline gauging techniques [8] and [9] straightforwardly produce synchronous expectations for different predefined future time steps.

2.4. RNN - Recurrent Neural Network

Intermittent brain organizations (RNNs) are intended to deal with worldly issues, and they are broadly utilized for successive information and time-series investigation. The RNN family is a normal decision for monetary gauging, all things considered.

See, for example, [10]. Despite the fact that RNN can precisely describe the logical connection between consecutive information, this relationship debilitates as the whole distance between them develops. 2 presents the unfurled engineering of an RNN. Also, an RNN model has the evaporating slope issue for the long grouping information. Nonetheless, LSTM can mitigate this issue during preparation.

Moreover, an RNN model has the vanishing gradient problem for the long sequence data. However, LSTM can prevent this problem during training.

2.5. CNN - Convolutional neural network

A convolutional brain organization (CNN) includes unique layers that perform convolution and pooling. CNN designs act as the most broadly involved ANNs in picture handling and PC vision, yet they are once in a blue moon utilized for monetary.

2.6. LSTM - Long Short-Term Memory

Long Momentary Memory (LSTM) is an exceptional RNN that has a memory component with entryways, making the organization able to learn long haul conditions and forestalling the evaporating inclination issue. The design of an LSTM cell is exhibited in 3.

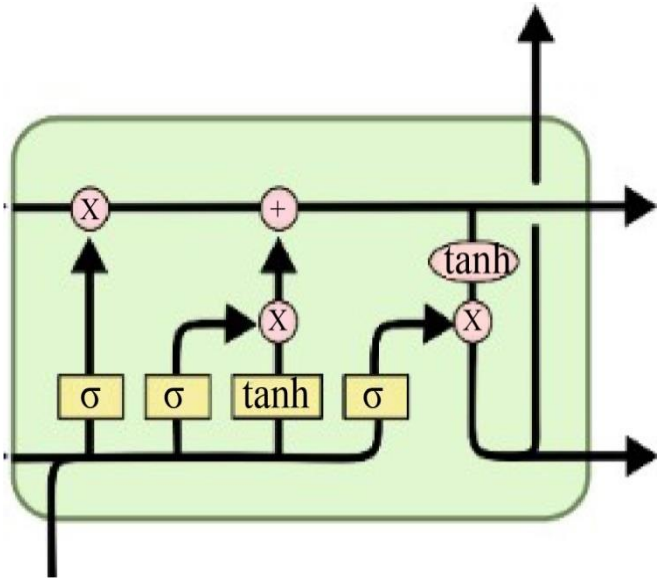

Fig. 3 LSTM - Long Short-Term Memory architecture (from [12])

LSTM [4] is generally utilized for monetary time series expectations, especially in estimating stock costs. There are likewise quite a large number of factors of LSTM like BiLSTM, Conv-LSTM [13], and Mindful-LSTM [14] with a consideration system to foresee stock cost development. Nonetheless, [5] brings up that LSTM can separate 50 positions close by with a viable setting size of around 200. That implies that LSTM-based models experience the ill effects of the trouble in catching very long haul conditions in time series

2.7. Time2Vec

Time2Vec [15] is an approach that provides a model–agnostic vector representation of time. Time decomposition technique that encodes a temporal signal into a set of frequencies.

$$t2v(\tau)[i] = \begin{cases} w_i\tau + \varphi_i & \text{if } i = 0 \\ \sin(w_i\tau + \varphi_i), & \text{if } 1 \le i \le k \end{cases} \quad (7)$$

This plan has three significant properties:

- Catching both occasional and non-intermittent examples.
- Being invariant to time re-scaling.
- Being adequately basic so, it very well may be joined with many models.

The transgression actuation capability is propelled pieces of positional encoding from [6], which can be joined with the transformers model.

2.8. Transformer

The advancement of Transformer in profound learning [6] has carried extraordinary interest as of late because of its magnificent exhibitions in normal language handling (NLP) [16]. Throughout the course of recent years, various Transformers have been proposed to propel the cutting edge exhibitions of different errands ”altogether” Transformers have shown incredible demonstrating skill for long-range conditions and cooperations in consecutive information and along these lines are interesting to time series displaying.

Numerous variations of Transformer have been proposed to address extraordinary difficulties in time series demonstrating and have been effectively applied to different time series errands, like gauging oddity recognition and characterization [17]. 4 shows the engineering of the Transformer.

2.9. Transformer and Time-Embedding

Unlike natural language data, where positional encoding suffices to capture word order, processing sequential data with Transformers necessitates a nuanced approach to extract temporal dependencies. This gap is bridged by Time Embedding, which imbues the model with an understanding of chronological order, preventing nonsensical predictions where distant historical prices hold equal sway as recent ones.

To address the temporal challenges inherent in Transformers, the Time2Vec methodology is adopted, offering a model-agnostic vector representation of time. This approach encapsulates both periodic and non-periodic patterns while maintaining invariance to time re-scaling, ensuring the model’s ability to comprehend and utilize temporal features effectively in predicting stock prices.

2.10. Optimizer

In supervised learning, the training of ANNs is usually carried out as a data-driven numerical optimization of a lost function. In feedforward NNs, where the neurons are organized in layers, backpropagation and a stochastic gradient descent optimizer are mainly used. There is an enormous optimizer out there, but in this thesis, we are using Adam Optimizer.

2.11. Adaptive Moment Estimation

Versatile Second Assessment is a calculation for improvement procedure for slope drop. The strategy is truly proficient while working with huge issues, including a ton of information or boundaries. It requires less memory and is effective. Instinctively, it is a blend of the ’slope drop with energy’ calculation and the ’RMSP’ calculation. Adam Enhancer acquires the qualities or the positive characteristics of the two techniques and expands upon them to give a more enhanced inclination drop.

2.12. Momentum

This calculation is utilized to speed up the angle plunge calculation by taking into thought the ’dramatically weighted normal’ of the angles. Utilizing midpoints causes the calculation to unite towards the minima at a quicker pace.

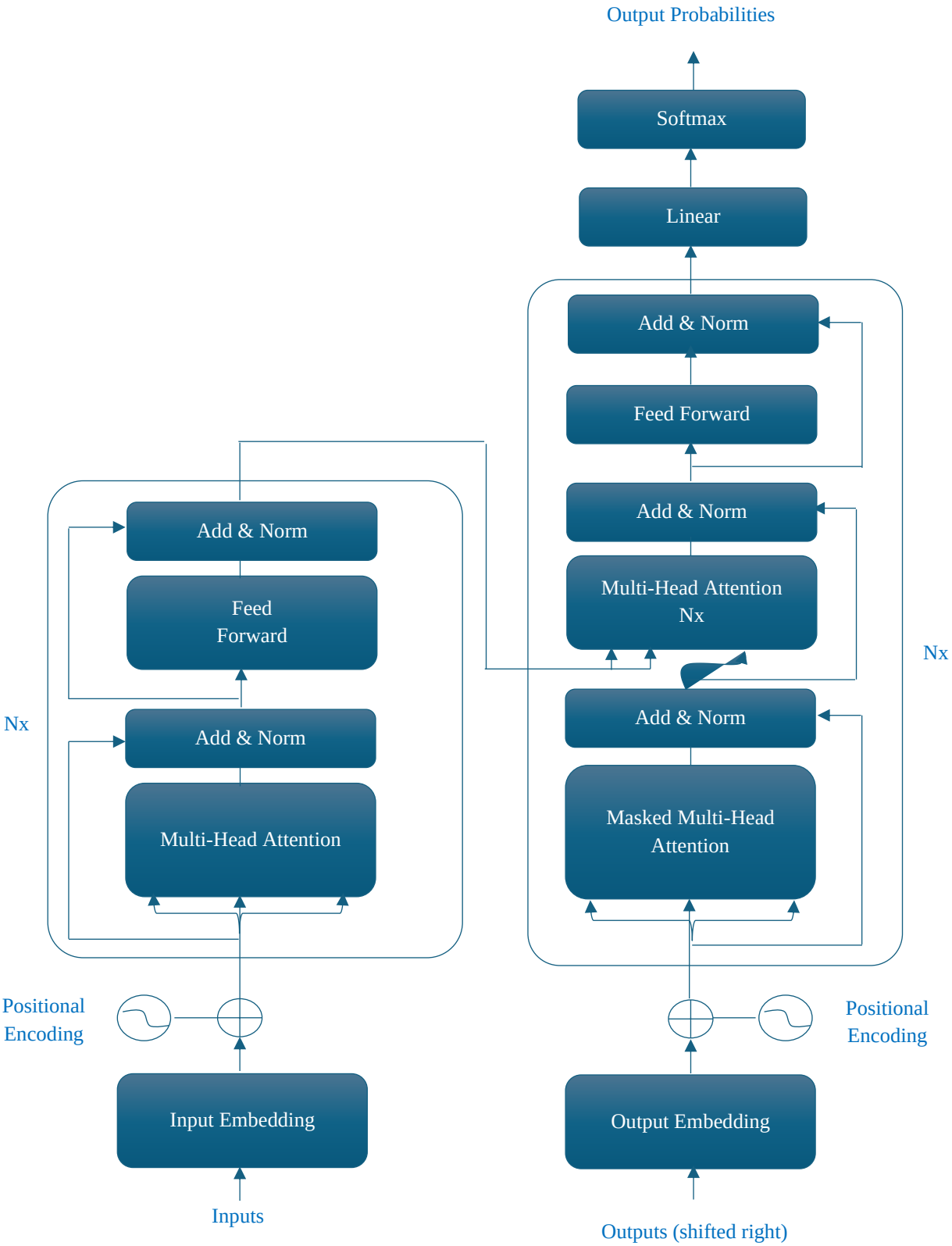

Fig. 4 The transformer-model architecture (from [18])

2.13. Root Mean Square Propagation - RMSP

Root mean square prop or RMSprop [19] is a versatile learning calculation that attempts to develop AdaGrad [20] further. Rather than taking the total amount of squared slopes like in AdaGrad, it takes the 'dramatic moving normal'.

2.14. Evaluation metrics

In this part, we outline the main measurements that are utilized to assess the presentation of the expectations of a determining technique.

2.15. MAPE - Mean Absolute Percentage Error

Otherwise called mean outright rate deviation (MAPD), is a proportion of the expectation exactness of an estimating strategy in measurements. It ordinarily communicates the exactness as a proportion characterized by the equation.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Y_i - \hat{Y}_i}{Y_i}\right| \tag{8}$$

Y_i: Actual value.

$\hat{Y}_i$: Predicted value. n: number of values.

2.16. MAE - Mean Absolute Error

Mean outright mistake (MAE) is a proportion of blunders between matched perceptions communicating a similar peculiarity.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} | Y_i - \hat{Y}_i | \tag{9}$$

Y_i: Actual value.

$\hat{Y}_i$: Predicted value. n: number of values.

2.17. RMSE - Root Mean Square Error

Root mean square blunder (RMSE) is the residuals' standard deviation or the normal contrast between the anticipated and real qualities delivered by a factual model.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{n}} \tag{10}$$

Y_i: Actual value.

$\hat{Y}_i$: Predicted value. n: number of values.

2.18. MSE - Mean Square Error

Mean Squared Mistake (MSE) is the typical squared contrast between the worth seen in a factual report and the qualities anticipated from a model.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{11}$$

Y_i: Actual value.

$\hat{Y}_i$: Predicted value. n: number of values.

2.19. R2 - R-Square

R-squared esteem shows how well the model predicts the result of the ward variable. R-squared values range from 0 to 1. A squared worth of 0 implies that the model makes sense of or predicts 0% of the connection between the ward and autonomous factors.

$$R^2 = 1 - \frac{\sum(Y_i - \hat{Y}_i)^2}{\sum(Y_i - \bar{Y})^2} \tag{12}$$

Y_i: Actual value.

$\hat{Y}_i$: Predicted value. $\bar{Y}$: Mean of all values. n: number of values.

2. 3. Our Work

3.1. Motivation

The forecasting finance area is currently under intense research focus: many researchers and specialists are trying to combine Time2Vec with CNN, RNN, LSTM, and Attention mechanisms. Moreover, not only finance but also other fields, such as Aeroengine Risk Assessment [21] and Predicting Production in Shale and Sandstone Gas Reservoirs [22], were studied.

They usually use only one feature to predict things (for instance, only one stock price to predict its prices). However, using techniques that rely solely on a single feature can be quite challenging when it comes to predicting unexpected financial events, which are often influenced by external factors beyond our control. On the flip side, in the world of finance, it is common practice to focus on identifying and analyzing correlations between specific features and target prices when making investment decisions.

Building upon previous challenges, we were motivated to explore the integration of correlations from multiple features into the Transformer model. We represent a new approach to existing methods by developing a neural network architecture that consists of Time2Vec, residual, multiple attention layers, pooling, and a densely connected part.

Given the limitations inherent in financial data, often sparse and messy, we sought to address these issues by selecting multiple stocks exhibiting similar behavior and interconnectedness. The subsequent sections of this paper will detail our approach to time series modeling.

3.2. Methodology

The performance of a stock market predictor heavily depends on the correlations between historical data for training and the current input for prediction. The results are shown below; here, in 5 and 6, we use Exxon Mobil and NASDAQ stock prices as the base market. Graph (5) reveals that base markets' auto-correlation is solely non-zero at the origin.

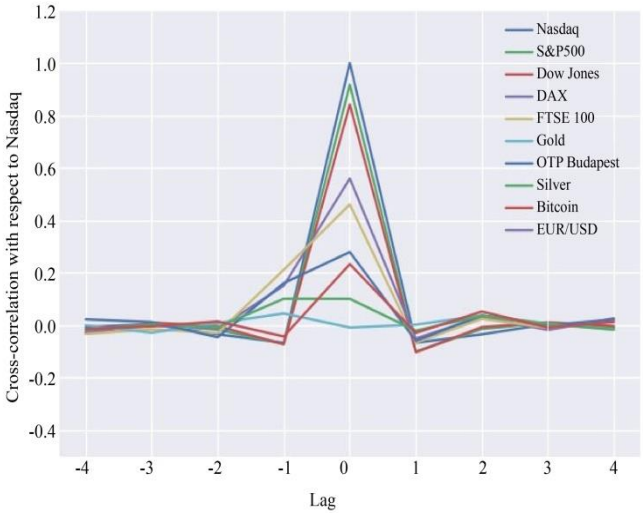

Fig. 5 Auto-correlation and cross-correlation of markets trend using Exxon Mobil as a based market

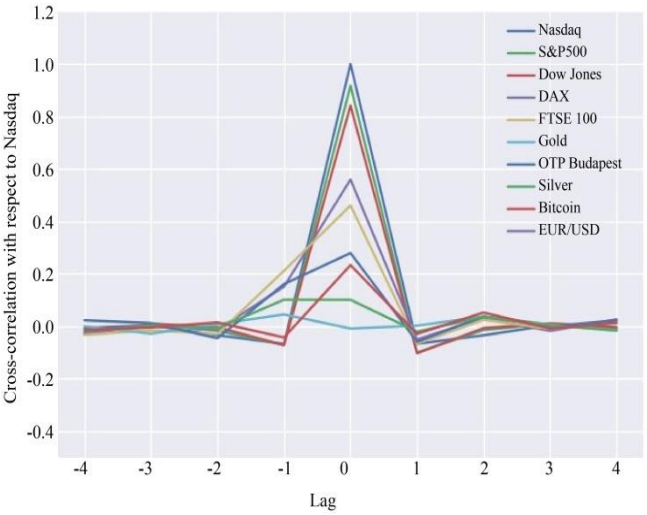

Fig. 6 Auto-correlation and cross-correlation of market trends using NASDAQ as a based market

This observation suggests that the daily trend of Exxon stock approximates a Markov process [23]. Consequently, historical data offers limited insight into its future movements. However, data sources such as Chevron are promising features for our approach. From 6, we get the same conclusion that NASDAQ and S&P500 will be good data to pair with. Moreover, we will try to evaluate whether it will be better if we put more relatable data (for example, NASDAQ, S&P500, Dow Jones, and DAX into the model) than only 2 features (NASDAQ and S&P500).

3.3. Mean Not NaN

Each time series might have missing data points represented as NaN values, and we have to use a proper combination approach to handle this issue. In this paper, we considered Geometry Mean Not NaN - GMNN (7) and Arithmetic Mean Not NaN - AMNN (8).

3.4. Geometry Mean Not NaN - GMNN

GMNN is the technique we found to combine multiple normalized data into one normalized data. This technique assures that the transformed data will be between 0 and 1 – which is the attribute the combined data must have. How do GMNN work? – GMNN iterates by row and takes the Geometry Mean of real numbers in each row and assigns it as an output value for that row.

3.5. Arithmetic Mean Not NaN - AMNN

AMNN follows the same idea as GMNN, that is, combining multiple normalized data into one normalized data. AMNN iterates by row, takes the Arithmetic Mean of real numbers in each row and assigns it as an output value for that row.

3.6. AMNN or GMNN?

Both AMNN and GMNN can resolve the task. However, by experiment several times, we can conclude that GMNN yields better results in predicting stock prices than AMNN, so we will choose GMNN as a method to represent our result in this thesis.

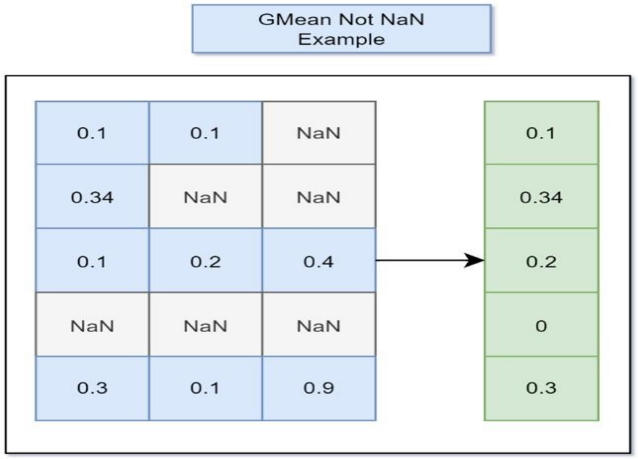

Fig. 7 Simple illustration with 1-dimensional input pass through GMNN step

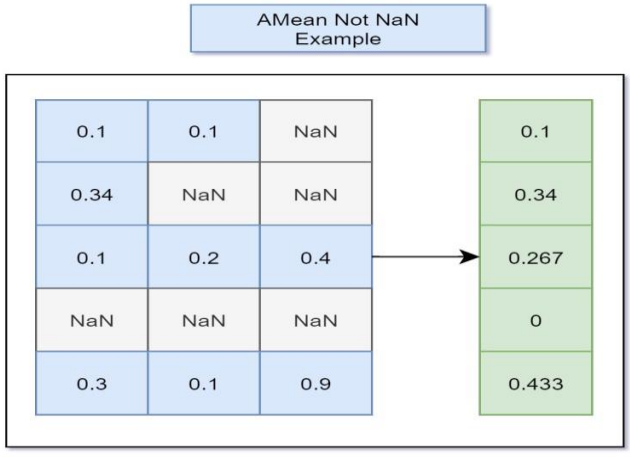

Fig. 8 Simple illustration with 1-dimensional input passes through the AMNN step

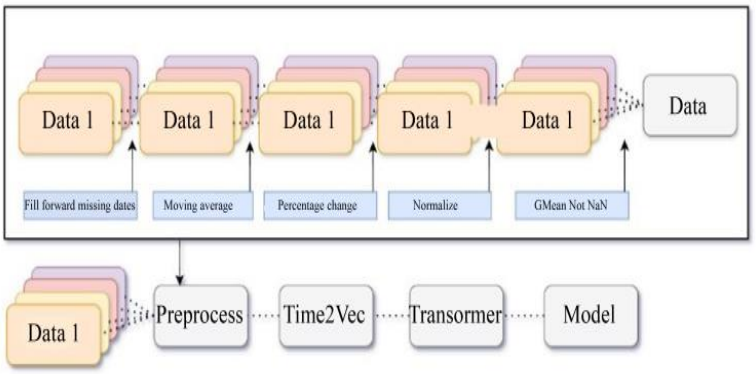

Fig. 9 Pre-processing data pipeline

4. Experiment

4.1. Data Collection

To evaluate the proposed method. This paper uses two groups of stocks:

1) Group 1: NASDAQ, S&P500, DJI, DAX. 2)
2) Group 2: Exxon Mobil, Chervon.

In each group, group members are highly correlated with each other (see 6 for group 1, 5 for group 2).

We collect the daily quote data of those stocks from Yahoo Finance [24].

I will show an example of a data download from Yahoo.

Date: The particular day when the exchange happened.
Open: The cost of the stock record toward the start of the exchange day (USD).
High: The most exorbitant cost arrived at by the stock record during the exchanging day (USD).
Low: The most minimal cost arrived at by the stock file during the exchanging day (USD).
Close: The cost of the stock record toward the finish of the exchanging day (USD).
Volume: The complete number of offers exchanged during the day (share).

4.2. Data Pre-processing

First, this paper fills forward all missing dates in the range from the starting date to the current date. To smooth out the ups and downs in stock prices, we start by averaging the values over 14 days (see 2). Then, this paper calculates the percentage change for the next day (see 3). When building our model, this paper makes sure to scale these percentage changes so they fall between 0 and 1. Using a standard min-max scaling method, this paper gets the result of 4 for NASDAQ and V for S&P500.

$$x_{\text{standardized}} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{13}$$

Finally, this paper pushes all the processed data (4 and 5) to the Geometry Mean Not NaN block (7), which will be the final input to feed for the model (see 6). We use the technique of moving average, which uses the sequence of the previous day's movements to predict the next day's change, which is our target. This approach takes into account the short-term trends in stock prices and uses time series indicators as features to classify the trend.

4.3. Our Model

We will introduce a novel network architecture, presented on 10, that combines the Time2Vec and the Transformer model. Each colored rectangle in the 10 represents a layer. Each layer have a tuple of numbers represent the output size after the input passes through it.

4.3.1. Input Layer: Take the processed data.

4.3.2. Time2Vec Layer: This layer contains 4 sub-layers

a) Linear

This layer is used for capturing linear trends, including steady increases and decreases in prices over time - which might not be periodic but still play an important role in time-series.

b) Sine and Cosine

These two layers are really important for our model. Their first task is encoding positions, which we can consider as time, which represents continuous time instead of discrete positions. They also capture the periodic behaviors of prices. In the Time2Vec paper, they use only one periodic function, which is sine 7. But after experiment, we realize that using an additional periodic function will yield a better result in this case.

c) Concat

Use to concatenate the above three layers and prepare for the next steps.

4.3.3. Concat

This layer takes the Input layer and combines it with the output from the Time2Vec layer. We make residual connections known as ResNet [25]. This action significantly improves the behavior of the network, which makes the

Time2Vec layer more valuable and powerful for our architecture and our model.

4.3.4. Attention Layers

As known as the Multi-head Attention layer, this group contains 5 single consecutive attention layers. This structure allows our model to jointly attend to values from different aspects at different positions to ensure that the model can study the trend as clearly as possible.

4.3.5. Pooling 1D

This layer is responsible for reducing the spatial dimensions of the output from Multi-head Attention layers in terms of width and height while retaining the most important information.

4.3.6. Dropout

We need to prevent over-fitting the model. The chosen dropout rate is 0.1.

4.3.7. Dense

Dense is a fully connected neural network layer. The first Dense applies the activate functions ReLU (see 3), and the other applies the linear activation function. We use it to decrease the data dimension.

4.4. Why it must be this Model

We chose Time2Vec to catch continuous attributes of time in the data, Attention to get a deep understanding of the movement of the trend, and Dropout to prevent over-fitting.

We believe that each layer plays an important role in the architecture.

4.5. Result and Evaluation

For evaluation, we will split the input into 3 parts before actually put them into the training process. Train data will be 80% of the input. Validation data will be the next 10% of the input. Test data will be 10% left of the input.

Table 1. NASDAQ data download from Yahoo Finance

Date	Open	High	Low	Close	Volume
1971-02-05	100.0000	100.0000	100.0000	100.0000	0
1971-02-08	100.8399	100.8399	100.8399	100.8399	0
1971-02-09	100.7600	100.7600	100.7600	100.7600	0
1971-02-10	100.6900	100.6900	100.6900	100.6900	0
1971-02-11	101.4499	101.4499	101.4499	101.4499	0
...	...	...	...	...	...
2024-05-01	15646.0898	15926.2197	15557.6396	15605.4804	5277790000
2024-05-02	15758.1103	15862.7900	15604.7304	15840.9599	4901610000
2024-05-03	16147.4804	16204.7099	16068.3398	16156.3300	4887310000
2024-05-06	16208.5400	16350.0800	16197.8603	16349.2500	4460130000
2024-05-07	16208.5000	16396.4589	16326.2109	16373.5625	2693650000

Table 2. NASDAQ data after filling missing dates and applying moving average

Date	Open	High	Low	Close	Volume
1971-02-18	101.3850	101.3850	101.3850	101.3850	6.2860e+7
1971-02-19	101.4350	101.4350	101.4350	101.4350	6.2860e+7
1971-02-20	101.3521	101.3521	101.3521	101.3521	6.2860e+7
1971-02-21	101.2692	101.2692	101.2692	101.2692	6.2860e+7
1971-02-22	101.1864	101.1864	101.1864	101.1864	6.2860e+7
...	...	...	...	...	...
2024-05-03	15729.2649	15846.3864	15614.3349	15750.9149	4.870674e+9
2024-05-04	15787.2942	15904.3207	15680.9206	15815.0535	4.859489e+9
2024-05-05	15845.3235	15962.2550	15747.5064	15879.1921	4.848303e+9
2024-05-06	15903.3528	16020.1893	15814.0921	15943.3307	4.837117e+9
2024-05-07	15952.1349	16067.8352	15870.7528	15988.7533	4.815941e+9

Table 3. NASDAQ data after computing percentage change

Date	Open	High	Low	Close	Volume
1971-02-19	0.000493	0.000493	0.000493	0.000493	0
1971-02-20	-0.000817	-0.000817	-0.000817	-0.000817	0
1971-02-21	-0.000818	-0.000818	-0.000818	-0.000818	0
1971-02-22	-0.000818	-0.000818	-0.000818	-0.000818	0
1971-02-23	-0.000734	-0.000734	-0.000734	-0.000734	0

...	...	...	...	...	...
2024-05-03	0.002734	0.002839	0.003883	0.003981	-0.006248
2024-05-04	0.003689	0.003656	0.004264	0.004072	-0.002297
2024-05-05	0.003676	0.003643	0.004246	0.004056	-0.002302
2024-05-06	0.003662	0.003629	0.004228	0.004039	-0.002307
2024-05-07	0.003067	0.002974	0.003583	0.002849	-0.004378

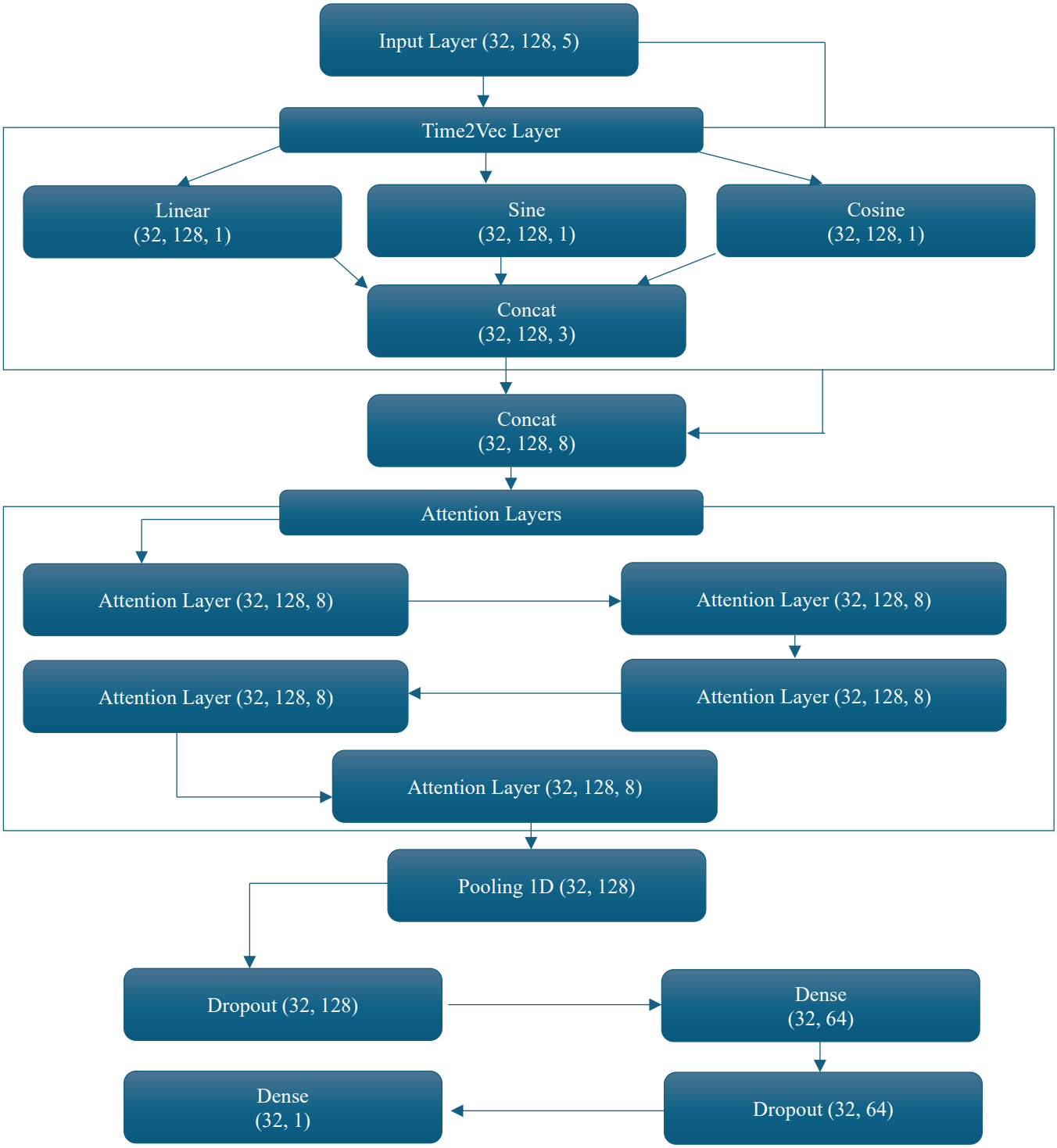

Fig. 10 Propose model

Table 4. NASDAQ data after normalizing

Date	Open	High	Low	Close	Volume
1971-02-19	0.635564	0.635564	0.635564	0.635564	0.481983
1971-02-20	0.606946	0.606946	0.606946	0.606946	0.481983
1971-02-21	0.606932	0.606932	0.606932	0.606932	0.481983
1971-02-22	0.606917	0.606917	0.606917	0.606917	0.481983
1971-02-23	0.608753	0.608753	0.608753	0.608753	0.481983
...	...	...	...	...	...
2024-05-03	0.684514	0.686807	0.709619	0.711752	0.449510
2024-05-04	0.705385	0.704658	0.717949	0.713747	0.470046
2024-05-05	0.705089	0.704367	0.717554	0.713387	0.470019
2024-05-06	0.704795	0.704079	0.717161	0.713029	0.469991
2024-05-07	0.691800	0.689762	0.703062	0.687029	0.459228

Table 5. S&P500 data after normalizing

Date	Open	High	Low	Close	Volume
1928-01-13	0.521913	0.521913	0.521913	0.521913	0.458965
1928-01-14	0.490008	0.490008	0.490008	0.490008	0.458965
1928-01-15	0.489934	0.489934	0.489934	0.489934	0.458965
1928-01-16	0.489860	0.489860	0.489860	0.489860	0.458965
1928-01-17	0.490609	0.490609	0.490609	0.490609	0.458965
...	...	...	...	...	...
2024-05-03	0.561882	0.562470	0.570750	0.574165	0.460139
2024-05-04	0.572563	0.568649	0.577909	0.576782	0.455490
2024-05-05	0.572467	0.568569	0.577788	0.576667	0.455481
2024-05-06	0.572371	0.568490	0.577668	0.576552	0.455472
2024-05-07	0.573196	0.563337	0.571300	0.561493	0.421249

Table 6. Combine NASDAQ and S&P500 using GMNN block

Date	Open	High	Low	Close	Volume
1928-01-13	0.521913	0.521913	0.521913	0.521913	0.458965
1928-01-14	0.490008	0.490008	0.490008	0.490008	0.458965
1928-01-15	0.489934	0.489934	0.489934	0.489934	0.458965
1928-01-16	0.489860	0.489860	0.489860	0.489860	0.458965
1928-01-17	0.490609	0.490609	0.490609	0.490609	0.458965
...	...	...	...	...	...
2024-05-03	0.620174	0.621537	0.636408	0.639268	0.454793
2024-05-04	0.635514	0.633011	0.644135	0.641621	0.462711
2024-05-05	0.635327	0.632836	0.643890	0.641394	0.462693
2024-05-06	0.635141	0.632662	0.643647	0.641169	0.462675
2024-05-07	0.629712	0.623353	0.633766	0.621098	0.439829

Table 7. Metrics for models in group 1

(a) M1 1

Train MAE	0.0127
Train MAPE	2.3391
Train loss	0.0004
Val MAE	0.0131
Val MAPE	2.1929
Val loss	0.0004
Test MAE	0.0147
Test MAPE	3.3804
Test loss	0.0006

(b) M1 2

Train MAE	0.0137
Train MAPE	2.2923
Train loss	0.0004
Val MAE	0.0124
Val MAPE	2.0052
Val loss	0.0003
Test MAE	0.0192
Test MAPE	3.2285
Test loss	0.0008

(c) M1 3

Train MAE	0.0106
Train MAPE	2.1629
Train loss	0.0003
Val MAE	0.0119
Val MAPE	2.4171
Val loss	0.0004
Test MAE	0.0108
Test MAPE	2.1768
Test loss	0.0003

(d) M1 4

Train MAE	0.0105
Train MAPE	2.0188
Train loss	0.0003
Val MAE	0.0128
Val MAPE	2.3027
Val loss	0.0004
Test MAE	0.0119
Test MAPE	2.15
Test loss	0.0003

Table 8. Metrics for models in Group 2

(a) M2 1

Train MAE	0.0138
Train MAPE	2.1378
Train loss	0.0003
Val MAE	0.012
Val MAPE	1.8924
Val loss	0.0003
Test MAE	0.0303
Test MAPE	6.14
Test loss	0.0028

(b) M2 2

Train MAE	0.0114
Train MAPE	1.9157
Train loss	0.0002
Val MAE	0.0099
Val MAPE	1.6689
Val loss	0.0002
Test MAE	0.0163
Test MAPE	3.7077
Test loss	0.0009

(c) M2 3

Train MAE	0.011
Train MAPE	1.7753
Train loss	0.0002
Val MAE	0.0096
Val MAPE	1.5592
Val loss	0.0002
Test MAE	0.018
Test MAPE	3.8575
Test loss	0.0011

4.6. Result of Group 1: NASDAQ, S&P500, DJI, DAX

We have trained 5 models for this group.

Note: Multi-feature models are M1 1 and M1 _4. Note: One feature model is M1 2 and M1 3.

1) M1 1: nas sp dji dax (7a)
Features: NASDAQ, S&P500, DJI, and DAX
2) M1 2: nas (7b)
Features: NASDAQ
3) M1 3: sp (7c)
Features: S&P500
4) M1 4: nas sp (7d)
Features: NASDAQ, S&P500

4.7. Evaluation for Group 1

We can see that M1 4 (7d) yields better results than M1 1 (7a), which means that 4 features do not lead to a better model. We can also point out that M1 4 (7d) have better results than M1 2 (7b) and M1 _3 (7c). This means that the multi-feature models may work better than the single-feature models.

4.8. Result of Group 2: Exxon Mobil Chervon

We have trained 3 models for this group.

Note: Multi-feature model is M2 3.

Note: One feature model is M2 1 and M2 2.

1) M21: Exxon (8a)
Features: Exxon
2) M22: Chervon (8b)
Features: Chervon
3) M2 3: exxon _chervon (8c)
Features: Exxon, Chervon

4.9. Evaluation for Group 2

The results in 4-G show similar findings here. Multi-feature models outperform single-feature models, as evidenced by M2 3 (8c) being superior to M2 1 (8a). However, model M2 2 slightly outperforms the multi-feature models, as indicated in 8b. This shows that although the multi-feature model outperforms one-feature models under certain circumstances, the special onefeature model may still have better performance.

4.10. Further Evaluation

From now on, we will not split the data into 3 parts (train, validation, and test) to evaluate since we have the models (as trained above). Moreover, we will use RMSE (from 2-Q), MSE (from 2-R), and R2 (from 2-S) as additional metrics with MAPE (from 2-O) and MAE (from 2-P) to have a clear evaluation of models. We also use Accuracy, Precision (Pre), Recall, and F1-score (F1) to evaluate our trend prediction (classification problem). Note: Accuracy, Pre, Recall, and F1-score will be evaluated on 2 labels: Decreasing and Non-decreasing. We will see there is only one value for accuracy but two for the others (the left value will be for label 0, and the right value will be for label 1).

4.11. Method for Decoding Predicted Output to Prices

We need to follow the pipeline of how we pre-processed the data to decode it. Figure 11 describes the decoding pipeline. Walk- Through for each step in the pipeline.

Note that x is used for predicted values, and v is used for real values.

(1) De-normalized step: This step includes 2 small steps in it.
(a)Making sure that the output is between 0 and 1, if there exist values such that $x_i > 1$, we will take its reciprocal instead. Cases that $x_i < 0$ seem to not occur in the experiment.
(b)Apply min-max de-normalized formula:

$$\text{xpct} = \text{xnor} \times (\max - \min) + \min \tag{14}$$

$x_{predictor}$: vector of output from a model (predicted normalized values). *max*, *min*: max and min value from 13. x_{pct}: vector of predicted percentage change values. Note that *max* and *min* are max and min of open, high, low, and close.

(2) De-percentage change step:

$$x_{mva} = \begin{cases} v_{pct}, i = 0 \\ v_{pct} \times (1 + x_{pct}), otherwise \end{cases} \tag{15}$$

v_{pct}: vector of real percentage change values.
x_{pct}: vector of predicted percentage change values.
v_{mva}: vector of predicted moving average prices.

3) De-moving average step:

$$x_{close} = \begin{cases} v_{close} & , i \leq s-1 \\ x_{mva} \times s & -\sum_{k=i-s}^{i-1} v_{close}, i \geq s \end{cases} \tag{16}$$

x_{mva}: vector of predicted moving average prices.
v_{close}: vector of real close prices.
x_{close}: vector of predicted close prices.
s: moving average step

In this thesis, we follow a moving average strategy every fortnight, which means our step will be 14. So, 16 can also be written as

$$_{lose} = \begin{cases} v_{close} & , i \leq 13 \\ x_{mva} \times 14 & -\text{P}_{ik-=1i-14}\, v_{close}, i \geq 14 \end{cases} \tag{17}$$

4.12. Deep evaluation with respect to NASDAQ

In terms of NASDAQ, we can use M1 1, M1 2, and M1 4. This model contains NASDAQ as a feature in them. Base on 9,10,11 and 12. We can proudly say that, with respect to NASDAQ, model M1 2 (one-feature) and model M1 4 (two-feature) outperform model M1 1 (four-feature).

Table 9. Predicting normalized values metrics of the NASDAQ dataset

(a) M1 1

RMSE	0.025553224
MSE	0.000652967
MAPE	2.83452573
MAE	0.016752568
R2	0.836478961

(b) M1 2

RMSE	0.01870842
MSE	0.000350005
MAPE	2.006188642
MAE	0.011983401
R2	0.91235242

(c) M1 4

RMSE	0.02125059
MSE	0.000451588
MAPE	2.408059498
MAE	0.014652739
R2	0.886914298

Table 10. Predicting moving average values of the NASDAQ dataset

(a) M1 1

RMSE	5.73749947
MSE	32.91890017
MAPE	0.076206824
MAE	2.399314816
R2	0.999997427

(b) M1 2

RMSE	4.56276691
MSE	20.81884188
MAPE	0.054502324
MAE	1.727510579
R2	0.999998373

(c) M1 4

RMSE	4.774897561
MSE	22.79964672
MAPE	0.066603007
MAE	2.009664443
R2	0.999998218

Table 11. Predicting close price metrics of the NASDAQ dataset.

(a) M1

RMSE	80.29814013
MSE	6447.791308
MAPE	1.069284617
MAE	33.56795279
R2	0.999498387

(b) M1 2

RMSE	63.85738223
MSE	4077.765265
MAPE	0.764418767
MAE	24.16898073
R2	0.999682766

(c) M1 4

RMSE	66.82621854
MSE	4465.743484
MAPE	0.929676058

MAE	28.11649422
R2	0.999652583

Table 12. Predicting trend metrics of the NASDAQ dataset

(a) M1 1

Acc	0.8921	
Pre	0.85623003	0.91801682
Recall	0.88347914	0.89922817
F1	0.86964119	0.90852537

(b) M1 2

Acc	0.9026	
Pre	0.87157555	0.92468766
Recall	0.89146697	0.91027196
F1	0.88140905	0.91742318

(c) M1 4

Acc	0.912	
Pre	0.89040576	0.92683562
Recall	0.89311525	0.92490906
F1	0.89175845	0.92587134

Table 13. Predicting normalized values metrics of the S&P500 dataset

(a) M1 1

RMSE	0.018034203
MSE	0.000325232
MAPE	2.242464144
MAE	0.011113481
R2	0.883783973

(b) M1 3

RMSE	0.017224297
MSE	0.000296676
MAPE	2.208480955
MAE	0.01112233
R2	0.893987027

(c) M1 4

RMSE	0.016504522
MSE	0.000272399
MAPE	1.960718136
MAE	0.009845696
R2	0.902662108

Table 14. Predicting moving average values metrics of the S&P500 dataset

(a) M1 1

RMSE	1.020056917
MSE	1.040516114
MAPE	0.055009158
MAE	0.34066933
R2	0.99999897

(b) M1 3

RMSE	1.016457032
MSE	1.033184897
MAPE	0.055034722
MAE	0.347453224
R2	0.999998977

(c) M1 4

RMSE	0.922520875
MSE	0.851044764
MAPE	0.048713757
MAE	0.303149401
R2	0.999999157

In that area, it is less than 1%, which is not considerable, so our multi-feature model works well in this case.

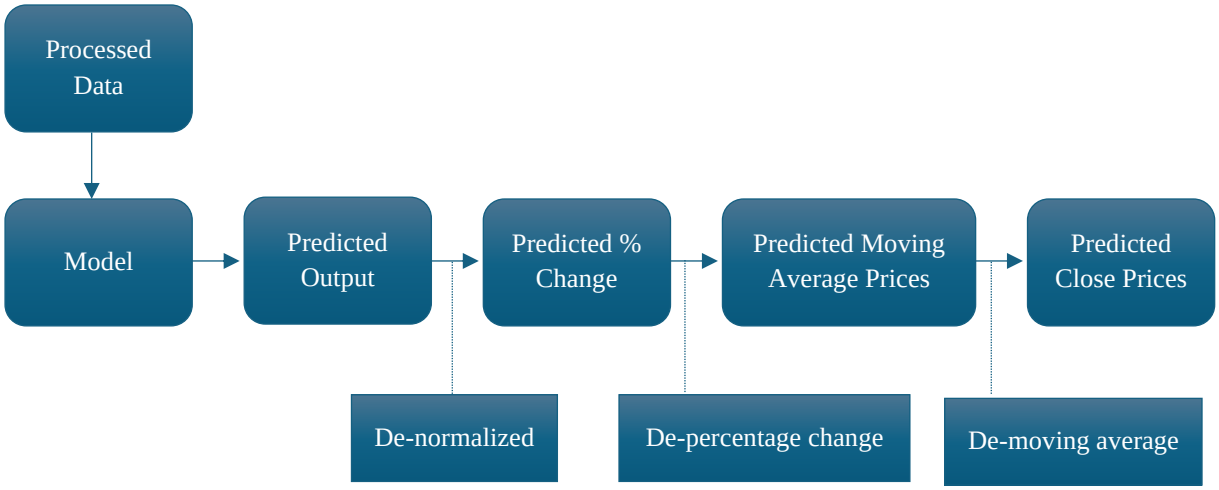

Fig. 11 Decode predicted output pipeline

In terms of comparing M1 2 and M1 4, we can say that the multi-feature one (M1 5) is better in the real world - when people usually look for the trend of a stock rather than at its real prices. However, both play very good predictions. Their trade-offs are tiny, so we can pick any of them to use.

4.13. Deep evaluation with respect to S&P500

In terms of S&P500, we can use M1 1, M1 3, M1 4. This model contains S&P500 as a feature in them. Base on 13, 14, 15 and 16. We can proudly say that with respect to S&P500, model M1 5 outperform the others.

4.14. Deep evaluation with respect to Exxon Mobil

In terms of Exxon Mobil, we can use M2 1 and M2 3. This model contains Exxon Mobil as a feature in them. Base on 17,18,19, and 20. We can proudly say that with respect to Exxon Mobil, model M2 3 outperform the other one. Despite the fact that M2 1 is slightly better at predicting trends, the difference between M2 1 and M2 3

4.15. Deep evaluation with respect to Chervon

In terms of Chervon, we can use M2 2 and M2 3. This model contains Chervon as a feature in them. Based on 21,22,23, and 24. We can proudly say that, with respect to Chervon, M2 _2 and M2 _3 only have better in total. Our multi-feature model is less than the other one at most, about 1% - that is not significant in real life problems.

4.16. Comparing to SOTA models

We will follow all the pre-processing steps before putting the data into the LSTM model and RNN model, as well as all the decoding steps, to ensure the most fair competition between models.

4.17. With Respect to NASDAQ

Based on results from our model (9,10,11, and 12) and SOTA models (25,26,27, and 28). We can point out that our multi-feature model (M1 _4) completely outperforms the LSTM model and RNN model.

Table 15. Predicting close price metrics of the S&P500 dataset

(a) M1

RMSE	14.27815877
MSE	203.8658177
MAPE	0.773167971
MAE	4.767608709
R2	0.999798843

(b) M1 3

RMSE	14.22776968
MSE	202.4294301
MAPE	0.771584123
MAE	4.86254813
R2	0.99980026

(c) M1 4

RMSE	12.91290642
MSE	166.7431522
MAPE	0.683400608
MAE	4.242523745
R2	0.999835472

Table 16. Predicting trend metrics of the S&P500 dataset

(a) M1 1

Acc	0.8911	
Pre	0.86583744	0.90993943
Recall	0.88077678	0.90356932
F1	0.87324322	0.90674319

(b) M1 3

Acc	0.8914	
Pre	0.86586099	0.9103421
Recall	0.87746126	0.90150512
F1	0.87162253	0.90590206

(c) M1 4

Acc	0.8975	
Pre	0.87868519	0.91116987
Recall	0.87725827	0.912242
F1	0.87797115	0.91170562

4.18. With Respect to Exxon Mobil

Based on the result from our model (17,18,19, and 20) and SOTA models (29,30,31, and 32). Again, the same conclusion goes here: our multi-feature model (M2 /3) utterly surpasses the LSTM and RNN models.

4.19. Visualization

Note: In 12 and 13, Accuracy and R2-score are performance-related metrics, and others (MAPE, MAE, RMSE, and MSE) are error-related metrics.

5. Comparison with State-of-the-Art Techniques

The proposed Multi-Feature Time2Vec-Transformer model demonstrates superior performance compared to state-of-the- art models such as LSTM and RNN. Several factors contribute to this improvement:

5.1. Integration of Multi-Feature Correlations

By leveraging correlated market features (e.g., NASDAQ and S\&P500 for Group 1), the model effectively captures interdependencies and complex relationships. This approach allows for a more nuanced understanding of stock price movements compared to single-feature models.

5.2. Temporal Representation with Time2Vec

The inclusion of Time2Vec enhances the model's ability to encode both periodic and non-periodic patterns in time-series data. This representation is particularly effective in addressing challenges posed by sparse or noisy financial data.

5.3. Attention Mechanism for Long-Term Dependencies

Unlike RNN and LSTM, which struggle with vanishing gradients and long-term dependencies, the Transformer's multi-head attention mechanism excels at capturing global patterns over extended sequences. This is evident in the improved metrics for models M1_4 and M2_3 across various datasets.

5.4. Regularization and Robustness

The use of dropout layers prevents overfitting, while residual connections in the architecture ensure stable training. These design choices contribute to the model's robustness and generalizability. Quantitatively, the Multi-Feature Time2Vec-Transformer outperformed baseline models in key evaluation

metrics such as MAE, RMSE, and R^{2}. For instance, in the NASDAQ dataset, the M1_4 model achieved a test MAE of 0.0119 compared to 0.0137 for the LSTM model, reflecting a 13.1\% improvement. Similarly, the test RMSE of 0.0212 for M1_4 indicates enhanced prediction accuracy. Qualitatively, the proposed model's ability to adapt to extreme market conditions and incorporate diverse market signals underscores its practical relevance. These results suggest that the Multi-Feature Time2Vec-Transformer model is well-suited for dynamic financial environments, offering significant advantages over traditional methods.

6. Summary

This thesis explored deep learning for stock price prediction, focusing on correlation-based features and a novel neural network architecture combining the Transformer model with Time2Vec as an Encoder. The results demonstrated that the proposed model outperforms simpler models (LSTM, RNN) and single-feature approaches, owing to its ability to integrate diverse features effectively. These findings emphasize the importance of feature selection and model complexity in enhancing predictive accuracy. The proposed approach provides a robust framework for practical stock price forecasting, offering valuable insights for smarter investment decisions and improved risk management in dynamic financial markets.

Table 17. Predicting normalized values metrics of the Exxon dataset

(a) M2 1

RMSE	0.023179541
MSE	0.000537291
MAPE	2.290878355
MAE	0.013910819
R2	0.801418014

(b) M2 3

RMSE	0.02010469
MSE	0.000404199
MAPE	2.158960848
MAE	0.01338959
R2	0.85060855

Table 18. Predicting moving average values metrics of the Exxon dataset

(a) M2 1

RMSE	0.048242004
MSE	0.002327291
MAPE	0.075914309
MAE	0.01831591
R2	0.999996625

(b) M2 3

RMSE	0.03920276
MSE	0.001536856
MAPE	0.073085054
MAE	0.016113275
R2	0.999997771

Table 19. Predicting close price metrics of the Exxon dataset

(a) M2

RMSE	0.675197252
MSE	0.455891329
MAPE	1.061601097
MAE	0.256277127
R2	0.999341621

(b) M2 3

RMSE	0.548681928
MSE	0.301051858
MAPE	1.020793064
MAE	0.225457042
R2	0.999565233

Table 20. Predicting trend metrics of the Exxon dataset

(a) M2 1

Acc	0.8794	
Pre	0.85236517	0.90044542
Recall	0.86948059	0.88686216
F1	0.86083781	0.89360217

(b) M2 3

Acc	0.8746	
Pre	0.84454201	0.89820924
Recall	0.86721311	0.88009851
F1	0.85572743	0.88906165

Table 21. Predicting normalized values metrics of the Chervon dataset

(a) M2 2

RMSE	0.01682611
MSE	0.000283118
MAPE	1.929007573
MAE	0.010853351
R2	0.887236384

(b)M2 3

RMSE	0.017909977
MSE	0.000320767
MAPE	2.090819573
MAE	0.011799609
R2	0.872240963

Table 22. Predicting moving average values metrics of the Chervon dataset

(a)M2 2

RMSE	0.062136657
MSE	0.003860964
MAPE	0.073510284
MAE	0.021515203
R2	0.999997359

(b)M2 3

RMSE	0.061813397
MSE	0.003820896
MAPE	0.07992873
MAE	0.022514326
R2	0.999997386

Table 23. Predicting close price metrics of the Chervon dataset

(a) M2 2

RMSE	0.869664813
MSE	0.756316888
MAPE	1.031343171
MAE	0.301040856
R2	0.999484581

(b) M2 3

RMSE	0.865140471
MSE	0.748468035
MAPE	1.123356882
MAE	0.315020592
R2	0.99948993

Table 24. Predicting trend metrics of the Chervon dataset

(a) M2 2

Acc	0.8841	
Pre	0.85960136	0.90432272
Recall	0.88110425	0.88647799
F1	0.87021999	0.89531145

(b) M2 3

Acc	0.8798	
Pre	0.85623658	0.89910457
Recall	0.87424016	0.88418901
F1	0.86514472	0.89158441

Table 25. Predicting normalized values metrics of the NASDAQ dataset

(a) LSTM

RMSE	0.022196892
MSE	0.000492702
MAPE	2.385642716
MAE	0.014343534
R2	0.876628911

(b) RNN

RMSE	0.047336384
MSE	0.002240733
MAPE	6.698210951
MAE	0.041594085
R2	0.438909067

Table 26. Predicting moving average values of the NASDAQ dataset

(a) LSTM

RMSE	5.125561613
MSE	26.27138185
MAPE	0.065215182
MAE	2.026053739
R2	0.999997954

(b) RNN

RMSE	9.676503936
MSE	93.63472843
MAPE	0.188923519
MAE	5.049059684
R2	0.999992717

Table 27. Predicting close price metrics of the NASDAQ dataset

(a) LSTM

RMSE	71.73388027
MSE	5145.749579
MAPE	0.912244328
MAE	28.34579584
R2	0.999601079

(b) RNN

RMSE	135.4257838
MSE	18340.1429
MAPE	2.624377079
MAE	70.63959959
R2	0.998580178

Table 28. Predicting trend metrics of the NASDAQ dataset

(a) LSTM

Acc	0.891	
Pre	0.85934969	0.91332924
Recall	0.87460378	0.90225954
F1	0.86690964	0.90776065

(b) RNN

Acc	0.9027	
Pre	0.86621869	0.92918703
Recall	0.89894764	0.90521942
F1	0.88227974	0.91704665

Table 29. Predicting normalized values metrics of the Exxon dataset

(a) LSTM

RMSE	0.021515314
MSE	0.000462909
MAPE	2.475415234
MAE	0.015313505
R2	0.828910244

(b) RNN

RMSE	0.044859558
MSE	0.00201238
MAPE	6.108091144
MAE	0.039855796
R2	0.256180494

Table 30. Predicting moving average values of the Exxon dataset

(a) LSTM

RMSE	0.042892362
MSE	0.001839755
MAPE	0.083627915
MAE	0.018872521
R2	0.99999734

(b) RNN

RMSE	0.083857919
MSE	0.007032151
MAPE	0.217348967
MAE	0.045318649
R2	0.999989843

Table 31. Predicting close price metrics of the Exxon dataset

(a) LSTM

RMSE	0.600321653
MSE	0.360386087
MAPE	1.174699108
MAE	0.264064471
R2	0.999480945

(b) RNN

RMSE	1.173675761
MSE	1.377514792
MAPE	3.018434554
MAE	0.634098928
R2	0.998018214

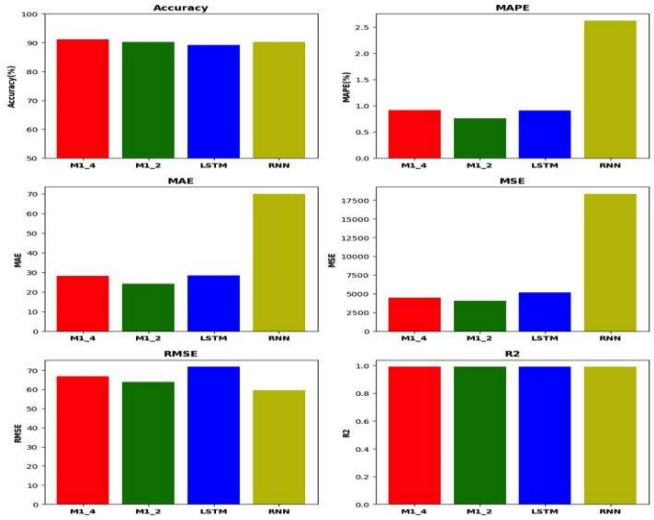

Fig. 12 Comparing metrics among Multi-feature model, Single-feature model, LSTM model, and RNN model using NASDAQ as a target

Table 32. Predicting trend metrics of the Exxon dataset

(a) LSTM

Acc	0.8586	
Pre	0.82496029	0.88531568
Recall	0.8509832	0.86433244
F1	0.83776971	0.87469823

(b) RNN

Acc	0.8728	
Pre	0.84051938	0.89851641
Recall	0.86831866	0.87618315
F1	0.8541929	0.88720926

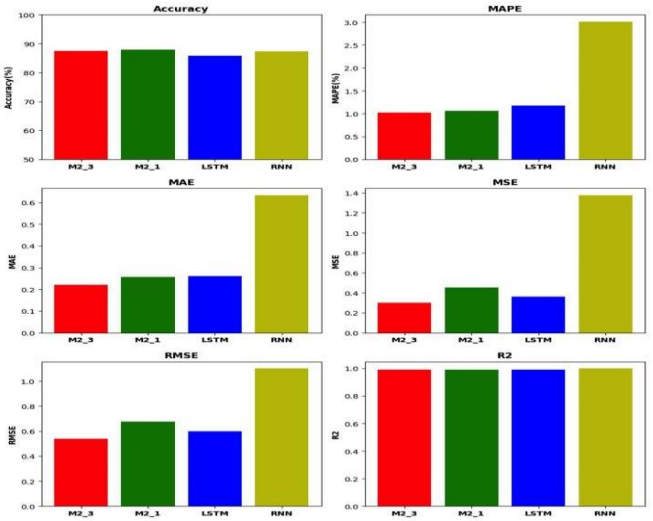

Fig. 13 Comparing metrics among Multi-feature model, Single-feature model, LSTM model, and RNN model using Exxon Mobil as a target

References

[1] Lasse Heje Pedersen, *Efficiently Inefficient: How Smart Money Invests and Market Prices Are Determined*, Princeton University Press, pp. 1-368, 2019. [Google Scholar] [Publisher Link]

[2] Everette S. Gardner, and E.D. McKenzie, "Forecasting Trends in Time Series," *Management Science*, vol. 31, no. 10, pp. 1237-1246, 1985. [CrossRef] [Google Scholar] [Publisher Link]

[3] Artemios-Anargyros Semenoglou et al., "Investigating the Accuracy of Cross-Learning Time Series Forecasting Methods," *International Journal of Forecasting*, vol. 37, no. 3, pp. 1072-1084, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Sepp Hochreiter, and Jürgen Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997. [CrossRef] [Google Scholar] [Publisher Link]

[5] Shiyang Li et al., "Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting," *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019. [Google Scholar] [Publisher Link]

[6] Ashish Vaswani et al., "Attention Is All You Need," *ArXiv*, pp. 1-15, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[7] Neural Networks from Scratch, Victor Zhou, 2019. [Online]. Available: victorzhou.com/series/neural-networks-from-scratch/

[8] Souhaib Ben Taieb, Antti Sorjamaa, and Gianluca Bontempi, "Multiple-Output Modeling for Multi-Step-Ahead Time Series Forecasting," *Neurocomputing*, vol. 73, no. 10-12, pp. 1950-1957, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[9] Massimiliano Marcellino, James H. Stock, and Mark W. Watson, "A Comparison of Direct and Iterated Multistep AR Methods for Forecasting Macroeconomic Time Series," *Journal of Econometrics*, vol. 135, no. 1-2, pp. 499-526, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[10] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, "Learning Representations by Back-Propagating Errors," *Nature*, vol. 323, no. 6088, pp. 533-536, 1986. [Google Scholar] [Publisher Link]

[11] Tianyu Wang et al., "From Model-Driven to Data-Driven: A Review of Hysteresis Modeling in Structural and Mechanical Systems," *Mechanical Systems and Signal Processing*, vol. 204, pp. 1-39, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Understanding LSTM Networks, Colah's Blog, Github.io, 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[13] Jia Wang et al., "Clvsa: A Convolutional Lstm Based Variational Sequence-To-Sequence Model with Attention for Predicting Trends of Financial Markets," *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence Organization (IJCAI)*, pp. 3705-3711, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[14] Fuli Feng et al., "Enhancing Stock Movement Prediction with Adversarial Training," *ArXiv*, pp. 1-7, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[15] Seyed Mehran Kazemi et al., "Time2vec: Learning A Vector Representation of Time," *ArXiv*, pp. 1-16, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[16] Jacob Devlin et al., "BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171-4186, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] Qingsong Wen et al., "Transformers in Time Series: A Survey," *ArXiv*, pp. 1-9, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] RMSProp - Cornell University Computational Optimization Open Textbook - Optimization Wiki, Cornell.edu, 2022. [Online]. Available: https://optimization.cbe.cornell.edu/index.php?title=RMSProp

[19] AdaGrad - Cornell University Computational Optimization Open Textbook - Optimization Wiki, Cornell.edu, 2022. [Online]. Available: https://optimization.cbe.cornell.edu/index.php?title=AdaGrad

[20] Jian-Hang Li et al., "Multi-Head Attention Based Hybrid Deep Neural Network for Aeroengine Risk Assessment," *IEEE Access*, vol. 11, pp. 113376-113389, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] Mandella Ali M. Fargalla et al., "Timenet: Time2vec Attention-Based Cnn-Bigru Neural Network for Predicting Production in Shale and Sandstone Gas Reservoirs," *Energy*, vol. 290, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Shunrong Shen, Haomiao Jiang, and Tongda Zhang, "*Stock Market Forecasting Using Machine Learning Algorithms*," Department of Electrical Engineering, Stanford University, pp. 1-5, 2012. [Google Scholar] [Publisher Link]

[23] Yahoo Finance, Yahoo, 2024. [Online]. Available: https://finance.yahoo.com/

[24] Residual Neural Network, Wikipedia.org, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Residual_neural_network