Predicting Stock Closing Prices in Emerging Markets with Transformer Neural Networks: The Saudi Stock Exchange Case

Nadeem Malibari[1], Iyad Katib[2], Rashid Mehmood[3]
Department of Computer Science, Faculty of Computing and Information Technology[1,2]
High Performance Computing Center[3]
King Abdulaziz University, Jeddah 21589, Saudi Arabia

Abstract—**Deep learning has transformed many fields including computer vision, self-driving cars, product recommendations, behaviour analysis, natural language processing (NLP), and medicine, to name a few. The financial sector is no surprise where the use of deep learning has produced one of the most lucrative applications. This research proposes a novel fintech machine learning method that uses Transformer neural networks for stock price predictions. Transformers are relatively new and while have been applied for NLP and computer vision, they have not been explored much with time-series data. In our method, self-attention mechanisms are utilized to learn nonlinear patterns and dynamics from time-series data with high volatility and nonlinearity. The model makes predictions about closing prices for the next trading day by taking into account various stock price inputs. We used pricing data from the Saudi Stock Exchange (Tadawul) to develop this model. We validated our model using four error evaluation metrics. The applicability and usefulness of our model to fintech are demonstrated by its ability to predict closing prices with a probability above 90%. To the best of our knowledge, this is the first work where transformer networks are used for stock price prediction. Our work is expected to make significant advancements in fintech and other fields depending on time-series forecasting.**

***Keywords*—Stock price prediction; time-series forecasting; transformer deep neural networks; Saudi Stock Exchange (Tadawul); financial markets**

I. Introduction

We have come a long way in developing our societies, improving and optimising every task and thing we do, and artificial intelligence (AI) is at the heart of these endeavours [1], [2]. Machine and deep learning-based AI has revolutionised many aspects of our daily activities, be it healthcare [3], [4], transportations [5], [6], big data [7], distance learning [8], disaster management [9], risk prediction in aviation systems [10], DNA profiling [11], smart cities [12], [13], and more. The use of machine and deep learning in the financial sector is one of the most lucrative tasks. Forecasting time-series data is an important topic that plays a key role in analysis, decision-making, and resource management in many industrial sectors. For example, in the financial sector, forecasting based on historical data can be helpful for investors in maximizing return and reducing risk on investments [14], [15]. Many works have been reported on the use of AI for the financial sector, such as the use of multilayer perceptrons (MLP) for NSADA stock index [16], the use of stacked autoencoders for US stock forecasting [17], and the use of Long short-term memory network (LSTM) to predict the closing prices of iShares MSCI United Kingdom index [18] (for further motivation on the subject, see Section II).

A time-series forecast is a way of determining future values based on historical experience. Correlational data is used for this process, either time-based correlations (years, months, weeks, etc.) or sequential correlations, for gaining insights that inform decisions. A range of methods has been developed to predict, ranging from traditional to machine-learning approaches. Despite their wide usage, traditional time-series prediction methods such as auto-regression (AR), Seasonal Naïve, ETS, and integrated moving average ARIMA are designed to fit each time-series separately [19]. Moreover, practitioners should learn how to select specific trends, seasonal components, and other data components manually, especially for financial data series with highly nonlinear and fluctuating data. These drawbacks have limited their applications in advanced large-scale time-series prediction tasks.

The challenges mentioned above can be overcome by algorithms that can capture the patterns in the data and the dynamics underlying them. In deep neural networks, continuous developments have led to breakthroughs that are proposed as another alternative. An array of deep neural network architectures has been applied to time-series models to understand trends and patterns by learning from ground truth data. However, many challenges remain. For example, while a Recurrent Neural Network (RNN) can model and process sequential and time-series data, its gradient vanishing and exploding properties prevent them from detecting long-term dependencies (relationship between entities that are several steps apart). In real-world forecasting, there are long-term and short-term repeating patterns [20], which means that complex RNN models are required to analyze long-term time series and study long-term effects. Therefore, long short-term memory (LSTM) models have been proposed to improve the standard RNN model for time series analysis. Theoretically, they are explicitly geared towards minimizing long-term dependency problems. However, according to [21], LSTM has an adequate context size of 200 tokens on average, but they are only able to distinguish 50 tokens within a context, suggesting that even it is incapable of capturing long-term trends. Furthermore, RNNs and all their variants use mostly sequential operations, thus cannot benefit from the performance advantages offered by modern GPUs.

Rather than RNNs, the next big step was a completely new architecture – Transformer [22] utilizes attention mechanisms that leverage self-attention mechanisms to process the entire sequence of data. The transformer architecture is the most prevalent model for natural language modelling and has proven quite successful in several other applications. The complexity of the space corresponding to self-attention can grow quadratically as sequence length increases; for this reason, self-attention cannot be extended to extremely long sequences [20]. The quadratic complexity of computing poses a significant challenge when forecasting time series with long-term solid dependence and fine granularity. Researchers had the same challenges adapting transformers from language to computer vision applications due to pictures containing more significant amounts of information than sentences. However, they are able to replace this quadratic computational complexity with a linear computational complexity to image size.

In this work, we specifically delve into adapting the computer vision transformer model [23] to time series forecasting. We propose a novel fintech machine learning method that uses Transformer neural networks for stock price predictions. In our method, self-attention mechanisms are utilized to learn nonlinear patterns and dynamics from time-series data with high volatility and nonlinearity. Our Contributions follow.

- We propose a novel predictive Transformer based model with divided time series data into patches for predicating future value. Regardless of how complex a situation is, our proposed method can discover the broad conditional probability distribution of the future values.
- The model makes predictions about closing prices for the next trading day by taking into account various inputs, Open, High, Low, Volume, and Closing Prices. We used pricing data from the Saudi Stock Exchange (Tadawul) to develop this model. We validated our model using a range of metrics; Mean Absolute (MAE), Square (MSE), Root MSE (RMSE), and Percentage (MAPE) Error.
- The applicability and usefulness of our model to fintech are demonstrated by its ability to predict closing prices with a probability above 90%.

Novelty: As mentioned earlier, transformers are relatively new and while these have been applied for NLP and computer vision, they have not been explored much with time-series data. Our work is expected to make significant advancements in fintech and other fields depending on time-series forecasting. To the best of our knowledge, this is the first work where transformer networks are used for stock price prediction.

The structure of this paper is as follows. We discuss related research in Section II as well as past innovations using deep learning for stock forecasts. The methodology for this study, the dataset, and the transformer model with divided space are described in Section III. This section also provides details of data prepossessing, and hyperparameters selection. Section IV provides the results and analysis. Section V concludes and provides directions for the future work.

II. Related Work

In the not-so-distant past, Neural Networks (NN) were criticized by many forecasting practitioners as not suitable and not being competitive in forecasting fields [24]. Consequently, practitioners have usually selected statistical methods that were considered more straight forward to apply [19]. However, with the ever-increasing availability of data, neural networks (NNs) and deep learning have revolutionized and achieved remarkable success in many research fields and practical scenarios, including medical predictions, NLP, image recognition, etc. Because of their capabilities to identify complex nonlinear patterns and explore unstructured relationships without hypothesizing them a priori. These technological breakthroughs have attracted significant attention from the enthusiasts' researcher community presenting many complex novel NN architectures on time series forecasting. Over recent decades, plenty of works and research exist where deep learning is used for forecasting. There is a possibility to predict stock price changes and foreign exchange rates according to [14]. As a result, AI applications are becoming increasingly popular among investors to increase returns and reduce the risk [15].

Selvin et al. [25] illustrated how deep neural network architectures can capture hidden dynamics and can be used to forecast. Guresen, Kayakutlu, and Daim [16] predicts the NASDA stock index by using multilayer perceptrons (MLP), dynamic, and hybrid artificial neural networks. Using a stacked autoencoder and deep neural network, Takeuchi and Lee [17] obtains an accuracy of 53.36 % when predicting the US stock direction.

Since their inception in 2014 by Hochreiter and Schmidhuber [26], the Long short-term memory network (LSTM) introduced is a variation of the Recurrent neural network model (RNN), which is the most commonly used architecture for sequence prediction problems [8]. In contrast to RNN, LSTM networks are capable of detecting long-term dependency and can prevent gradient vanishing. It utilizes historical information via the input, forget and output gates. In their study, Nikou, Mansourfar, and Bagherzadeh [18] predict the closing prices of iShares MSCI United Kingdom index using an LSTM model. The model performed significantly better than the ANN, Support Vector Regression (SVR), and RF models. LSTMs are utilized in another study by [27] in order to forecast future stock returns. Also, an Autoregressive Integrated Moving Average (ARIMA) and an LSTM model were utilized to improve forecast accuracy [28]. According to Nelson, Pereira, and De Oliveira [29], the average accuracy for predicting the direction of some stocks traded on the Brazilian stock exchange could reach up to 55.9% with the LSTM model.

Because of its powerful pattern recognition ability, the convolutional neural network (CNN) is a variation of the multilayer perceptron (MLP). Its use has extended increasingly for time-series forecasting. The work by [30], [31], and [32] used CNN to predict stock trends. Ugur Gudelek, Arda Boluk, and Murat Ozbayoglu [32] have also experimented with 2D CNN for trend detection. The model performance evaluation has 72% accuracy values and looks promising.

A comparison study of differences between Multi-layer Perceptron (MLP), Convolutional Neural Network, and Long Short-Term Memory (LSTM) was performed by [33]. They

TABLE I. SUMMARY OF RELATED WORKS

Research	Model Architecture								Dataset	Accuracy
	ANN	MLP	RF	SVR	RNN	CNN	LSTM	Transformer		
Selvin et al. [25]					✓	✓	✓		1721 NSE	2.36
Guresen et al. [16]	✓	✓							NASDAQ	MSE: 1472
Nikou et al. [18]	✓		✓	✓			✓		iShares MSCI	MSE: 0.09396
Naik et al. [27]					✓		✓		Indian NSE	RMSE: 23.78
Nelson et al. [29]							✓		BMF Bovespa	55.9%
Gudelek et al. [32]						✓			ETF	72%
Persio et al. [33]		✓				✓	✓		SP500	MSE: 0.2491
Our Study								✓	Tadawul	90%

adopted a sliding window approach, using 30 previous days to predict the value of the 31st day using historical data of the S&P500 index from 1950 to 2016. CNN's results are exceptional even without the use of additional features such as technical analysis.

Recently, the well-known self-attention-based Transformer [22] was proposed for sequence modeling is the most prevalent model for natural language modeling and has proven quite successful in several other applications such as as translation, speech, image generation, and music [22], [34], [35]. The extension of self-attention to extremely long sequences would, however, be computationally prohibitive since space complexity increases quadratically with the sequence length [20]. However, Vision Transformer (ViT) [23] and TimeSformer (Time-Space Transformer) [36] offer entirely new architectures for image classification, and video understanding based solely on Transformers eliminating the problems associated with long sequences. In particular, ViT divides an image into patches (also called tokens) with fixed length; then following the practice of using transformers to model language, ViT then uses transformer layers to model the relationship among tokens for classification. The TimeSformer, on the other hand, translates the input video into a sequence of image patches derived from the individual frames. The model then captures the semantic information about each patch through comparison with those of the other patches. This allows TimeSformer to capture the space-time dependency based on the whole video. Transformers' recent success in natural language processing (NLP) has motivated researchers to implement this model in computer vision applications and tasks.

Table I summarises the related works discussed in this section. It lists the various ML models that the researchers have used for stock price prediction along with the respective datasets used and the model accuracies reported in the respective works. The most commonly used architecture for problems involving stock price prediction is the LSTM. It can detect long-term dependency and prevent gradient vanishing to some extent. However, LSTM accuracy is much less than the convolutional neural network (CNN) because of CNN's powerful pattern recognition ability. The accuracy metrics are reported in the table if these are provided by the researchers, otherwise, we reported the numeric value from the article without the accuracy metric name. As shown in the table, the best result achieved is 72% accuracy. Our transformer model with its attention features has provided 90% or higher accuracy. We have kept the content in the table to the minimum due to the space issue, please refer to the listed works for details.

III. METHODOLOGY, DATASETS AND MODEL DESIGN

The objective of our study is to predict the subsequent and future closing of the trades in the Saudi Stock Exchange (Tadawul). We use a transformer-based temporal model architecture. In this section, we describe our methodology, Transformer neural network model design, datasets, preprocessing, and validation metrics.

We first present an overview of our methodology in Section III-A. The transformer-based temporal model architecture is described in Section III-B. The the Saudi Stock Exchange (Tadawul) datasets are explored in Section III-C. The data modelling methodology using transformer neural networks is summarised in Section III-D. In Section III-E, we describe the preparation of the dataset, including data splitting, normalization, and feature selection. We discuss the hyperparameter configuration for the model. Section III-F describes the concept of sliding window for framing the dataset. Section III discusses hyperparameter configuration of our model.

A. Methodology Overview

The overall methodology we have adopted is depicted in Fig. 1. It consists of seven main phases as highlighted in the figure. The first process involved extracting Saudi Stock Exchange (Tadawul) data, followed by data cleaning and normalization. As a result of this procedure, we only get data that is appropriate for machine learning algorithms. We then select the four features (open, low, high, previous closing) that the model will use. Thereafter, the data are sorted into non-overlapping batches, which are then fed into the model until performance measures are optimized. Ultimately, the optimized model is used to forecast future closing prices for unseen stock data.

B. Transformer Neural Network Architecture

A significant influence on our architecture is a vision transformer (ViT) [36] using divided space. The vision transformer (ViT) is among the first attempts to apply the outstanding performance of Transformers [22] to image classification tasks rather than natural language processing. The vision transformer (ViT) model, which comprises three main elements: a linear layer for patch embedding, a stack of transformer blocks with multi-head self-attention and feed-forward layers, and a linear layer classification score prediction.

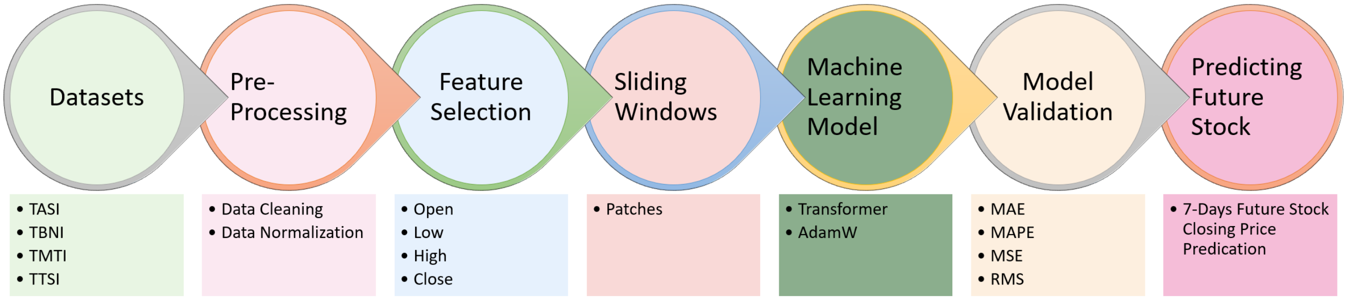


Fig. 1. An Overview of Our Methodology.

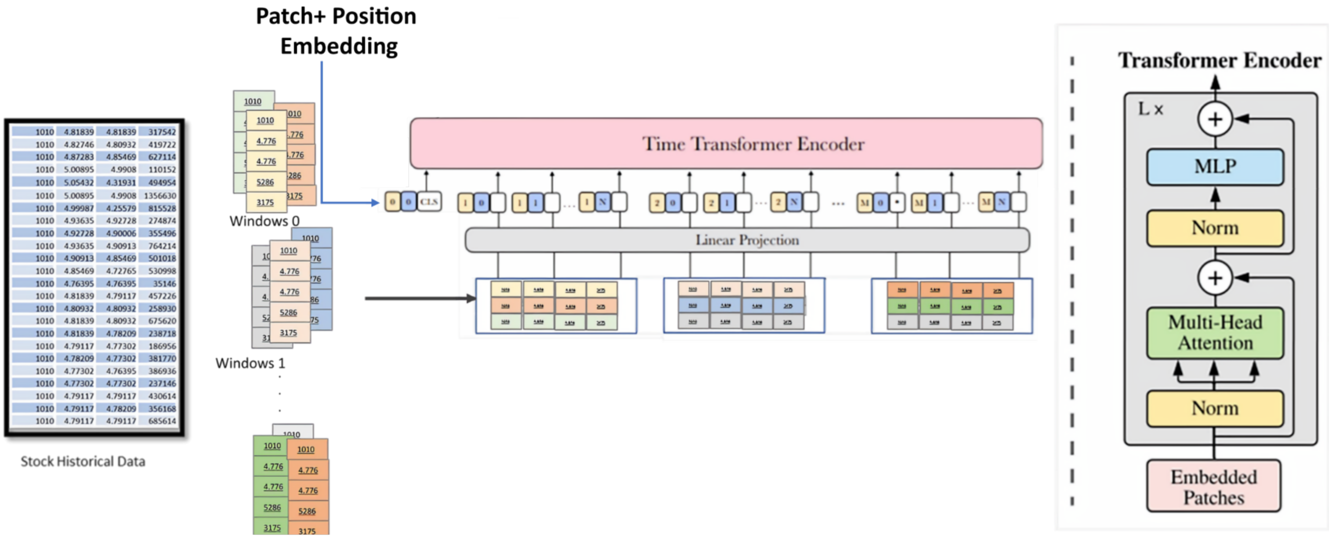


Fig. 2. Proposed Transformer Encoder Model Overview (Illustration Inspired by [22]).

An overview of our suggested model is depicted in Fig. 2. The Vision Transformer (ViT) model serves as the basis for our predictive model. Our suggested model added one more component to ViT architecture. Its primary purpose is to create sliding windows from historical data. Since daily trading volumes on the stock market are substantial, historical data on the market can be challenging to manipulate, and manipulating it can cause a computational burden. Furthermore, the effect of more recent data on a training model is greater than that of older data [37]. Braverman et al. [38] developed a sliding-window method that utilizes recent data while disregarding older observations to solve this problem.

The range of data of interest is selected using a window. The sliding window represents a period that stretches backward in time from the present to the past. The sliding window is held steady (the number of data stays constant), and only the window is moved. Resulting, the training data volume is reduced while maintaining the model's efficiency and general usability [37].

In summary, Fig. 2 depicts our proposed model as follows. The historical data is split into windows and then those windows are divided into fixed-size patches. Linear embeddings are then applied to the patches, followed by position embeddings. Then we feed the resulting sequence of vectors to the Transformer encoder. As a standard approach, we add an extra token to the sequence of learnable tokens to perform prediction. The Transformer encoder diagram in Fig. 2 was inspired by [22].

C. Datasets

The Saudi Stock Exchange (Tadawul) database contains stock trading information for more than 200 Saudi Arabian listed companies. The companies are grouped into sectors with different indices for each industry. The data we downloaded spans the period from 1993-01-02 through 2021-06-17 and consists of 772,189 trading days. Listed companies' and indices trading information includes their Open, High, Low, Volume, and Closing Price for each trading day. From the dataset, we extracted four indices to illustrate model capabilities and performance. These are Tadawul All Share Index (TASI), the Banks Index (TBNI), Materials Index (TMTI), and Telecommunication Services Index (TTSI).

Table II lists a small selection of the dataset. Specifically, it shows the trading information in the dataset for the TASI index for the period 1994-01-26 to 2021-07-01, which corresponds to 7311 trading days. The rows correspond to one trading day and contain the following features: the index column, the transaction date, the ticker code, High, Low, Volume, and Closing Price. Fig. 3 depicts the histograms of the closing price feature of the four datasets. There are four panels in the figure,

TABLE II. TASI Sample Data

	date	ticker	open	low	high	vol	close
0	1994-01-26	TASI	1751.71	1751.71	1751.71	312907	1751.71
1	1994-01-29	TASI	1751.71	1750.91	1751.71	204831	1750.91
—-	—-	—	—-	—-	—-	——	——
7310	2021-06-30	TASI	11002.74	10940.44	11009.70	374658538	10984.15
7311	2021-07-01	TASI	10987.13	10968.11	11006.66	352200486	10979.05

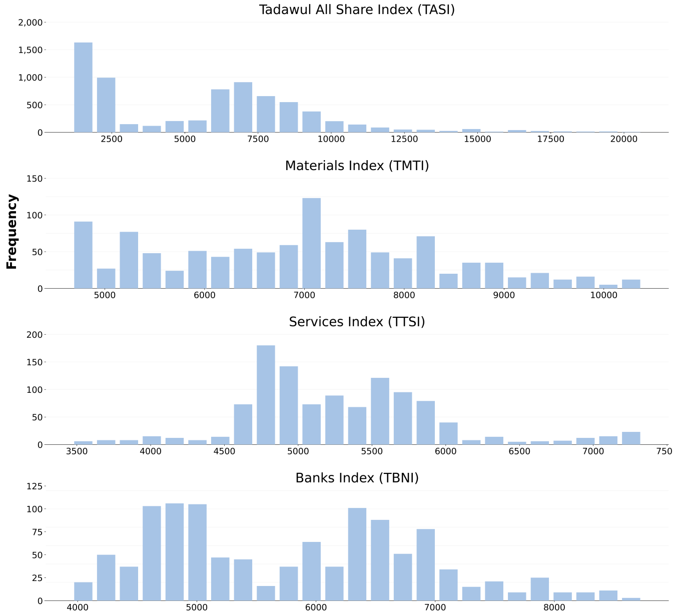


Fig. 3. The Histograms of the Closing Price for the Four Datasets.

each showing the histogram of its respective index. A display of the closing price is shown on the x-axis for each panel, grouped into 25 bins of equal width. Each bin is plotted as a bar whose height (the y-axis) indicates the number of closing prices (frequencies) occurrences in that bin.

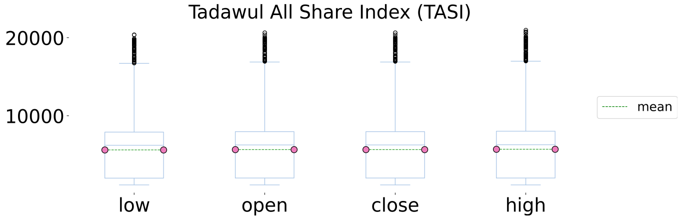


Fig. 4. TASI Boxplot.

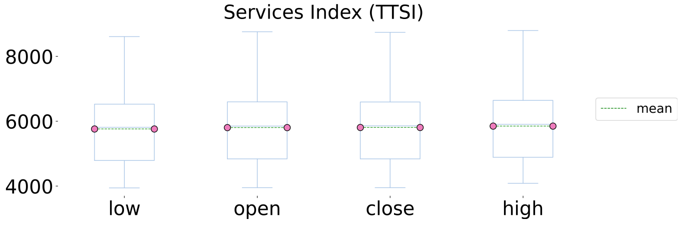


Fig. 5. TTSI Boxplot.

Fig. 4 to 7 depict the four features (open, low, high, close) of the four indices in our dataset as a boxplot. A boxplot is an in-depth statistical data analysis tool for gaining a broad perspective on the center and spread of the data distribution, which can assist with checking for errors and protecting other analyses. The median, interquartile range box, and whiskers are the primary elements of the boxplot to help understand the center and spread of the sample data. You'll see the green line representing the median in each box, which is the center of each feature. The interquartile range (the range between the third quartile and the first quartile) box, on the other hand, represents the middle 50% of the data and reflects how the data is distributed. The whiskers extend from both sides of the box (the bottom line is called lower whiskers, whereas the upper one is called higher whiskers). The whiskers denote the ranges for the bottom 25% and the top 25% of the data values, excluding outliers. Graphs that are skewed have the majority of data on the high or low side. Skewed graphs indicate that the data isn't normally distributed.

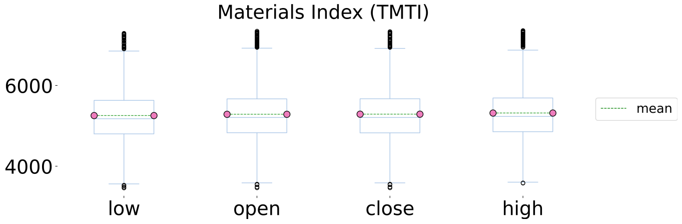


Fig. 6. TMTI Boxplot.

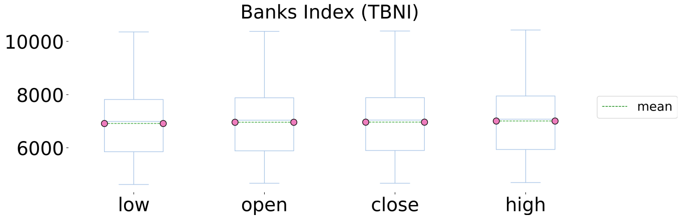


Fig. 7. TBNI Boxplot.

The data distribution for the TTSI, TMTI, and TBNI in the figures (Fig. 5 to 7) is almost normally distributed while it is positively skewed for the TASI index (Fig. 4). Moreover, any value greater than higher whiskers and less than lower whiskers values is an outlier and is represented in the figure as circles beyond the minimum and maximum values. Fig. 4, shows reasonable outliers points for TASI, which is expected as the closing of TASI is directly impacted by each and every listed company.

Fig. 8 highlights the correlation between the features (High, Low, Volume, and Closing Price), which is considered an essential step in the feature selection phase of data pre-

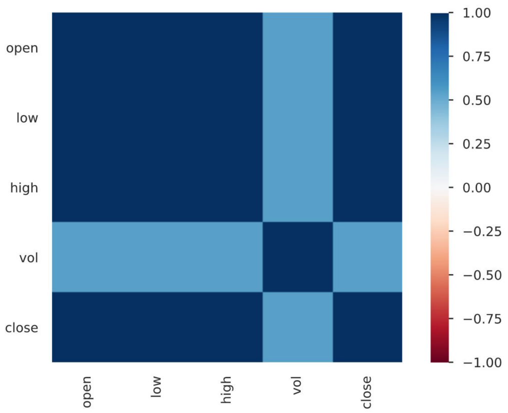


Fig. 8. TASI Correlation Matrix.

processing, especially if the data types of the features are continuous. As you can see in the figure, there is a high correlation between volume and the other features.

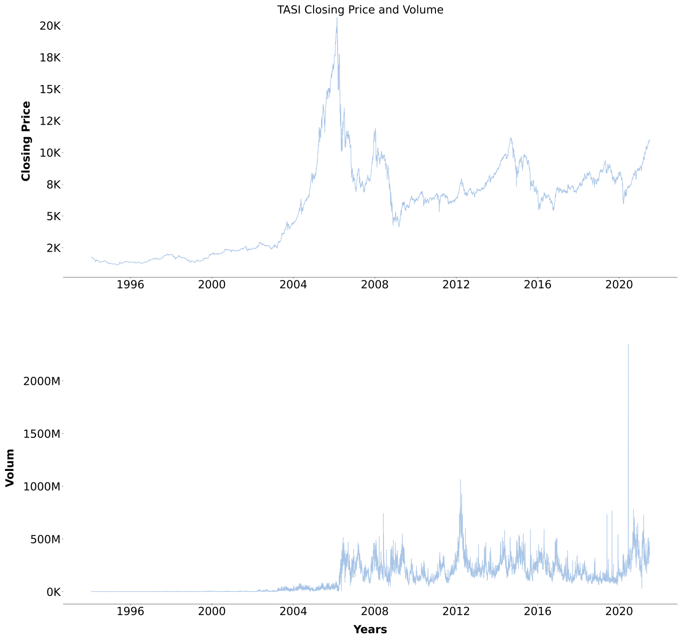


Fig. 9. TASI Closing Price and Volume.

Tadawul All Share Index (TASI) volume and closing figures are shown in Fig. 9.

D. Data Modelling Methodology

At first, the historical stock data of earlier days $X \in \mathbb{R}^{M \times F}$ consisting of M periods with F features (previous closing, opening, high ,low and volume) is split into a sequence of flattened 2D Windows $\mathbf{x}_w \in \mathbb{R}^{L \times F}$ of size $M - L$, where L is look back time interverls. Then the input window is divided into non-overlapping temporal patches of size $\mathbf{x}_p \in W \times (F \times 2)$.

Finally, following the protocol in ViT ,the patches $\mathbf{x}_p \in \mathbb{R}^{W \times (F \times 2)}$ are flattened forming a sequence of embeddings. Using learnable 1D position embeddings, we embed positional information into the patch embeddings so that all patches within a given window w are given the same temporal position. This allows the model to determine the temporal positions of patches.

E. Data Prepossessing

It is imperative to preprocess data in order to achieve good predictions. The indexes data were checked to determine whether the Tadawul Dataset contained inconsistencies. All the numerical data were normalized, and the missing values were removed. The open, high, low, volume and close prices were used to calculate the features, but information such as the stock code and stock name was omitted since they do not make sense. The following sections describe how the various preprocessing steps are implemented.

1) Splitting the Dataset: The training and test datasets are separated, similar to the ideas presented by [39]. We reserve apart from the end the training for validation from each time series. This approach is illustrated in the Fig. 10.

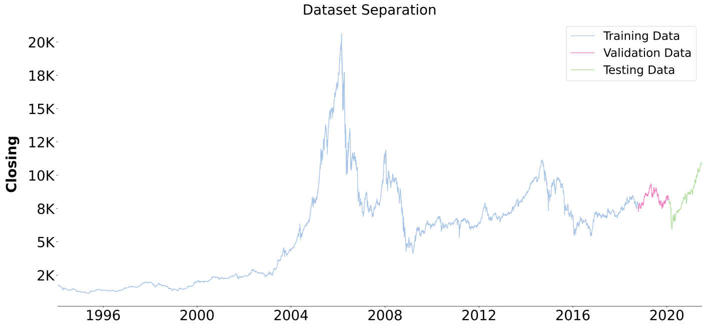


Fig. 10. Training, Validation and Testing Dataset Allocation.

2) Data Normalization: Normalization refers to the process of changing the range of values in a set of data. As we use prices and volume data, all the stock data must be within a typical value range. In general, machine learning algorithms converge faster or perform better when they are close to normally distributed and/or on a similar scale. Also, in a machine learning algorithm, the activation function, such as a sigmoid function, has a saturation point after which the outputs are constant [40]. As a result, when using model cells, the inputs should be normalized before being used. This process was done using MinMaxScaler methods of the scikit-learn library. When MinMaxScaler is applied to a feature, it subtracts the minimum value from each value in the feature and divides the range by the result. Thus, the range of a feature is the difference between the maximum and minimum values. In this way, MinMaxScaler preserves the shape of the original distribution. MinMaxScaler normalizes input values to be between [0,1].

3) Feature Selection: The downloaded data contains several features, including stock code, stock name, opening price, high price, low price, volume and closing price. Aside from some features that may not make any sense, these initial data have a lot of noise. For this reason, the data should be neglected when it is being trained. Based on [41] using open price, high price, low price, volume and close price, the input features will yield a satisfactory result. Therefore, we have selected the first five features as our input and have neglected irrelevant data like stock names and stock codes.

F. Divided Space

At this stage, we apply the concept of the sliding window for framing the dataset. With a window size of 2, we use the data before two days to predict the subsequent day closing. The process is repeated until all data are segmented. Then, the framing dataset is further split into patches.

G. Hyperparameter Selection

A number of parameters, called hyperparameters, are usually included in all deep learning models (apart from Naïve Bayes) that need to be adjusted to optimize results [42]. The various hyperparameters used during training are summarized in Table III. The AdamW optimizer is used during training with a learning rate of 0.001 and a weight decay of 0.0001. We train the model for 500 epochs with early stopping and dropout to prevent overfitting using TensorFlow [43] library.

TABLE III. Various Hyperparameters used in this Model with their Values

Hyperparameter	Value
Learning Rate	0.001
Optimizer	AdamW
Batch size	256
Epochs	500
Early stopping	Patience = 70 epochs Monitoring parameter = validation loss
Loss Function	MSE

H. Evaluation Metrics

Deep learning evaluation is categorized into accuracy index, financial index, and error-index [44]. Accuracy and financial index are widely used for prediction by classifying data (e.g., price direction prediction) and stock trading and portfolio management. On the other hand, error terms are frequently used in the evaluation for predicting numeric dependent variables (for instance, exchange rates or stock market predictions). The error terms evaluation rules compare the Real Data Y_{t+1} and the prediction data F_{t+1} using performance metrics: MAE, MSE, MAPE, and RMSE. Detailed information about the measures is provided below:

MSE is used to assess model performance based on the average error of forecasting. The formula of the MSE is given below:

$$\sum_{i=1}^{m} \frac{\left(Y_{t+i} - F_{t+i}\right)^2}{m} \tag{1}$$

RMSE is one of the most commonly used error metrics in regression. It is equal to the square root of the MSE. RMSE is a measure of how spread out the residuals are. Based on the RMSE formula, it is possible to determine how well the data was focused around the optimal line. The optimal RMSE value is close to zero.

$$\sqrt{\sum_{i=1}^{m} \frac{\left(Y_{t+i} - F_{t+i}\right)^2}{m}} \tag{2}$$

MAPE shows how much error was in the forecast. It measures how accurate the forecast is. A value of accuracy is calculated by subtracting the actual values from the average values of the previous period. The concept of MAPE is separated from the measurement level by data conversion. MAPE has minimal deviation in practice and cannot tell which direction the error is coming from. Ideally, MAPE should be close to zero. MAPE can be calculated using the following equation:

$$\frac{100}{m} \sum_{i=1}^{m} \frac{|Y_{t+i} - F_{t+i}|}{Y_{t+i}} \tag{3}$$

IV. Result and Discussion

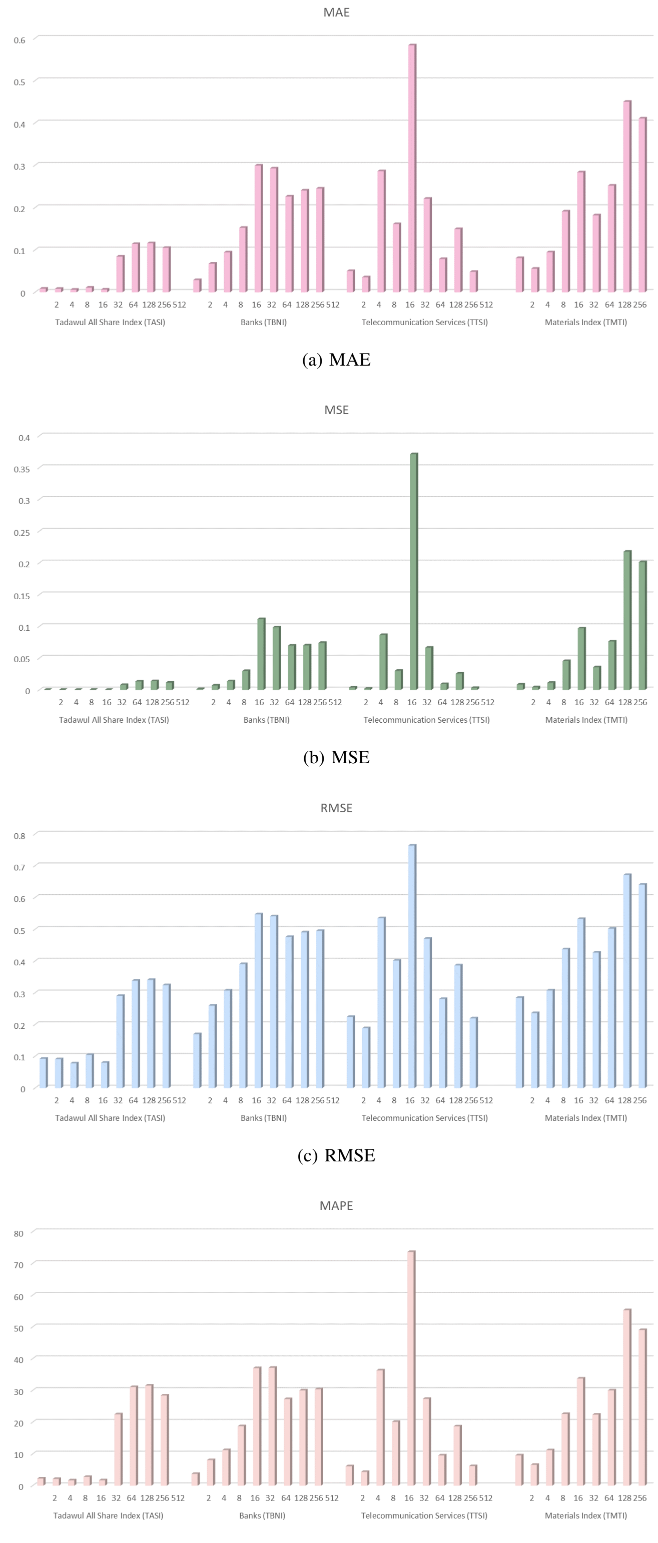
We now discuss the results beginning with results on model optimisation in Section IV-A, model validation in Section IV-B, and future stock closing price prediction in Section IV-C.

A. Model Optimisation

For this study, we experimented with different batch sizes and kept all the other hyper-parameters unchanged. Our predictive transformer model is implemented using TensorFlow written in Python. The study found that training smaller batches yielded better estimates but had a long training process. Our findings indicate that models perform better for all the four indices until the batch sizes reach around 4, with other batches not delivering significant performance improvements worth the time and effort devoted to estimating them. Fig. 11 depicts the four prediction performance measures MAE, MAPE, MSE, and RMSE results, respectively, for the Tadawul All Share Index (TASI), the Banks Index (TBNI), the Materials Index (TMTI), and the Telecommunication Services Index (TTSI).

Fig. 11a Mean Absolute Error (MAE) measure. It shows the batch size of the Tadawul All Share Index(TASI) is increased, and we find that forecast measures enhance until it reaches its best results at a batch size of 8 at a value of 0.0001, while it gets fluctuated for the other indices. Taking the MSE for Banks (TBNI) Index as an example, the optimal value for the TBNI index at batch size 2 is 0.0013, and it increases to 0.1114 at batch size 32. Thereafter, the index decreases with batch size. Similarly, when batch size over batch size exceeds eight, the Materials Index (TMTI) and Telecommunication Services Index (TTSI) also apply. Fig. 11b illustrates the mean square error (MSE) measure. It shows the batch size of the Tadawul All Share Index(TASI) is increased, and we find that forecast measures enhance until it reaches its best results at a batch size of 8 at a value of 0.0001, while it gets fluctuated for the other indices. Taking the MSE for Banks (TBNI) Index as an example, the optimal value for the TBNI index at batch size 2 is 0.0013, and it increases to 0.1114 at batch size 32. Thereafter, the index decreases with batch size. Similarly, when batch size over batch size exceeds eight, the Materials Index (TMTI) and Telecommunication Services Index (TTSI) also apply.

Fig. 11c, however, presents the root means square error (RMSE) of each batch size determined by the indices. For each batch size, each experiment was repeated 500 times(number of Epochs). As indicated in the figure, the RMSE is substantially higher for the Banks Index (TBNI), the Materials Index



(a) MAE

(b) MSE

(c) RMSE

(d) MAPE

Fig. 11. Model Performance Versus Varying Batch Sizes.

(TMTI), and the Telecommunication Services Index (TTSI) mainly because of the number of trading days for these indices compared to the Tadawul All Share Index(TASI). Results show an evident dependence on the number of trading days. The RMSE ranges from 0.1697 to .5409 obtained for the Banks Index (TBNI), from 0.1885 to 0.7638 for Telecommunication Services (TTSI), from 0.2361 to 0.5323 for Materials Index (TMTI). Fig. 11d shows the Mean Absolute Percentage Error (MAPE) for each batch size for the four indices. Despite outperforming practically all other indices with a MAPE value of 1.681 for batch size 8, there is another batch size where TASI does just as well on this accuracy measure. Considering the effects of sampling (trading days) on results, it makes sense that the result would differ.

B. Model Validation

Fig. 12 to 15 depict the predicted versus actual closing price for the four datasets: Tadawul All Share Index (TASI), Banks (TBNI), Telecommunication Services (TTSI), and Materials Index (TMTI) indices. These graphs show the best results based on a comparison between the actual and forecasted stock prices (close prices). On each chart, orange and blue lines depict the actual values and predicted values, respectively. The plots provide the timelines of the whole dataset. Fig. 12 plots the closing prices for the TASI dataset for the period from early 1990s to 2021. Note in the figure that there is a relatively bigger difference between the actual and predicted values of the stock closing prices in the earlier period of the data. However, the differences get smaller for the later time periods. Overall, all the four figures show a reasonably small differences between the actual and predicted values, indicating a good model performance. The results of our study indicate that the proposed model is very effective in analyzing and capturing trends, as well as forecasting them accurately.

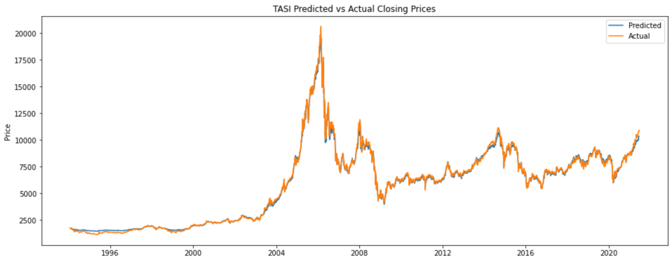


Fig. 12. TASI-Predicted vs Actual Closing Price for the whole dataset.

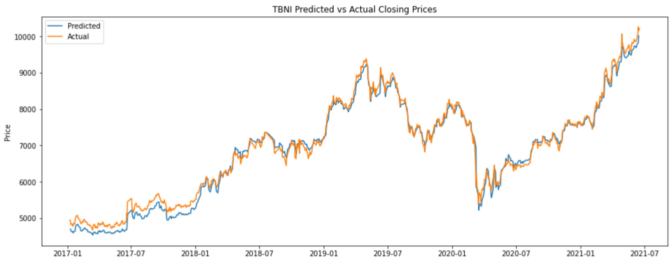


Fig. 13. TBNI-Predicted vs Actual Closing Price.

C. Predicting Future Stock Closing Prices

The next day's closing price of the selected stock is derived from the model prediction. Fig. 16 depicts the predicted and

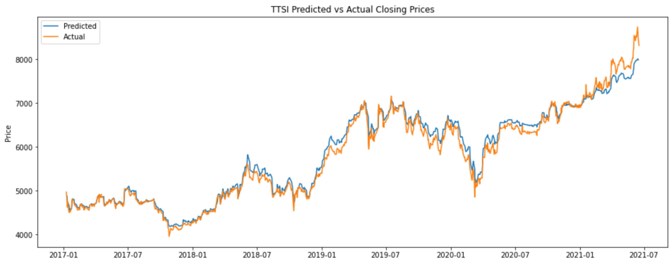


Fig. 14. TTSI-Predicted vs Actual Closing Price.

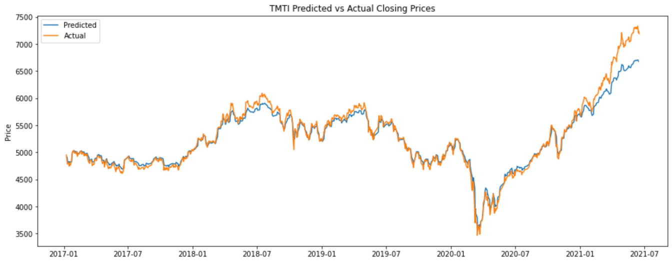


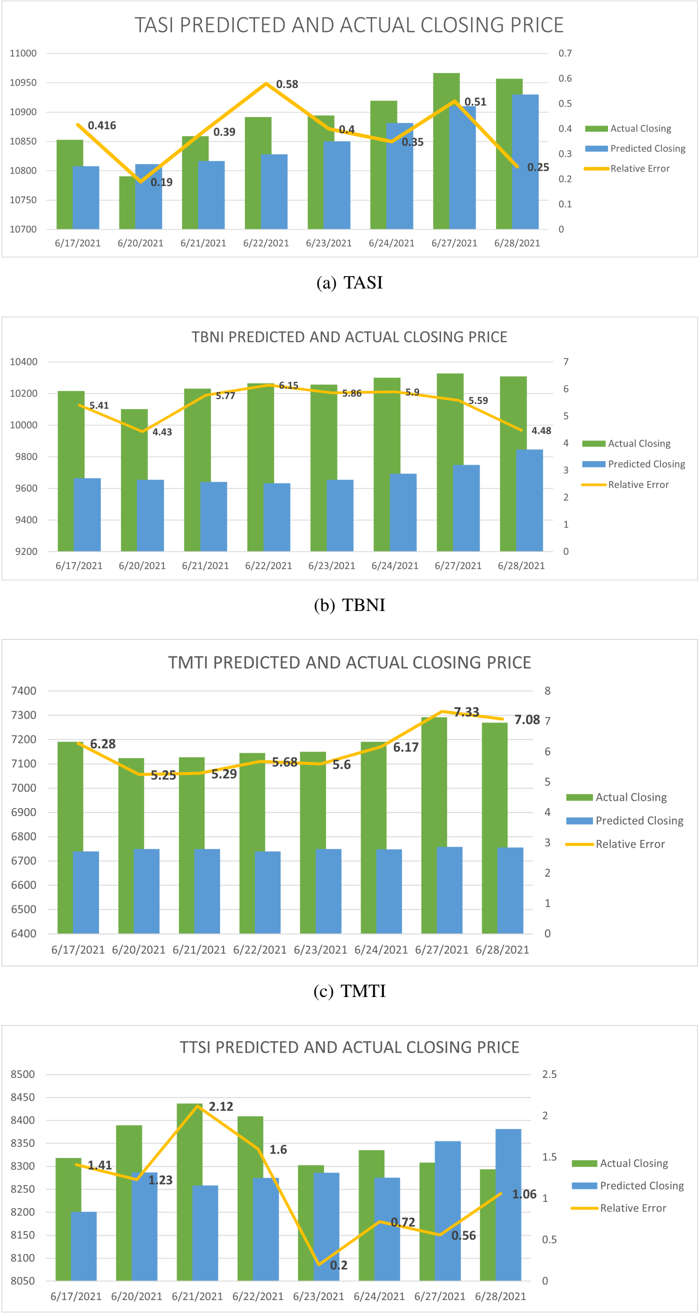
Fig. 15. TMTI-Predicted vs Actual Closing Price.

actual closing for eight trading days starting from 6/17/2021 till 6/28/2021 by using the model for the four indices TASI, TBNI, TTSI, and TMTI. The figure illustrates that the range of relative error fluctuation within the eight working days is between 0.19 and 0.58 for Tadawul All Share Index (TASI) and between 4.43 and 6.15 for the Banks index. As a result, the model accurately predicted the closing price of TASI and Bank with more than 99 and 94 percent, respectively. According to the model, TASI's closing price, for example, on 2021/06/17, will be 10807.94, while it was actually 10853.12 at the time. 45.18 points is a relatively small difference. In contrast, the relative error of Telecommunication Services (TTSI) fluctuated between 0.2, and 2.12, while Materials Index (TMTI) fluctuated between 5.25, and 7.33. Consequently, TMTI and TTSI closing prices were correctly predicted with more than 92, and 97 percent, respectively. The proposed model predicts the market closing price with a better than 90% accuracy, making it an exceptionally effective and practical model.

V. Conclusion and Future Work

We propose a transformer-based formalization model for stock price prediction. A significant influence on our architecture is a vision transformer (ViT) [36] using divided space. The vision transformer (ViT) is among the first attempts to apply the outstanding performance of Transformers. Using transformer network architectures with split time series into patches shows that hidden dynamics can be captured and predictions made reasonably. The model was trained using data from the Saudi Stock Exchange (Tadawul). As a result, we were able to predict the stock price of the TadawulAll Share Index (TASI), Telecommunication services Index (TTSI), Banks Index (TBNI), and Materials Index (TMTI) with accuracy that exceeds 90%.

We evaluated the proposed transformer model using four accuracy metrics, MAE, MSE, MAPE, and RMSE. We described the experimental results related to model optimisation and model validation for all the four datasets. Subsequently, we presented results for the prediction of future stock closing prices. We were able to achieve over 90% accuracy compared to the best 72% reported in the literature (see Table I). Furthermore, the experiments showed that the proposed model architectures that split time series into patches were able to identify the dynamics and complex patterns from irregularities in financial time series. Transformer architecture has also been shown to identify sudden changes in stock markets, as reflected in the results. However, the changes occurring may not always



(a) TASI

(b) TBNI

(c) TMTI

(d) TTSI

Fig. 16. Prediction of Unseen Future Stock Closing Price.

appear regularly or follow the same cycles each time.

Acknowledgment

The experiments reported in this paper were performed on the Aziz supercomputer at King Abdulaziz University.

References

[1] T. Yigitcanlar, L. Butler, E. Windle, K. C. Desouza, R. Mehmood, and J. M. Corchado, "Can Building "Artificially Intelligent Cities" Safeguard Humanity from Natural Disasters, Pandemics, and Other Catastrophes? An Urban Scholar's Perspective," *Sensors*, vol. 20, no. 10, p. 2988, may 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/10/2988

[2] T. Yigitcanlar, N. Kankanamge, M. Regona, A. Maldonado, B. Rowan, A. Ryu, K. C. Desouza, J. M. Corchado, R. Mehmood, and R. Y. M. Li, "Artificial Intelligence Technologies and Related Urban Planning and Development Concepts: How Are They Perceived and Utilized in Australia?" *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 6, no. 4, p. 187, dec 2020. [Online]. Available: https://www.mdpi.com/2199-8531/6/4/187

[3] E. Alomari, I. Katib, A. Albeshri, and R. Mehmood, "COVID-19: Detecting Government Pandemic Measures and Public Concerns from Twitter Arabic Data Using Distributed Machine Learning," *International Journal of Environmental Research and Public Health*, vol. 18, no. 1, p. 282, jan 2021. [Online]. Available: https://www.mdpi.com/1660-4601/18/1/282

[4] S. Alotaibi, R. Mehmood, I. Katib, O. Rana, and A. Albeshri, "Sehaa: A Big Data Analytics Tool for Healthcare Symptoms and Diseases Detection Using Twitter, Apache Spark, and Machine Learning," *Applied Sciences*, vol. 10, no. 4, p. 1398, feb 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/4/1398

[5] E. Alomari, I. Katib, A. Albeshri, T. Yigitcanlar, R. Mehmood, and A. A. Sa, "Iktishaf+: A Big Data Tool with Automatic Labeling for Road Traffic Social Sensing and Event Detection Using Distributed Machine Learning," *Sensors*, vol. 21, no. 9, p. 2993, apr 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/9/2993

[6] M. Aqib, R. Mehmood, A. Alzahrani, I. Katib, and A. Albeshri, "A Deep Learning Model to Predict Vehicles Occupancy on Freeways for Traffic Management," *IJCSNS - International Journal of Computer Science and Network Security*, vol. 18, no. 12, pp. 246–254, 2018.

[7] S. Usman, R. Mehmood, and I. Katib, "Big data and hpc convergence for smart infrastructures: A review and proposed architecture," in *Smart Infrastructure and Applications Foundations for Smarter Cities and Societies*. Springer Cham, 2020, pp. 561–586.

[8] R. Mehmood, F. Alam, N. N. Albogami, I. Katib, A. Albeshri, and S. M. Altowaijri, "UTiLearn: A Personalised Ubiquitous Teaching and Learning System for Smart Societies," *IEEE Access*, vol. 5, pp. 2615–2635, 2017.

[9] M. Aqib, R. Mehmood, A. Alzahrani, and I. Katib, *A smart disaster management system for future cities using deep learning, gpus, and in-memory computing*, 2020.

[10] A. Omar Alkhamisi and R. Mehmood, "An Ensemble Machine and Deep Learning Model for Risk Prediction in Aviation Systems," in *2020 6th Conference on Data Science and Machine Learning Applications (CDMA)*. Riyadh, Saudi Arabia: Institute of Electrical and Electronics Engineers (IEEE), mar 2020, pp. 54–59. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9044233

[11] H. Alotaibi, F. Alsolami, and R. Mehmood, "DNA Profiling: An Investigation of Six Machine Learning Algorithms for Estimating the Number of Contributors in DNA Mixtures," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 12, pp. 130–137, 2021.

[12] R. Mehmood, S. See, I. Katib, and I. Chlamtac, *Smart Infrastructure and Applications: foundations for smarter cities and societies*, R. Mehmood, S. See, I. Katib, and I. Chlamtac, Eds. Springer International Publishing, Springer Nature Switzerland AG, 2020.

[13] S. Alotaibi, R. Mehmood, and I. Katib, "Sentiment Analysis of Arabic Tweets in Smart Cities: A Review of Saudi Dialect," in *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 2019, pp. 330–335.

[14] Z. Hu, Y. Zhao, and M. Khushi, "A Survey of Forex and Stock Price Prediction Using Deep Learning," *Appl. Syst. Innov.*, vol. 4, no. 1, p. 9, feb 2021. [Online]. Available: https://www.mdpi.com/2571-5577/4/1/9

[15] J. Sirignano and R. Cont, "Universal features of price formation in financial markets: perspectives from deep learning," *Quant. Financ.*, vol. 19, no. 9, pp. 1449–1459, 2019. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/14697688.2019.1622295

[16] E. Guresen, G. Kayakutlu, and T. U. Daim, "Using artificial neural network models in stock market index prediction," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 10 389–10 397, 2011. [Online]. Available: https://www.researchgate.net/publication/220219343

[17] L. Takeuchi and Y. Lee, "Applying Deep Learning to Enhance Momentum Trading Strategies in Stocks," Tech. Rep. December 1989, 2013. [Online]. Available: http://cs229.stanford.edu/proj2013/TakeuchiLee-ApplyingDeepLearningToEnhanceMomentumTradingStrategiesInStocks.pdf

[18] M. Nikou, G. Mansourfar, and J. Bagherzadeh, "Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms," *Intell. Syst. Accounting, Financ. Manag.*, vol. 26, no. 4, pp. 164–174, 2019. [Online]. Available: https://www.researchgate.net/publication/337735594

[19] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *Int. J. Forecast.*, vol. 37, no. 1, pp. 388–427, 2021.

[20] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting," *Adv. Neural Inf. Process. Syst.*, vol. 32, jun 2019. [Online]. Available: http://arxiv.org/abs/1907.00235

[21] U. Khandelwal, H. He, P. Qi, and D. Jurafsky, "Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context," in *Proc. 56th Annu. Meet. Assoc. Comput. Linguist. (Volume 1 Long Pap.*, vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, may 2018, pp. 284–294. [Online]. Available: http://arxiv.org/abs/1805.04623 http://aclweb.org/anthology/P18-1027

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem. Neural information processing systems foundation, jun 2017, pp. 5999–6009. [Online]. Available: https://arxiv.org/abs/1706.03762v5

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," oct 2020. [Online]. Available: http://arxiv.org/abs/2010.11929

[24] R. J. Hyndman, "A brief history of forecasting competitions," Tech. Rep. 1, 2020. [Online]. Available: http://monash.edu/business/ebs/research/publications

[25] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," *2017 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2017*, vol. 2017-Janua, pp. 1643–1647, 2017.

[26] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, nov 1997. [Online]. Available: http://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf

[27] N. Naik and B. R. Mohan, "Study of stock return predictions using recurrent neural networks with LSTM," in *Commun. Comput. Inf. Sci.*, vol. 1000. Springer Verlag, may 2019, pp. 453–459. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-20257-6_39

[28] T. Skehin, M. Crane, and M. Bezbradica, "Day ahead forecasting of FAANG stocks using ARIMA, LSTM networks and wavelets," in *CEUR Workshop Proc.*, vol. 2259, 2018, pp. 186–197.

[29] D. M. Nelson, A. C. Pereira, and R. A. De Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *Proc. Int. Jt. Conf. Neural Networks*, vol. 2017-May, 2017, pp. 1419–1426. [Online]. Available: https://www.researchgate.net/publication/318329563

[30] S. Y. Shih, F. K. Sun, and H. yi Lee, “Temporal pattern attention for multivariate time series forecasting,” *Mach. Learn.*, vol. 108, no. 8-9, pp. 1421–1441, sep 2019. [Online]. Available: https://doi.org/10.1007/s10994-019-05815-0

[31] G. Lai, W. C. Chang, Y. Yang, and H. Liu, “Modeling long- and short-term temporal patterns with deep neural networks,” Tech. Rep., 2018. [Online]. Available: https://doi.org/10.475/123_4

[32] M. U. Gudelek, S. A. Boluk, and A. M. Ozbayoglu, “A deep learning based stock trading model with 2-D CNN trend detection,” in *2017 IEEE Symp. Ser. Comput. Intell.* IEEE, nov 2017, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/document/8285188/

[33] L. Di Persio and O. Honchar, “Artificial neural networks architectures for stock price prediction: Comparisons and applications,” Tech. Rep., 2016.

[34] D. Povey, H. Hadian, P. Ghahremani, K. Li, and S. Khudanpur, “A time-restricted self-attention layer for ASR,” in *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2018-April. Institute of Electrical and Electronics Engineers Inc., sep 2018, pp. 5874–5878.

[35] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, “Image transformer,” Tech. Rep., jul 2018. [Online]. Available: http://proceedings.mlr.press/v80/parmar18a.html

[36] G. Bertasius, H. Wang, and L. Torresani, “Is Space-Time Attention All You Need for Video Understanding?” feb 2021. [Online]. Available: http://arxiv.org/abs/2102.05095

[37] J.-S. Chou, D.-N. Truong, and T.-L. Le, “Interval Forecasting of Financial Time Series by Accelerated Particle Swarm-Optimized Multi-Output Machine Learning System,” *IEEE Access*, vol. 8, pp. 14 798–14 808, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8955860/

[38] V. Braverman, R. Ostrovsky, and C. Zaniolo, “Optimal sampling from sliding windows,” *J. Comput. Syst. Sci.*, vol. 78, no. 1, pp. 260–272, jan 2012. [Online]. Available: http://dx.doi.org/10.1016/j.jcss.2011.04.004 https://linkinghub.elsevier.com/retrieve/pii/S0022000011000493

[39] K. Bandara, C. Bergmeir, and S. Smyl, “Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach,” *Expert Syst. Appl.*, vol. 140, p. 112896, feb 2020.

[40] S. Smyl and K. Kuber, “Data Preprocessing and Augmentation for Multiple Short Time Series Forecasting with Recurrent Neural Networks,” Tech. Rep., 2016. [Online]. Available: https://www.researchgate.net/publication/309385800

[41] K. Chen, Y. Zhou, and F. Dai, “A LSTM-based method for stock returns prediction: A case study of China stock market,” in *2015 IEEE Int. Conf. Big Data (Big Data).* IEEE, oct 2015, pp. 2823–2824. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7364089/ http://ieeexplore.ieee.org/document/7364089/

[42] M. Nabipour, P. Nayyeri, H. Jabani, S. S., and A. Mosavi, “Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis,” *IEEE Access*, vol. 8, pp. 150 199–150 212, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9165760/

[43] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” mar 2016. [Online]. Available: http://arxiv.org/abs/1603.04467

[44] J. Huang, J. Chai, and S. Cho, “Deep learning in finance and banking: A literature review and classification,” *Front. Bus. Res. China*, vol. 14, no. 1, p. 13, dec 2020. [Online]. Available: https://fbr.springeropen.com/articles/10.1186/s11782-020-00082-6