Stock Market Prediction of the Saudi Telecommunication Sector Using Univariate Deep Learning Models

Hadi S. AlQahtani[1], Prof. Mohammed J. Alhaddad[2], Dr. Mutasem Jarrah[3]
Department of Information Technology-Faculty of Computing and Information Technology,
King Abdulaziz University, Jeddah, Saudi Arabia[1, 2]
Department of Artificial Intelligence and Data Science, Applied Science University, Jordan[3]

***Abstract*—Stock market volatility, randomness, and complexity make accurate stock price prediction very elusive, though it is required for logical investment and risk management. This study compares four Deep Learning (DL) models, Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and a CNN-LSTM model, to predict the Saudi Telecommunication sector by focusing on the closing price time series. The daily historical closing prices of STC, Mobily, and Zain companies are gathered and preprocessed, involving duplicate removal, feature selection, and Min-Max scaling. Models were trained with MSE loss, whereas validation was done with the RMSE and MAE. The study points toward the ability of deep learning to capture complex nonlinear regression patterns in the ebbs and flows of volatile financial markets. A comparative analysis reveals that the LSTM model yielded the lowest Test RMSE in all cases (Mobily: 1.169705, STC: 0.708495, Zain: 0.27147), therefore, presenting the best overall predictive accuracy. On the other hand, RNN almost always had the highest Test RMSE values (Mobily: 1.688603, STC: 1.143664, Zain: 0.666184), highlighting its limitations. The CNN and CNN-LSTM models showed intermediate performance, with implications for enhanced financial forecasting and decision-making within this specific market segment.**

Keywords*—Deep learning; stock market; prediction; models; regression; time series

I. INTRODUCTION

The stock market is complex and difficult to forecast, but it is extremely valuable to financial firms, hedge funds, traders, and market speculators [1]. The share prices are influenced by numerous variables, such as economic factors, worldwide events, company reports, and market sentiment, and are therefore random and extremely difficult to predict [2]. Having the ability to forecast changes in stock prices even slightly and accurately is extremely valuable in designing investment strategies, risk management, and overall financial decision-making [4]. Historically, stock market prediction has evolved over time through various methods. Initially, many employed traditional statistical techniques such as Autoregressive Integrated Moving Average (ARIMA) models [20]. These are simple and suitable when working with stable data, but struggle when dealing with complex patterns and long-term trends often found in finance data [4]. The advancement of machine learning, particularly approaches through deep learning, has significantly transformed financial prediction. Deep learning models are distinctive in that they are able to learn from past data without depending on pre-set mathematical formulas. This is helpful in finding underlying relationships in data that linear models may not detect [9]. That is why they are particularly suitable for the fast and dynamic character of stock market data [13]. The objective of this study is to answer the question: Which deep learning architecture—RNN, LSTM, CNN, or a hybrid CNN-LSTM—provides the highest predictive accuracy and robustness in a univariate stock market forecasting? It closely examines and contrasts the predictive capabilities of individual deep learning models on the Saudi Telecommunication market. This study hypothesizes that of the selected models of deep learning (LSTM, RNN, CNN, and CNN-LSTM), the Long Short-Term Memory (LSTM) network forecasts closing stock prices of the Saudi Telecommunication industry more accurately. The reason is that LSTM is designed to learn long-term patterns in sequence data [21]. A lot of financial literature on deep learning in financial markets exists, but requires powerful and precise models on a specific stock market that forecast daily closing stock prices based on a single variable in various market scenarios. This research investigates this requirement by studying the Saudi Telecommunication sector, which is significant but not as frequently researched as major international stock indices. This study will provide useful information regarding how the new models perform in an evolving market.

This research compares various approaches in the field of stock market prediction, which has not been researched much, and it provides additional evidence. How the research is described illustrates how it was conducted, making it simple for other researchers to conduct the research again and comprehend it. It also acts as a useful guideline for additional research in this area. The research indicates that, despite being complex, deep learning models are more capable of examining stock market trends and providing a more accurate means of predicting them [24].

The remainder of this study is structured as follows: Section II examines in detail significant research regarding deep learning models applied to time series prediction. Section III describes the methods, such as how data was gathered and prepared, and how the model was constructed and trained.

Section IV presents the results of the model tests across each company. Section V discusses what these results imply, including model comparison and the study's limitations. Finally, Section VI concludes the study by outlining the major findings and providing recommendations for future research.

II. Literature Review

Deep learning models are powerful predictors of time series. They are capable of learning intricate patterns and long-term correlations without the necessity of specific feature engineering [6]. Deep models are particularly effective in detecting trends in the stock market, which usually exhibits chaotic and non-linear tendencies [13].

A. Recurrent Neural Networks (RNNs)

Recurrent neural networks tend to be appropriate when the data comes in sequence and when the position of components within it matters. RNNs have loop markings that feed-forward networks do not have. At a particular instant in time, when the output of a neuron is generated, this output is fed into the RNN as an input for the subsequent time. The name loop comes from the fact that the state in the network, called the hidden state, is retained in RNNs. They can learn patterns and relationships in sequences [1]. Hence, they can perform well over time series forecasting, as the input data is sequential, and the past step information can be recalled. The main problems of RNNs are the so-called vanishing gradient problem. When being trained, the network learns backwards; however, gradients tend to become very small with a large number of time steps, making the network difficult to learn from past data [2]. These make RNNs forget salient details from earlier steps and lack the capability to recall long sequences and times. RNNs could also suffer from overfitting, especially when considering noisy and unpredictable data such as stock prices.

The inherent recurrent structure of RNNs allows the design of a model of sequential correlations unavailable to feedforward networks [4]. Nonetheless, it is difficult for vanilla RNNs to handle longer sequences because these models have issues with exploding/vanishing gradients [14]. RNNs with memory solve this problem using gated memory cells that help with long-term dependency. The recurrent connections allow RNNs to capture temporal dependencies within the series of observations [16]. In [17], the authors discussed deep learning stock forecasting models and noted that RNNs with memory can identify temporal dependencies that are not extractable by other models.

B. Long Short-Term Memory (LSTM) Networks

LSTM came into being as an enhancement to the Recurrent Neural Network to address the vanishing gradient problem. According to research [3], LSTMs help to learn long-range patterns in sequential data. LSTMs achieve the feat because of their special inner architecture consisting of "memory cells" and three distinct "gates": the input gate, the forget gate, and the output gate [3]. These intelligent gates decide how information flows in and out through the cell state and assist the network in knowing what to remember and what to forget over time. The majority of authors who write about it assert that LSTM outperforms the basic RNN and other simplistic models in stock price forecasting. Particularly, LSTM models are mostly regarded by researchers as somewhat more accurate in stock price forecasting. Experimenting further, it has been established that the LSTM tends to make fewer errors than RNNs, especially when it comes to short-term forecasting [8].

Researchers of [10] proved that LSTM networks perform better on financial time series data characterized by nonlinearity and nonstationarity. Furthermore, they refined the algorithm to improve its predictive accuracy. In [12], the authors proposed a strong LSTM model coupled with sentiment analysis to improve stock price prediction accuracy, proving the robustness of the model in the noisy financial market. In [24], the authors confirmed that LSTM consistently outperforms RNN and CNN models in long-term prediction, thereby cementing its suitability for financial time series.

C. Convolutional Neural Networks (CNNs)

In [3], the authors noted that CNNs can be very effective at modeling short-term stock price changes, although CNNs have more trouble with long-term dependencies compared to LSTM models. A time series CNN typically contains convolutional layers. Pooling layers, such as max pooling, which reduce features to a smaller size, are followed by convolutional layers. This retains relevant features but with reduced data. Additionally, the output of these layers is converted into a 1D vector and then passed through one or more fully connected layers. These layers perform the bulk of the predicting, such as future stock prices [5].

In [7], the authors proved that CNN-based models are competitive, especially when combined with other architectures. CNNs are renowned for dealing with images and spatial information. CNNs have been used for stock market prediction, leveraging their ability to model local trends and patterns in sequential data. CNNs have also been transformed smartly to deal with 1D time series information [7]. The transformation employs convolutions on a time axis to enable CNNs to detect local patterns, features, and trends in sequential data [24].

D. Hybrid CNN and LSTM Models

The primary motivation for combining the use of both CNNs and LSTMs, is to capitalize on what each system excels at. This is a combined approach, typically beginning with a CNN scanning the raw time series data for significant, general patterns [3]. The product of these CNN layers, typically following pooling and converting it into a sequence, is then passed as input to the LSTM layers.

In [6], the authors also pointed out that hybrid models would have higher prediction accuracy at the cost of increased computational complexity. This collaboration is intended to provide a stronger insight into the time series data, including close as well as long-term connections, which typically results in more accurate prediction outcomes, as compared to the use of each model in isolation.

In [8], the authors clarified that a CNN-LSTM model performs better than single CNN and LSTM models on stock prediction tasks. Researchers of [24] proposed a hybrid CNN-LSTM model for stock forecasting and improved performance over single models by capturing both spatial patterns and temporal dependencies.

E. Comparing Various Methods of Predicting Stocks

Research papers typically compare various deep learning algorithms used in predicting stock prices, such as LSTM, RNN, and CNN ([1]-[5], [24]). The majority of these papers indicate that LSTM is typically the highest performing because it is capable of handling long-range dependencies in sequential data [5]. Other papers discussed that CNNs and RNNs are also not bad, but hybrid models such as CNN+LSTM—trained with single-price data—make minimal improvements [7]. Increasingly, more researchers are employing hybrid models to ensure predictions are accurate and dependable. According to [8], hybrid models attempt to address the issues of standalone models through the union of multiple types, frequently doing so better by blending the spatial knowledge of CNNs with the temporal considerations of RNNs or LSTMs.

III. Methodology

This study utilizes a deep learning structure, and it focuses on applying single-variable models in stock market prediction. This study entails the application and comparison of four different deep learning structures: Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and a hybrid of CNN and LSTM on the Saudi Telecommunication sector stock market.

A. Data Collection

Historical closing prices were obtained daily from the Yahoo Finance API. The source is reliable and easily accessible for financial data [15]. The companies chosen from the Saudi Telecommunication sector and their respective Yahoo Finance tickers are STC (7010.SR), Mobily (7020.SR), and Zain (7030.SR). Specifying the companies and tickers makes the dataset readable and easily reproducible. Under the single feature scheme of this study, the "Close" price per share was utilized as the only input.

B. Preprocessing Data

Accurate data preparation is paramount to the performance and strength of deep learning algorithms. The following cleaning processes were done:

1) Duplicate records were meticulously removed from the collected dataset to ensure data consistency, accuracy, and integrity.

2) As the study takes one thing at a time into consideration, only the "Close" price for each stock was kept for the model. All other potential features, such as Open, High, Low, and Volume, were excluded.

3) Min-Max Scaling is employed for normalizing the input data. It scales data linearly to a bounded range [0, 1]. Scaling enhances model convergence and numerical stability in gradient-based deep learning. The mathematical Formula (1) for Min-Max Scaling is expressed as:

$$x\_scaled = \left(\frac{x - min(x)}{(max(x) - min(x)}\right) \quad (1)$$

4) Preprocessed data was divided into an 80/20 train and test split to estimate generalization and avoid overfitting.

5) Windowing/Sequence Preparation was done. We used sliding windows to create input-output pairs of data, a common practice in time series deep learning.

C. Model Building

Each company had four different deep learning models to predict stock prices using one variable. Choosing proper hyperparameters, that is, the number of layers, units (neurons) per layer, dropout rates, learning rates, batch sizes, and epochs, is very important to improve the performance of the deep learning model. The precise hyperparameter values are not given in the data, but the parameters are generally adjusted by careful methods so that the performance is optimal, facilitates convergence, and reduces problems like overfitting. Common optimizers like Adam are generally used to train such models.

D. Long Short-Term Memory (LSTM)

LSTM architecture is crafted to identify long-term trends in sequential data. It solves the common issue of vanishing gradients in standard RNNs [11]. Its building blocks therein—input gate, forget gate, output gate, and cell state—make this possible. These clever gates control the flow of information, making it possible for the network to determine what to retain or ignore through long sequences. The mathematical Formula (2) to Formula (5), governing the operations within an LSTM cell, are as follows:

$$\text{Input Gate} = i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2)$$

$$\text{Forget Gate} = f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (3)$$

$$\text{Output Gate} = o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (4)$$

$$\text{Cell State Update} = c_t = f_t \times c_{t-1} + i_t \times tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (5)$$

$$\text{Hidden State Update} = h_t = o_t \times tanh(c_t)$$

where,

xt is the current input vector

ht−1 is the previous hidden state vector

ct−1 is the previous cell state vector.

W is the Weight matrix for the input

U is the Recurrent weight matrix for the hidden state

B is the bias vector

σ is the Sigmoid activation function

tanh: Hyperbolic tangent activation function.

This detailed mathematical formulation demonstrates the internal workings of the LSTM and its fundamental advantage over simpler RNNs in handling long-term dependencies, which is crucial for capturing the complex patterns in stock price movements.

E. Recurrent Neural Network (RNN)

The Recurrent Neural Network (RNN) is the basic model used to predict time series in this study. Its fundamental building block is with recurrent connections, where a neuron's output at a given time is reused as input to the network at the next time [18]. This feedback mechanism allows the network to have a "hidden state", a memory that processes information in a sequence. The Formula (6) and Formula (7) for an RNN cell, describing how it works, are:

$$\text{Hidden state} = h_t = \sigma(W_x x_t + W_h h_{t-1} + b_h) \quad (6)$$

$$\text{Output} = y_t = \sigma(W_Y h_t + b_Y) \quad (7)$$

where,

xt is the current input vector

ht−1 is the previous hidden state vector

ht is the current hidden state vector

yt is the current output vector

Wx is the weight matrix for the input

Wh is the recurrent weight matrix for the hidden state

Wy is the weight matrix for the hidden state to output

bh, by are the bias vectors

σ is the activation function (typically sigmoid or tanh).

Including the RNN formula gives the explicit mathematical difference from LSTM. This explains why LSTMs were created and why LSTMs usually work better with long sequences. It shows how deep learning models have progressed in time series analysis.

F. Convolutional Neural Network (CNN)

The CNN architecture is adapted to handle time series data for the purpose of capturing local patterns and features in sequences [5]. Adaptation is mainly done through one-dimensional (1D) convolutions, which are apt for handling time data by identifying "spatial" patterns along the time axis [19]. The architecture generally comprises:

1) Convolutional layers: These layers have filters (known as kernels) that move over the 1D input data. Every filter applies a convolution operation to extract local patterns or features from subsets of the time series.

2) Pooling layers: Convolutional layers are usually followed by pooling layers, like Max Pooling, to decrease the feature maps' size. This lowers their size without losing the important features, making the model work better with small changes in input and lessening the computation involved [23].

3) Flattening and fully connected (dense) layers: The feature map output of the convolutional and pooling layers is flattened into a 1D vector. This vector is fed into one or more fully connected (dense) layers that learn useful features and carry out the final regression task [23].

4) Output layer: For predicting one type of time series, the output layer generally has one neuron to predict the future value of the stock price. The main operation of a 1D CNN is defined by the discrete convolution Formula (8):

$$(f * g)[n] = \int_{m=-\infty}^{+\infty} f[m] \cdot g[n - m] \quad (8)$$

where,

f is the input signal (the time series data)

g is the convolutional filter (kernel)

n is the position of the output element

The formula is the general definition of discrete-time convolution [22]. It continues by sliding the filter over the signal, multiplying the filter by the corresponding part of the signal at each element, and adding the outcomes to create one output point. This move reflects how methods of identifying patterns in space, normally used in images, can be used in time-based data. This shows how flexible deep learning systems are [2].

G. Combination of CNN and LSTM Model

This model is a hybrid architecture that combines synergistically the strengths of Convolutional Neural Networks (CNNs) for learning features and Long Short-Term Memory (LSTM) networks for learning sequences [3]. The architecture is made up of an early CNN component that takes in the raw time series data. This CNN block extracts important characteristics and neighborhood trends, for example, detecting some of the price changes or trends in short-time periods, from the given sequence [7]. Whatever emerges from these CNN layers, after pooling and transforming into a sequential representation, is then offered as input to the LSTM blocks. The LSTM layers then learn significant connections and timing relationships from these prepared, detailed sequences [8]. The primary reason to use this combined approach is to take advantage of the strengths of both model types: CNNs are strong at extracting powerful local patterns and reducing the size of the input [12], while LSTMs are extremely capable at interpreting time-based sequences and remembering long-term connections [12]. This is a combination to provide greater insight into the time series data, potentially providing improved predictions compared to the use of singular models [24]. The hybrid model is an ingenious method, combining the positives of each method in time series analysis, indicating the sophistication of the study.

H. Model Training

We primarily utilized Mean Squared Error (MSE) for training all the models as the loss function. We can calculate MSE with the help of Formula (9):

$$MSE = \frac{\sum_{i=1}^{n}(yi - pi)^2}{n} \quad (9)$$

where,

yi is the observed value

pi is the corresponding predicted value

n is the number of values

MSE calculates the average of the squared errors or differences, representing the average squared difference between actual and predicted values. It is typically utilized for regression problems and provides more weight to large errors compared to small errors. To verify how well the models perform on the training as well as test datasets, we utilized Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) as crucial measures.

The RMSE Formula (10) is:

$$RMSE = \left(\frac{(\Sigma(P_i - O_i)^2)}{n}\right) \quad (10)$$

where,

Pi is the value predicted

Oi is the observed value

n represents the data points.

RMSE indicates the amount of error, expressed in the same units as the target variable. All input features were normalized to the range [0,1] prior to training. The RMSE values were computed directly in the normalized space; therefore, they are unitless and represent error relative to the scaled data range. The MAE Formula (11) is:

$$MAE = \left(\frac{(\Sigma|y_i - p_i|)}{n}\right) \quad (11)$$

MAE takes the average of the difference between actual and predicted values. MAE is less sensitive to outlier values compared to RMSE and directly indicates the average magnitude of the error. In RMSE and MAE, lower values indicate good prediction and better model performance. For regression problems in time series forecasting, it is typical to use MSE as the loss function and RMSE/MAE as the measure in assessing performance. This ensures we evaluate model performance to acceptable standards in academia.

IV. RESULTS

In this section, we display the results from the model tests of all three Saudi Telecommunication companies. The performance metrics, such as Train RMSE, Train MAE, Test RMSE, and Test MAE, for every deep learning model are presented in clear, readable tables, together with a summary, indicating key facts for each company.

Table I and Fig. 1 illustrate the performance of each of the four deep learning models on Mobily's stock price data. The findings indicate how accurate each model was in forecasting Mobily's prices, in that we can now compare the training and testing errors of the models. This is necessary for knowing how well the models can perform on new data and selecting the best model for this particular stock. For Mobily, the LSTM model recorded the smallest Test RMSE (1.169705) and Test MAE (0.913336), indicating that it was the most capable of forecasting new data out of the models. The RNN model registered the highest error rates in all metrics, indicating that it was not able to follow the exact changes in Mobily's share prices. The CNN and CNN+LSTM models performed well, with the hybrid model performing slightly better than the standalone CNN.

Table II and Fig. 2 present the performance measures of STC's stock price data for the deep learning models. For STC, LSTM performed the best with the lowest Test RMSE (0.708495) and Test MAE (0.518296), indicating that it can predict well. CNN performed the worst with the highest Test RMSE (1.784101) and MAE (1.270669), indicating that it was the most challenged by the STC data among the models. The hybrid CNN+LSTM performed better than the standalone CNN and RNN, but not as well as LSTM on Test RMSE and Test MAE.

TABLE I. MODEL'S PERFORMANCE FOR MOBILY (7020.SR)

Model Name	Train RMSE	Train MAE	Test RMSE	Test MAE
LSTM	1.203972	0.785214	1.169705	0.913336
RNN	2.168396	1.809754	1.688603	1.334345
CNN	1.571197	1.00751	1.41664	1.115295
CNN + LSTM	1.388986	0.875195	1.388765	1.114691

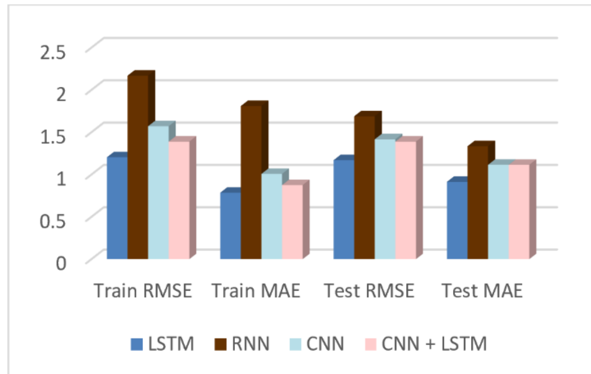

Fig. 1. Model's performance for Mobily (7020.SR).

TABLE II. MODEL'S PERFORMANCE FOR STC (7010.SR)

Model Name	Train RMSE	Train MAE	Test RMSE	Test MAE
LSTM	0.66395	0.367212	0.708495	0.518296
RNN	0.816863	0.494815	1.143664	0.906338
CNN	0.923335	0.533309	1.784101	1.270669
CNN + LSTM	0.780187	0.533703	1.027295	0.7784

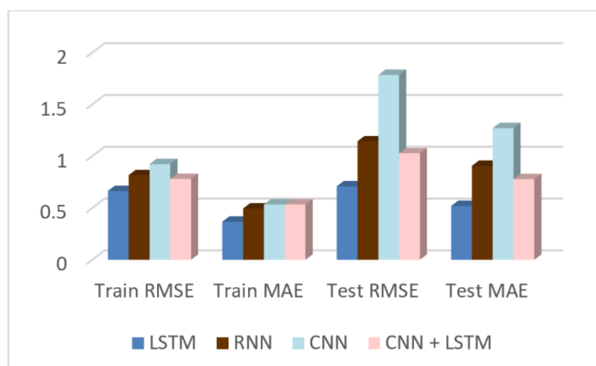

Fig. 2. Model's performance for STC (7010.SR).

TABLE III. MODEL'S PERFORMANCE FOR ZAIN (7030.SR)

	Train RMSE	Train MAE	Test RMSE	Test MAE
LSTM	0.446717	0.265711	0.27147	0.215682
RNN	0.770948	0.657144	0.666184	0.625866
CNN	0.518339	0.331256	0.302271	0.22412
CNN + LSTM	0.45294	0.281391	0.291989	0.223768

Table III and Fig. 3 present the last results for the third company. It provides the information required to make a complete comparison. It tells how the deep learning models performed with Zain's stock price data. For Zain, LSTM performed best with the smallest Test RMSE (0.27147) and Test MAE (0.215682). The CNN model also performed extremely well for Zain, placing next to the LSTM in both test metrics. The RNN model performed the worst, similar to Mobily and STC's. The CNN+LSTM hybrid model performed equally well as the pure CNN, and both are good options to forecast Zain's stock.

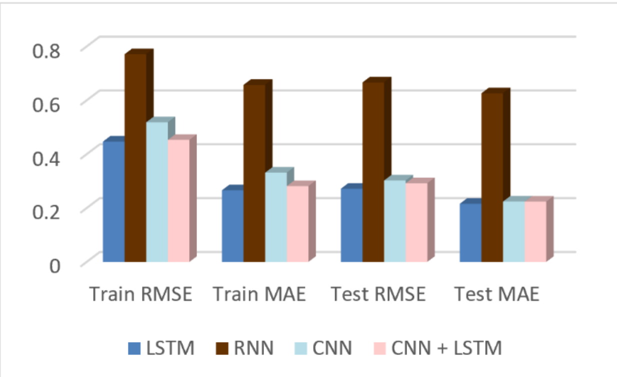

Fig. 3. Model's performance for ZAIN (7030.SR).

V. Discussion

The data in Tables I, II, and III indicate the performance of the deep learning models for three Saudi Telecommunications companies. The trend was consistent: the LSTM model always produced the lowest Test RMSE and Test MAE for Mobily, STC, and Zain, and thus was the most successful at forecasting accurate values from new data. Its consistency demonstrates the ability of LSTM to learn long-term trends and bypass the vanishing gradient problem, as cited in studies [11], and in this case, it is highly beneficial in forecasting stock prices.

A. *Comparative Analysis of Models*

LSTM performed stronger than the RNN models since the basic RNN model had the largest errors (MAE and RMSE) across all three firms. This is expected and is noted in other research regarding issues with simple RNNs and their lack of ability to address the vanishing gradient problem. This signifies that the models are unable to learn and remember information from long sequences [14]. This large disparity is why complex models such as LSTM were developed to address the primary issues of handling sequence data. The CNN model performed differently for each of the companies. It generally outperformed the RNN but did not always match up with the LSTM in terms of accuracy. In the case of Mobily and STC, the Test RMSE of the CNN was poorer than LSTM. Yet for Zain, the CNN model performed very competently with its Test RMSE (0.302271) and Test MAE (0.22412) close to the best of the LSTM (0.27147 and 0.215682, respectively). This indicates that CNNs are able to capture local trends and abrupt changes in the stock prices as reported in previous research ([5], [11]). The good performance for Zain can imply that its stock price variations have cleaner local trends or short-term trends that are easy for CNNs to detect. The CNN+LSTM hybrid model attempted to leverage the strength of CNN in discovering features and LSTM's ability in sequence learning. This was supposed to yield improved prediction, as various works demonstrate regarding hybrid models ([24], [3]). In the case of Mobily, the hybrid model performed slightly better than the sole CNN in Test RMSE. In the case of STC and Zain, its accuracy was comparable to or superior to single LSTM or CNN. This indicates that even if the hybrid approach is powerful, its benefits over an excellently designed single LSTM model for forecasting individual stock prices could be minor. The increased complications and computational demand of hybrid models do not always result in significantly superior performance, particularly if the data does not require both spatial discovery of features and long-term modeling of time [17].

B. *Implications of Findings*

It is vital to investors, traders, and decision-makers who make financial decisions in the Saudi Telecommunication market. LSTM has been performing better than any other model and has proven to be able to predict closing prices accurately every day. Identifying the best model will assist in the creation of automated trading systems or efficient forecasting tools for the market [9]. Accurate forecasts with minimal errors will provide useful information about the change in stock prices. This will assist in making more prudent investment decisions, identifying more profitable times for selling and buying, and developing more efficient risk management plans. Employing deep learning algorithms will assist in the analysis and prediction of this volatile market sector [19].

C. *Limitations Addressed by this Study*

This study addresses the limitations of previous research by conducting a comprehensive comparative analysis of four deep learning methods—LSTM, RNN, CNN, and a hybrid CNN+LSTM—applied to the Saudi Telecommunication sector, which encompasses three major companies and three distinct datasets. Unlike previous studies that typically rely on a single model or limited datasets, our research evaluates and compares the performance of these models across multiple datasets to enhance predictive accuracy and establish a strong basis for comparison. In doing so, this study not only identifies the most accurate predictive model for the sector but also offers practical insights into the trade-offs between model complexity and predictive performance. Moreover, by focusing on the Saudi Telecommunication market, the research fills an important geographic and industry-specific gap in the literature, providing contributions of both academic significance and practical value. Table IV summarizes the limitations of previous studies that are addressed by this study.

TABLE IV. LIMITATIONS OF PREVIOUS STUDIES ADDRESSED BY THIS STUDY

Limitations in Previous Studies	How This Study Bridges the Gap
Focus mainly on a single model (often LSTM, CNN, or LSTM-CNN).	Provides a comprehensive comparison of four deep learning models, including hybrid methods.
Use of univariate or limited datasets.	Applies models to major datasets of three companies in the Saudi telecommunications sector, giving broader validation.
Hybrids often add complexity with minimal evaluation of their true advantage.	Evaluates hybrid performance against standalone models to determine if the added complexity is justified.
Research concentrated on the US, Chinese, or Indian markets, etc.	Extends to an underexplored market (Saudi Telecom sector), contributing localized insights.

D. Limitations of the Study

This study provided useful information, but there exist certain limitations that need to be acknowledged. The model only examined the "Close" price as the input variable. This is less complicated to investigate trends in one variable, but it reduces the model's capacity to observe the complete picture of the stock market. Stock prices are influenced by various factors, such as Open, High, Low, and Volume prices, as well as external economic indicators, news sentiments, and technical indicators. Employing the multiple-variable method with the inclusion of the additional factors could result in improved and precise predictions [6]. It is greatly significant to tune the hyperparameters to achieve quality results from deep learning models [10]. These conclusions benefit the Saudi Telecommunication market but need to be examined in other markets, other contexts, or other financial markets in other nations without further tests. The unique characteristics of the market, regulation, and investors' behavior elsewhere could influence the extent to which the model works [22].

VI. CONCLUSION

This study examined the performance of four deep learning models. These models are LSTM, RNN, CNN, and a combination of CNN and LSTM. The above models were employed to predict stock prices for the Saudi Telecommunication industry. The study developed and evaluated the models using daily closing prices of Mobily, STC, and Zain. This involved meticulous data preparation and common performance-measuring methods. The overall outcome indicates that the LSTM model performed better than the other two models for all the companies. That is, it is able to learn long, non-linear, and complex patterns in stock prices. The RNN model had the largest errors, indicating that it does not handle long sequences well. The CNN model performed with mixed results, but performed exceptionally well for Zain, indicating that it is capable of identifying local patterns. The hybrid CNN+LSTM model, although a good concept, provided minor improvements over the single models in this instance. The comparison and analysis benefit the field of computational finance and deep learning in finance.

From the concepts of this research, future research should therefore consider integrating additional features such as Open, High, Low, Volume, technical indicators, and external variables (e.g., economic news, market sentiment) to enrich the predictive capacity of the models. Experimenting with advanced hybrid architectures and ensemble methods, which could better exploit the strengths of different models and capture both long-term and short-term dependencies. Expanding the application to other industries, sectors, and international markets to test the robustness and adaptability of the models under diverse financial environments.

In conclusion, this research underscores the continued dominance of LSTM in stock price forecasting, while also opening the door for refined CNN-based and hybrid approaches. By situating these results in the context of the Saudi telecommunications sector, the study provides a foundation for both academic inquiry and practical application in financial forecasting.

REFERENCES

[1] Edirisooriya, A., Weliwatta, R., Amarasena, H., & Ganesh, A. (2024). Deep learning architectures for time series forecasting. ResearchGate. https://doi.org/10.13140/RG.2.2.19904.75524

[2] Roy, S. S., R. C., Rohan C.,Kun C. L., C. S., & B M.-i. (2021), Random forest, gradient boosted machines, and deep neural network for stock price forecasting: a comparative analysis on South Korean companies, International Journal of Ad Hoc and Ubiquitous Computing 33(1), 62–71. https://doi.org/10.1504/IJAHUC.2020.104715

[3] Lu, Li, W., J., Y., Li, Wang, J. & Sun, A., (2020). A CNN-LSTM-based model to forecast stock prices. Complexity, 1–10. https://doi.org/10.1155/2020/6622927

[4] Kumar, Haider, K., & M. T. U. (2021 Application of neural network to technical analysis of stock market prediction. New Generation Computing, 39(1), 231–272. https://doi.org/10.1007/s00354-020-00104-0

[5] Borovykh, A., Bohte, S., & Oosterlee, C. W. (2021). Conditional time series forecasting with convolutional neural networks. arXiv preprint arXiv:1703.04691. https://arxiv.org/abs/1703.04691

[6] Chen, W., Hussain, W., Cauteruccio, F., & Zhang, X. (2023). Deep learning for financial time series prediction: A state-of-the-art review of standalone and hybrid models. Computer Modeling in Engineering and Sciences, 139, 187-224. https://doi.org/10.32604/cmes.2023.031388

[7] Chen, Y. C., & Huang, W. C. (2021). Constructing a stock-price forecast CNN model with gold and crude oil indicators. Applied Soft Computing, 112, 107760.

[8] Agnihotri, K., Yadav, S., Dahiya, S., Subramanyam, M., Patil, H., & Kalra, G. (2024). Next-generation stock market prediction: Integrating CNN and LSTM models. 2024 International Conference on Data Science and Network Security (ICDSNS), 1-6. https://doi.org/10.1109/ICDSNS62112.2024.10691069

[9] Khan, D. R., Patankar, A. B., & Khan, A. (2024). An experimental comparison of classic statistical techniques on univariate time series forecasting. Procedia Computer Science, 235, 2730-2740. https://doi.org/10.1016/j.procs.2024.04.257

[10] Peivandizadeh, A., Hatami, S., Nakhjavani, A., Khoshsima, L., Qazani, M. R. C., Haleem, M., & Alizadehsani, R. (2024). Stock market prediction with transductive long short-term memory and social media sentiment analysis. IEEE Access, 12, 87110–87130. https://doi.org/10.1109/access.2024.3399548

[11] García-Torres, Ayala, J., M., J. L. V., F., Divina, F., & Noguera, Gómez-Vela, (2021). Technical analysis strategy optimization using a machine learning approach in stock market indices. Knowledge-Based Systems, 225, 107119. https://doi.org/10.1016/j.knosys.2021.107119

[12] Park, K., Jayathilake, Y., Gajamannage & D. I. (2023). Real-time forecasting of time series in financial markets using sequentially trained

dual-LSTMs. Expert Systems with Applications, 223, 119879. https://doi.org/10.1016/j.eswa.2023.119879

[13] Mehtab, S., & Sen, J. (2021). A robust predictive model for stock price prediction using deep learning and natural language processing. TechRxiv. https://doi.org/10.36227/techrxiv.15023361.v1

[14] Noguer, A. M. (2024). The mathematics of recurrent neural networks. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5001243

[15] Peng, Z. (2019). Stocks analysis and prediction using big data analytics. International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS) (pp. 309-312). IEEE.

[16] Aseeri, A. O. (2023). Effective RNN-based forecasting methodology design for improving short-term power load forecasts: Application to large-scale power-grid time series. Journal of Computational Science, 68, 101984. https://doi.org/10.1016/j.jocs.2023.101984

[17] Jiang, W. (2021). Applications of deep learning in stock market prediction: Recent progress. Expert Systems with Applications, 184, 115537. https://doi.org/10.1016/j.eswa.2021.115537

[18] Francis, M. P. Fozap (2025). Hybrid Machine Learning Models for Long-Term Stock Market Forecasting: Integrating Technical Indicators. Journal of Risk and Financial Management, 18(4), 201. https://doi.org/10.3390/jrfm18040201

[19] Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied Soft Computing, 90, 106181. https://doi.org/10.1016/j.asoc.2020.106181

[20] Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2020). A comparison of ARIMA and LSTM in forecasting time series. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 1394–1401. https://doi.org/10.1109/ICMLA.2018.00227

[21] Vuong, P., Phu, L., Nguyen, T. H. V., Duy, L., Bao, P., & Trinh, D. (2025). A comparative study of deep learning approaches for stock price prediction. Digital Finance, 1-29. https://doi.org/10.1007/s42521-025-00149-0

[22] Yu, P., & Yan, X. (2020). Stock price prediction based on deep neural networks. Neural Computing and Applications, 32(6), 1609-1628.

[23] Zafar, A., Aamir, M., Mohd Nawi, N., Arshad, A., Riaz, S., Alruban, A., ... & Almotairi, S. (2022). A comparison of pooling methods for convolutional neural networks. Applied Sciences, 12(17), 8643.

[24] Zhao, Q., Hao, Y., & Li, Xuechen. (2024). Stock price prediction based on a hybrid CNN-LSTM model. Applied and Computational Engineering. 104. 110-115. 10.54254/2755-2721/104/20241065.