https://albalqajournal.ammanu.edu.jo/
Vol.(27) No.(4) 2024
Received: 23/05/2024
Accepted: 30/09/2024
Published: 31/12/2024

مجلة البلقاء للبحوث والدراسات
Al-Balqa Journal for Research & Studies

Navigating the Stock Market: The Role of State-of-the-Art Deep Learning Techniques

Anika Kanwal[1*], Man F. Lau[1], Al-Shamayleh Ahmad Sami[4], Akhunzada Adnan[3], Siva Chandrasekaran[1], SebastianP.H. Ng[1] and Kwan Yong Sim[2]

[1*]Department of Computing Technologies, School of Science, Computing and Engineering Technologies, Swinburne University of Technology, John Street Hawthorn, Melbourne, 3122, VIC, Australia.
[2]School of Engineering, Swinburne University of Technology, Sarawak, 93350, Malaysia.
[3]University of Doha for Science and Technology, Doha, Qatar.
[4]Department of Data Science and Artificial Intelligence, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, 19328, Jordan.

Abstract

Objectives: Certainty in investment decisions are the main tools for evaluation of stock markets. Every investor will have to identify the future stock prices before developing investment strategy so that their returns will be boosted. Due to the nonlinear, complicated, volatile, and dynamic character of stock data, forecasting stock prices is challenging. This study aims to address these challenges and improve prediction accuracy through advanced modelling techniques.

Methodology: We proposed a hybrid neural network that combines Bidirectional Gated Recurrent Units (BiGRU) with one-dimensional Convolutional Neural Networks (CNNs). Experiments were conducted on five stock price datasets, including three individual stock items and two performance indices of stock markets. The evaluation metrics included Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

Findings: The proposed hybrid model achieved the lowest MAE for individual datasets, ranging from 1.747 to 0.357. Additionally, our experiments demonstrated a 42–91% reduction in RMSE compared to standard CNN and GRU-based approaches, reflecting significantly improved prediction precision.

Implications: The results indicate that the hybrid Bidirectional GRU and CNN model is effective for precise stock price prediction, offering enhanced reliability for investment decision-making. This model can support investors and analysts in navigating the complexities of the stock market.

Conclusions: The hybrid Bidirectional GRU and CNN model demonstrates superior performance in stock price prediction compared to traditional methods. It holds promise for advanc- ing predictive analytics in financial markets, contributing to better investment strategies and reduced market risks.

Key words

Deep learning, CuDNNGRU, CNN, Stock price prediction, Time series data.

I. **Introduction**

In recent years, stock price forecasting has been one of the major research areas in financial sector. Due to the extremely volatile characteristic of stock's data, stock price forecasting is extremely difficult. It is quite difficult to accu- rately capture price movement since the price movement of stocks appears to be very random. At the stock market, maximizing investment returns for stock investors is possible if future pattern of stock market can be identified with appropriate method. Efficient market analysis states that it is possible to determine all appropriate information about a stock from available information that is individually sufficient for stock price prediction (Schumaker and Chen, 2009). Financial analysts face a significant obstacle in analyzing the extensive and fluctuating stock data, which is noisy and non-stationary, and identifying any valuable patterns within it (Abu-Mostafa and Atiya, 1996). As a result, Machine Learning (ML) techniques for stock price forecast gained popular- ity. With the huge successful outcome of Artificial Neural Network (ANN) in various forecasting tasks, researchers are keen to establish high level Deep Learning (DL) models for better prediction of stock prices. DL models have been used for prediction in different domains including water level prediction, electricity's bill prediction, text sentiment analysis, stocks price prediction and so on (Thakkar and Chaudhari, 2021). Since traditional ANN and ML tech- niques are not able to memorise the historical context but recent DL techniques such as recurrent neural networks (RNNs) have this ability (Shinde and Shah, 2018).

Recently RNNs gained popularity over traditional ML approaches to pre- dict stock prices. Since RNNs suffer from vanishing gradient and exploding gradient descent problem, Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al, 2014) net- works became state-of-art models for stock price prediction. LSTM and GRU models can capture the long term dependency. They resolved the vanishing gradient and exploding gradient descent problem of RNNs. Convolution Neu- ral Network (CNN) (Lecun et al, 1998) is also very popular to discover the abrupt changes from the current window frame.

Our study's main goal is to improve stock price forecasting accuracy by utilising cutting-edge deep learning models, particularly the Bidirectional CuDNNGRU. The complicated linkages and complex temporal dependencies seen in financial data are sometimes difficult for traditional methods to rep- resent. Our research overcomes these constraints by offering a more thorough and reliable examination of changes in stock prices. GRUs are more adept at managing long-term dependencies than conventional RNNs, which makes them a good fit for identifying patterns and trends in financial data. By reduc- ing the number of dimensions in the data, the GRU layer produces a compact representation that keeps important temporal properties, which the CNN can process quickly. Unlike the traditional strategy where CNNs are usually applied first, this hierarchical approach improves the model's recognition of complex patterns crucial to financial forecasting. This study is important because it aims to enhance prediction performance and provides financial analysts and investors with useful tools and insights.

In this paper, we propose a hybrid model, BiCuDNNGRU-1dCNN, that utilizes the benefits of the Bidirectional Cuda Deep Neural Network Gated Recurrent Unit (BiCuDNNGRU) model, one-dimensional CNN framework and utilizing CUDA-enabled GPUs, analyzes the spatial-temporal data of stock prices to determine the anticipated starting price of the stock on the upcoming trading day. Additionally, our model boasts high scalability, adaptability, and accuracy. The primary contribution of this study is:

1. Introduce a new intelligent hybrid predictive model that offers high accuracy and scalability, suitable for both individual stock items and stock market indices.

2. The proposed model's BiCuDNNGRU layers can learn the time series wave shape nature of each observation, while the CNN layers can extract spatial data features of stocks from several observations.

3. Conduct experiments to evaluate our suggested model with CuDNNGRU, GRU, GRU-CNN, and GRU-DNN (Dutta et al, 2020; Sheheryar et al, 2021), five different datasets with different values are used, 4 from the pub- lic domain and 1 from subscription. According to experimental results, our suggested model performs better in terms of prediction accuracy than the other four deep learning techniques.

This paper has the following organization. The literature review can be found in Section 2. Our proposed model, BiCuDNNGRU-1dCNN, is described in Section 3. Section 4 explains our experimental setup, system model and per- formance evaluation metrics. The outcomes and comparisons of our suggested GRU model with other existing GRU based models are shown in Section 5. Section 6 presents a discussion on potential threats to validity, while Section 7 provides the conclusion of the paper.

2 Literature Review

For investors, predicting the stock price is essential, and is one of the most interesting issues for researchers. The financial sector requires a decision- making model that can effectively increase investment returns and reduce investment risks. Due to the highly volatile and nonlinear structure of stock price data, forecasting stock prices is a difficult endeavor. Forecasting stock prices is a complex task since they depend on a variety of variables, including national regulations, historical data, exchange rates, news, and the economy of the nation. Research in this area is extensive and to date there is no existing theory that can describe and analyze the effect (Fama, 1990; Hult et al, 2012). In recent years, DL techniques have gained popularity in the field of stock price prediction. These models can estimate future stock prices by learning from historical data patterns. DL-based models are preferred over traditional machine learning methods because stock price data has nonlinear properties (Yu and Yan, 2020). In this section, we will review the related lit- erature about using DL models in stock price prediction. In particular, we will review the Convolutional Neural Network (CNN) (Lecun et al, 1998), Deep Neural Network (DNN) (Yong et al, 2017), Recurrent Neural Network (RNN) (Hiransha et al, 2018), Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al, 2014) models.

Deep Neural Network (DNN) model in which information flow is unidirec- tional from the input layer to the output layer without going backward. Yong et al. utilized a DNN-based model to forecast the Straits Time Index (STI) prices of the Singapore stock market (Yong et al, 2017). The authors of the study reported that their DNN model attained a profit factor of 18.67 and a 70.83% probability of successful trades based on market simulation data. How- ever, the DNN model suffers from two major limitations. The first limitation of the model is its fully connected nature, which prevents it from capturing the underlying patterns of the feature space. Secondly, during the training phase, DNN experiences the vanishing and exploding gradient problems, leading to poor long-term pattern detection performance (Glorot and Bengio, 2010). For instance, past features become less significant during the weight updating process than the latest features, resulting in poor generalization performance.

Several variants of Recurrent Neural Network (RNN) models have been utilized for stock price prediction, with LSTM and GRU exhibiting better prediction accuracy. According to previous studies, LSTM and its variations have shown excellent performance when handling time-varying data with feed- back loops for the prediction of financial time series. In particular, it has been demonstrated that LSTM models excel at making weekly predictions in volatile stock data by finding underlying trends and making long-term forecasts (Shah et al, 2018). Therefore, LSTM models are currently the most widely used DL models for financial time series prediction (Sezer et al, 2020). In contrast to a model named SingleNet, Baeka et al.'s LSTM-based technique for stock price prediction (Baeka and Kim, 2018) reduced the MSE to 54.1%, the MAPE to 35.5%, and the MAE to 32.7%. According to research studies, by regulating the information transfer between memory blocks, LSTM models address the problem of vanishing and exploding gradients (Hochreiter and Schmidhuber, 1997). Nonetheless, they may fail to detect sudden changes in patterns, which can adversely impact the accuracy of their predictions (Selvin et al, 2017).

Most researchers prefer the GRU model due to its relatively simpler model training phase (Houdt et al, 2020). GRU model has been applied for stock price prediction recently with good prediction results. Similar to LSTM model, The GRU model mitigates the issues of vanishing and exploding gradients by regulating the flow of information through its memory blocks (Cho et al, 2014). GRU has a simpler structure when compared with LSTM model. Huynh et al. proposed to use Bidirectional Gated Recurrent Unit to predict stock price (Huynh et al, 2017). Their approach is based on both historical stock price data and online financial news. It achieved 60% accuracy in S&P500 index prediction and 65% accuracy in individual stock prediction. GRU for stock price prediction with multiple look back windows has been investigated by Saud and Shakya (Saud and Shakya, 2020). Their results showed that GRU performed better than LSTM. It is because GRU uses less parameters than LSTM and hence it is more robust to train and predict results with GRU. They reported that the Mean Absolute Percentage Error (MAPE) of Nepal Investment Bank (NIB) and Nabil Bank Limited (NABIL) are 4.74 and 4.71 respectively. However, the prediction accuracy of GRU can be affected by abrupt changes in patterns that may be missed by the model.

Several academics have suggested using CNN models to forecast stock values due to their outstanding performance in feature extraction from the time series data of current feature space (Selvin et al, 2017; Hiransha et al, 2018). In a study by Selvin et al., they used three separate datasets to test three stock price forecasting approaches built on RNN, LSTM, and CNN. In two of the datasets, their suggested CNN model achieved a lower error percentage (Selvin et al, 2017). On the National Stock Exchange of India and the New York Stock Exchange, Hiransha et al. assessed CNN, RNN, LSTM, and MLP for stock price forecasting (Hiransha et al, 2018). They asserted that by accurately fore- casting the stock market's closing price the following day, the CNN approach surpassed the other three models. The CNN model, in contrast to previous models, employs the present sliding window to find sudden iterations in pat- terns rather than relying on prior evidence for forecast. However, because the CNN model focuses on the present frame for predicting and is influenced by semi-clean data, it is unable to capture long-term memory (Lecun et al, 1998).

To anticipate systemic financial risk, Zhou et al. suggested a hybrid tech- nique based on CNN and BiGRU and Systemic Financial Risk Indicator (SFRI) (Zhou and Yan, 2020). The CNN-BiGRU technique can predict sys- temic financial risk more accurately. The stacked LSTM, LSTM, and BiGRU methods have lower Symmetric Mean Absolute Percentage Error (SMAPE) values compared to the proposed method by 0.02. However, the proposed approach outperforms the stacked LSTM in 30-step prediction with a 22.2% improvement in SMAPE value. The data used for all samples are sourced from relevant departments in China. Inspired by CNN model, Xukuan et al. proposed a novel approach based on the time convolution with LSTM (TC-LSTM) networks (Zhan et al, 2018). TC-LSTM achieved Mean Square Residual (MSR) value of 1.3958 in the American stock market statistics, which is 45.5 percent lower than LSTM and 12.3 percent lower than GRU. An LSTM and GRU- based hybrid model were proposed by Hossain et al. (Hossain et al, 2018). Their model achieved 0.00098 MSE value for stock price prediction. They used S&P500 dataset for their experiments.

Nti et al. proposed a multivariate hybrid deep neural network model, IKN- ConvLSTM, based on CNN and LSTM (Nti et al, 2021). It is a model for fusing information from several sources to predict stock prices. This model integrated stock-related data from three quantitative and three qualitative sources. The outcomes showed 98.31% prediction accuracy of the integrated dataset com- pared with the different dataset. This model had 63 parameters, it required more training time and computational resources. Aldhyani et al. proposed a hybrid model based on CNN and LSTM to forecast the closing prices of Tesla, Inc. and Apple, Inc (Aldhyani and Alzahrani, 2022). Over the past two years of data for these two companies were used in their experiments. Their CNN- LSTM model performed better than the individual deep learning LSTM model. Their CNN-LSTM model achieved highest R-squared value for both datasets, the R-squared values of Telsa and Apple are 98.37% and 99.48%, respectively. The fact that the Financial Market Survey sentiment data was not used in this study is one of its limitations. Liu et al. proposed deep learning neural network (DLNN) to model price charts as images (Liu et al, 2022). To forecast short- term changes in stock prices, DLNN can mimic the work of technical analysts. using stock charts and stock fundamentals. DLNN offers adaptable analytical tools for efficiently examining price charts. More importantly, while predicting future price movements, price fluctuations are regulated by various periods of past daily closing prices for stocks. The DL model achieved an average 55.46% accuracy on the validation dataset. Gao et al. proposed a DL approach called Multiple-Branch Convolutional Neural Network (MBCNN) (Gao et al, 2022). It combined with genetic algorithms to forecast the direction of an intended stock market index's overnight return. Multiple convolutional units were used to obtain features from each region. To enhance the prediction accuracy of MBCNN, they used genetic algorithms to improve the model parameters and network architecture. The model predicted the stock markets of Asia, America and Europe. Their experimental findings demonstrated that their suggested model performed better in terms of accuracy, F-measure and Sharpe ratio. Since their focus is "on forecasting the daily direction", their prediction prob- lem becomes a classification problem and hence they use F-measure to evaluate their model. According to Gao and colleagues, there is a huge room for the improvement of computational efficiency. Li et al. proposed a hybrid model based on Logistic Weighted Dynamic Time Warping (LWDTW), by extend- ing a Weighted Dynamic Time Warping (WDTW) method to capture the valuable observations when calculating distance matrices (Li et al, 2022). The proposed model achieved lowest averages MAPE of 0.1278, MAE of 0.0536, MSE of 0.0059, RMSE of 0.0745 and highest average of R2 0.9517. Zhang et al. proposed a hybrid model based on variational modal decomposition (VMD), sample entropy (SE), and GRUs (Zhang et al, 2023). They used West Texas Intermediate daily closing oil prices as dataset. VMD-SE-GRU frame- work achieved RMSE of 0.6735, MSE of 0.4585 and MAPE of 0.8059. Fang et al. proposed a DL approach called VIX-Lasso-GRU Model, which is based on GRU by adding VIX information and a method called Least absolute shrink- age and selection operator (Fang et al, 2024). They used three dataset, namely SSE 50, CSI 300 & CSI 500. According to Fang and colleagues, they proved the feasibility of VIX and Lasso algorithm in optimizing stock forecasting model.

Eapen et al. proposed a hybrid CNN-Bi-directional LSTM model to improve stock market forecasting, which achieved a 9% improvement in predic- tion compared to a DL model using a single pipeline while avoiding overfitting on the difficult S&P500 dataset (Eapen et al, 2019). In another study, Lu et al. proposed a hybrid CNN-LSTM model for stock price forecasting (Lu et al, 2020). To predict stock prices, they used an LSTM model, and to extract char- acteristics from the input data, they used a CNN model. In their comparisons of MLP, CNN, RNN, LSTM, and CNN-RNN models, they found that their pro- posed CNN-LSTM model outperformed the others with the lowest MAE value (27.564) and RMSE value (39.688) and a coefficient of correlation that was close to 1.However, they used stock closing prices for their forecasting model, there is still opportunity for improvement in terms of MAE and RMSE values. Another hybrid model, CNN-BiLSTM-AM, was proposed by Lu et al., which achieved the lowest error rates for stock price forecasting in terms of MAE (21.952) and RMSE (31.694) (Lu et al, 2021). Another hybrid model proposed by Kanwal et al., BiCuDNNLSTM-1dCNN, is based on bidirectional CUDA enabled LSTM and one dimensional CNN. Their approach achieved MAE and RMSE values under 1 for three datasets (i.e. Crude oil, CL=F and DAX) and also performed well for ˆGDAXI and ˆHSI datasets (Kanwal et al, 2022).

Due to the non-linear dataset and market volatility, stock price forecasting is intrinsically difficult. When it comes to capturing the intricate temporal cor- relations and patterns found in financial data, traditional methods frequently fall short. A problem like the complex relationships in the data pattern may not be sufficiently addressed by the models currently in use. By offering a more thorough study of the underlying trends and patterns in stock price movements, our suggested approach—which makes use of sophisticated models such as the Bidirectional CuDNNGRU—aims to address these shortcomings by capturing bidirectional dependencies. Furthermore, a bidirectional design can improve its performance by running the input series in both directions to identify any underlying trends (Schuster and Paliwal, 1997). Our method stands out because we first use a GRU model to extract features, followed by a CNN for further processing. GRUs are better at handling long-term depen- dencies compared to traditional RNNs, making them ideal for capturing trends in financial data. By applying the GRU first, we reduce the dimensionality of the data while keeping key temporal features, which the CNN then processes effectively. This layered approach helps the model recognize complex patterns crucial for financial forecasting. Unlike the typical method where CNNs come first, our novel combination has shown promising results with improved RMSE values in our experiments. Although the aforementioned models have shown promising results, there is still a demand for a precise forecasting model for projecting stock prices, and they take a lot of time to train.

3 Our Proposed Model - BiCuDNNGRU-1dCNN

We suggest using a hybrid DL based approach, namely BiCuDNNGRU- 1dCNN, to increase stock price prediction accuracy. Our proposed hybrid model combines the advantages of the Bidirectional CuDNNGRU (CUDA based Gated Neural Network) and CNN models, aiming at achieving high prediction accuracy. The model is shown schematically in Figure 1. The first component is the Bidirectional CuDNNGRU network. The CuDNNGRU model is a fast GRU implementation backed by the cuDNN library, developed by NVIDIA. The purposes of using cuDNN library is to speed up the machine learning process via parallel computation of the GPUs, made available by Google's Collaboratory. CuDNNGRU, like the chronological information and long-term dependencies of the data are well captured by GRU. However, infor- mation is transferred from the input to the output in one direction, usually referred to as the forward direction. We propose to use a bidirectional CuD- NNGRU component. By going backwards and forwards through the supplied time series data, this allows for extra training (allowing loop back to earlier layers) directions. The purpose is to increase the chance of extracting valuable features that cannot be "learnt" from the forward direction. The output from BiCuDNNGRU is subsequently given to the CNN layer, the second com- ponent. This CNN layer with several convolution kernels (the Conv1D and MaxPooling1D layers) is used to extract more ethereal position-invariant ele- ments from the information. Therefore, our proposed BiCuDNNGRU-1dCNN model (or, simply our proposed GRU model whenever it is clear from the context) benefits from BiCuDNNGRU and 1dCNN.

Our method specifically employs a GRU model first to extract features, fol- lowed by a CNN for further processing. GRUs handle long-term dependencies better than traditional RNNs, making them suitable for capturing trends and patterns in financial data. The GRU layer reduces the dimensionality of the data, creating a compact representation that retains essential temporal fea- tures, which the CNN can then process efficiently. This hierarchical approach enhances the model's ability to recognize complex patterns relevant to finan- cial forecasting. This differs from the conventional approach where CNNs are typically applied first. Our novel combination has yielded promising results, as evidenced by improved RMSE values in our experiments, demonstrating its potential efficacy in financial forecasting. We provide a brief overview of the Bidirectional GRU (BiGRU) and CNN layers, the main components of our suggested model, in the following subsections.

Fig. (1): Framework of BiCuDNNGRU-1dCNN Prediction Model

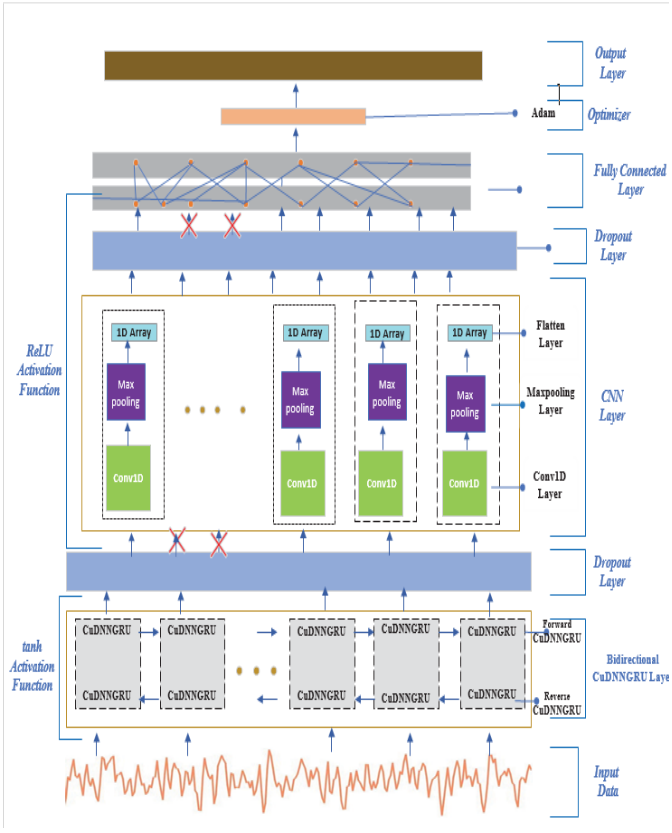
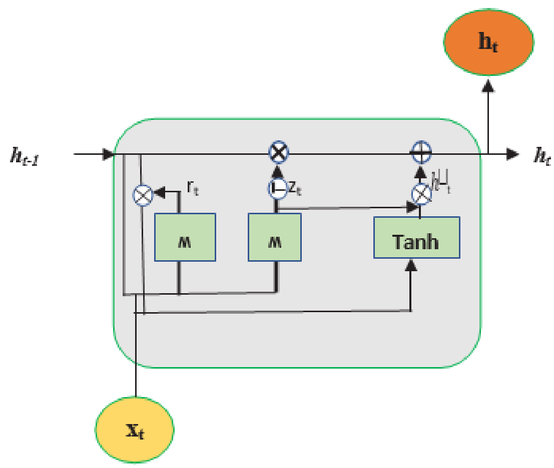


Fig. (2): GRU Architecture



3.1 Gated Neural Network (GRU) and Bidirectional GRU

The GRU model was proposed by Cho and colleagues (Cho et al, 2014). It is a recursive DL technique that does not vanish information quickly. Compared to LSTM, the GRU model is simpler. It has only two gates: the reset (r) and an update (z) gate in its architecture. GRU has one less gate than the LSTM. One of the most widely used recurrent neural network models today, GRU is essentially a version of LSTM. It mainly combines the input and forget gates of LSTM into a single update gate, limiting the training parameters and lowering the training complexity. The update gate preserves previous information to the current state. The input sequence of succeeding cell combines with preceding cell memory by reset gate. The reset gate has the ability to decide whether to integrate the previous information with the present condition. GRU has less parameters and a quicker training convergence time. The Mean Absolute Error (MAE) loss function can be used to calculate the discrepancies between the actual and expected values. Weights management of the neurons in GRU layer to achieve the minimum error handled by an Adam optimizer. Figure 2 depicts the GRU schematic presentation. The links between the pertinent quantities shown in Figure 2 are established by the equations that follow.

$$zt = \text{Sigmoid}\,(Wz[ht-1, xt] + bz) \qquad (1)$$

$$rt = \text{Sigmoid}(Wr[ht-1, xt] + br) \qquad (2)$$

$$\tilde{h}t = \tanh(Wh[rt + ht-1, xt] + bh) \qquad (3)$$

$$ht = (1 - zt) \cdot ht-1 + zt \cdot \tilde{h}t \qquad (4)$$

where z and r are update and reset gate, respectively, h˜t and ht are candidate hidden state and hidden state, respectively, Wz and Wr are the weight parameters to be trained, and bz and br are the noise vectors. These gates control the flow of information that can be used to store the expected values for arbitrary periods of time.

In our experiments, we used bidirectional GRU (BiGRU) since it makes full use of the context information from both directions. Fast GRU implementa- tion backed by cuDNN. Vector operations can utilise GPU parallelism via an interface provided by CUDA, or Compute Unified Device Architecture. Using CUDA, CuDNN implements kernels for complex matrix operations.

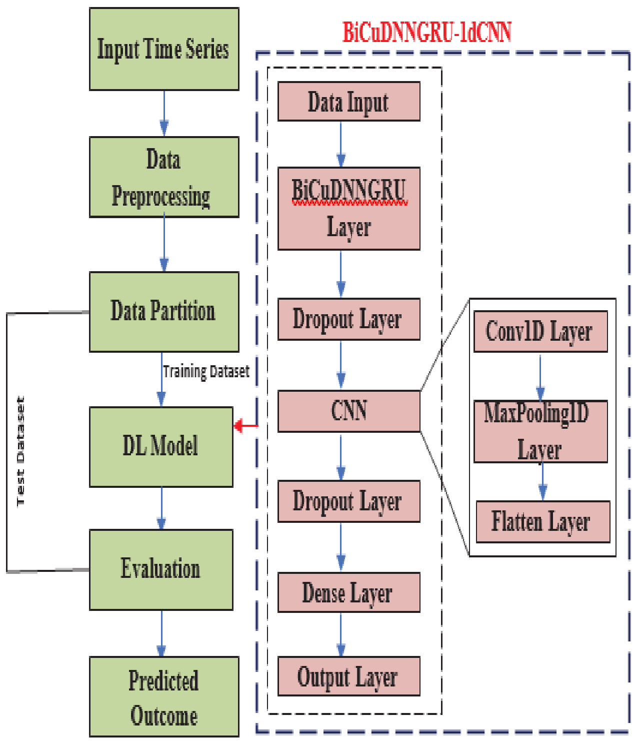
3.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) (Lecun et al, 1998) is an artificial neu- ral network designed to process high-dimensional data. Originally developed for image datasets, CNNs capture all regional relationships that are constant between different dimensions. Both one-dimensional data, like time series data, and multi-dimensional data, such pictures, are handled successfully by CNNs. The four main layers of a CNN are the convolutional layer, the max pooling layer, the flatten layer, and the fully connected layer. (Sezer et al, 2020). In the convolution layer, the input data is processed by a fixed-size sliding window, which captures local spatial features. The highest value of these local character- istics is subsequently obtained by the max pooling layer, which then represents the general features in a global context. Within its deep layers and memory cells, this knowledge is translated into more conceptual forms. The pooling pro- cedure reduces the number of nodes and the size of the matrices, even when the depth of the input matrix remains unchanged. This minimizes the parameters for the complete neural network. Highly abstract features are obtained through repeated convolution and pooling operations. The CNN model's multidimen- sional data output from the convolutional and pooling layers is converted into a one-dimensional vector using the flatten layer, making it suitable for use with the final fully connected layer.In order to learn information from the pre- vious layer and produce the model's output, the fully connected layer, which is made up of artificial neurones arranged in multiple layers, each fully con- nected to the next layer, uses weights, biases, and neurons. The output of the activation function determines the prediction results. The CNN model's per- formance is influenced by several hyperparameters, including the number of hidden layers, the number of neurons in each layer, kernel and filter size, activation functions, learning rate, the number of epochs, batch size, and dropout rate (Bergstra and Bengio, 2012). These factors must be carefully selected to optimize the model's performance.

3.3 BiCuDNNGRU-1dCNN Prediction Process

Our hybrid BiCuDNNGRU-1dCNN model's training and verification processes are covered in this part, and their flow is depicted in Figure 3. The following five important steps are discussed:

Fig. (3): BiCuDNNGRU-1dCNN Neural Network layer's Sequence



1. Data Preprocessing: The objective of data preoricessing is to prepare the data for further analysis by pre-processing it, which may involve han- dling missing values or outliers. There are various methods available in the literature to perform data pre-processing, including normalization, data transformation, and outlier removal. We used the two methods listed below to pre-process the data for our tests.

(a) Data Transformation: In the data preprocessing step, we handle any NaN values that may exist in the dataset. Since stock prices are repre- sented by numbers, any NaN entry is replaced by the dataset's mean of its neighboring values. This is done to ensure that NaN values do not unduly affect the value of records or the methodology for evaluation.

(b) Data Normalization: In order to improve the effectiveness of our proposed approach, we utilize the widely used technique of MIN-MAX scaling to normalize the feature vectors of the stock price dataset (Agar- wal and Muppalaneni, 2022; Sajjad et al, 2020; Senapati et al, 2018). This method is required to convert all values to a common scale since it lessens the impact of outliers on the dataset (Narayanadoss et al, 2019). The MIN-MAX scaling function ensures that values fall within the defined the logarithmic function's range (0,1]. Equation provides its formulation 5,

$$z_i = \frac{x_i - Min(X)}{Max(X) - Min(X)} \quad (5)$$

where zi denotes the normalized value, xi denotes the individual raw data, X = {x1, . . . , xn} denotes the set of all xi's, Min(X) and Max (X) are the minimum and maximum values of X.

2. Data Partition: The two sets of the dataset are created after preprocess- ing: a training set for training and validating the model, and a test set for evaluating the model's effectiveness. In our experiments, we split the dataset into training and test sets using a 90 % to 10 % ratio, respectively. (Chhajer et al, 2022; Jaggi

et al, 2021; Zou et al, 2007). For our dataset, we used a validation split technique. The data was first split into training and testing groups of 9:1 ratio. For validation, we additionally divided the training set into a 9:1 ratio during model training. This indicates that the test set was kept distinct while using a portion of the training data for validation. This is how we divided our dataset into training, validation, and testing sets.

3. Model's Training and Validation: For training, the training dataset is run through our suggested models. Since we use a sliding window of size 50 in our training, the first 50 consecutive days were used for training and the next day (51-th day) is used for validation. Using our previous example of 1000 days' data split into 900 days for training, the first instance is Day 1 to 50 used for training together with Day 51 for validation, the second instance is Day 2 to 51 for training with Day 52 for validation, . . . , and the last instance is Day 850 to 899 for training with Day 900 for validation. Once training is completed, the validation process is also completed.

The first step of the training process is initializing the hyperparameters of the model. In our architecture, the Bidirectional CuDNNGRU layer is responsible for feature extraction. To prevent overfitting, we use a Dropout layer with a properly chosen dropout rate. The One-dimensional CNN layer follows the feature extraction layer to capture local patterns. The final layer is a fully connected layer with Rectified Linear Activation Function (ReLU) that generates the prediction results.

In order to achieve optimal training results, it is recommended to exper- iment with different hyperparameters during the training process, such as the number of hidden layers, the number of neurons per layer, and the dropout rate. These hyperparameters can significantly affect the perfor- mance of the model, and are therefore crucial for obtaining the best results. Hyperparameters in neural networks, particularly in deep learning models, are chosen to minimise validation errors during training, according to Tran et al. (Thi Kieu Tran et al, 2020). The epoch size and the batch size are two more hyperparameters that might be taken into account. Since choosing the best settings for these hyperparameters can be time- and resource-consuming, research is still ongoing in this field. Ultimately, the choice of hyperparameters is arbitrary and depends on the researcher's expertise and knowledge of the problem at hand.

Algorithm 1 shows the algorithm on how to train our proposed BiCuDNNGRU-1dCNN model with our selection of the model benefits from a 1D CNN layer with MaxPooling1D and a flatten layer effectively compre- hend the sudden spatial patterns from the currently moving window (line 15). Finally, The output features from the preceding layer are then supplied as input to the Dense layer, which is a fully connected neural network. The learning rate and Adam optimizer are utilized in this process (line 28-33). Adding more layers to the dense layer section can empower our proposed GRU model's ability to classify the extracted features better. A second dense layer is used as the output layer (line 34). The pertinent hyperpa- rameters and their potential uses in Algorithm 1 are discussed below. The discussion includes the pertinent line numbers:

pISSN: 1684-0615 / eISSN: 2616-2814 AJRS

Algorithm 1 Hybrid BiCuDNNGRU-1dCNN Training Model

```
Input: Open Price of Stock Dataset
Bidirectional CuDNNGRU Layer=BiGL; CNN layer=CL; DropOut Layer=DL; Dense Layers=DeL;
epochs=e; batch-size=b; Weights=w; bais=b; Activation function=AC; Optimizer=O;
for trials = 1 : 5 do
    for epochs = 1 : e (with an early stopping call back function with patience value 10) do
        for batch-id=1 : b do
            for i in range(hp.Int(num-layers, 2, 20)) do
                if Layer[BiGL] == Bidirectional CuDNNGRU then
                    Create the w and b of BiCuDNNGRU at random;
                    Extract useful characteristics;
                    Compute the BiCuDNNGRU's hidden layers;
                end if
                if Layer[DL] == DropOut Layer then
                    Reduce node-to-node dependence on learning, avoid overfitting
                end if
                if Layer[CL] == Conv1D layer then
                    Create the w and b of CNN at random;
                    AF = relu
                end if
                if Layer[CL] == MaxPooling1D Layer then
                    Form the acquired features.;
                end if
                if Layer[CL] == Flatten Layer then
                    Ensure the output of the model is in one dimensional
                end if
                if Layer[DL] == DropOut Layer then
                    Reduce node-to-node dependence on learning, avoid overfitting
                end if
                if Layer[DeL] == Dense Layer then
                    units=hp.Int(units,min-value=32,max-value=512)
                    O= Adam
                    AF = relu
                    Randomly choose a learning rate from {0.01, 0.001, 0.0001}
                end if
                if Layer[DeL] == Dense Layer then
                    AC = relu
                    O = Adam
                end if
            end fori
        end forbatch-id
    end forepoch
    Store the BiCuDNNGRU-1dCNN trained model in a list of trained models
end fortrial
Return a BiCuDNNGRU-1dCNN trained model with the smallest MAE value, from the list
of trained models
```

GRU model's ability to classify the extracted features better. A second dense layer is used as the output layer (line 34). The pertinent hyperpa- rameters and their potential uses in Algorithm 1 are discussed below. The discussion includes the pertinent line numbers:

(a) Maximum number of trials (MaxTrial): This parameter defines the number of times the model is trained to generate a list of trained models. From this list, the model with the smallest Mean Absolute Error (MAE) is chosen as the best one. Increasing the number of training attempts is expected to enhance the forecasting accuracy of the model. In Algorithm 1, we set the number of training attempts to 5 (line 3).

(b) Number of epochs (epochs): In our model training, an epoch refers to one iteration over the entire training set. In this study,The number of epochs has been set to 32 (as indicated in line 4 of Algorithm 1).

According to (Yu et al, 2021), the selection of epochs in DNN can be supervised using "early stopping call back function" with a particular patience value, say 10. We adopt this approach in our training as well (line 4). If the call back function specifies that no additional enhance- ments can be made throughout the training process, the model will stop being trained. So, in case, the early stopping criteria cannot be satisfied, the training goes through all 32 epochs. However, after all the train- ing have been performed, we keep track of how many epochs were used throughout training. We discovered that between the range of 25 and 30 epochs, all training ceased.

(c) Batch size (batch): The batch size describes how many input sam- ples the model analyses before modifying its internal parameters. In our study, we experimented with different batch sizes, specifically 32, 64, and 128, but found that our proposed model performed best when trained with a batch size of 64 (line 5 in Algorithm 1). This result is consistent with prior research conducted by (Lu et al, 2021; Yang et al, 2020; Yan and Ouyang, 2018; Pouyanfar et al, 2018). Therefore, we believe that a batch size of 64 is justified for our model.

(d) Number of hidden layers (num-layers): For each trial in our exper- iment, we generate a random number between 2 and 20 hidden layers. (line 6). The quantity of covert layers are a critical hyperparameter between the input and output layers, and it may differ in each trial. This approach is motivated by two reasons. Firstly, according to (Yadav et al, 2019), the model's stability increases with the presence of more than one concealed layer. Secondly, researchers have set the number of hidden layers to different values, such as 2 (Yang et al, 2020), 5 (Bao et al, 2017), a random number between 1 to 7 (Yadav et al, 2019), or 10 (Singh and Srivastava, 2017), and reported that their trained models produced accurate forecasts.

(e) Dropout Layer: The dropout layer is an important approach for reg- ularization in deep learning models that helps to reduce interdependent learning between the nodes, thereby preventing the model from overfit- ting. The dropout rate, which is a number between 0 and 1, determines the proportion of nodes that are randomly ignored during training. In accordance with prior studies, the dropout rate was set to 0.2 or 20% in our experiments (Passricha and Aggarwal, 2020; Zhen et al, 2019; Fernandes et al, 2017)

(f) Optimizer: The optimization function is responsible to update model components like weights and learning rates in order to minimize the loss function.Our experiments utilize the Adam optimizer (line 27) due to its computational efficiency and suitability for large datasets. Additionally, Adam optimizer has been shown to perform well for forecasting problems in previous research (Fama, 1990).

(g) Learning rate: In the context of machine learning, the learning rate controls how frequently the model weights are modified to reflect the predicted inaccuracy. A smaller learning rate can result in more accurate model convergence but may require more epochs to train. In our exper- iments, we randomly select the learning rate from a set of 0.01, 0.001, and 0.0001 (line 29). This is consistent with prior research (Houdt et al, 2020; Lu et al, 2021; Singh and Srivastava, 2017), and it is generally advised to select a learning rate less than 0.1 (Hastie et al, 2009).

In our experiment, we aim to select the best prediction model among several trained models obtained. To do so, we use an evaluation criterion based on the mean absolute error (MAE), which is defined by Equation 6. Among all the trained models, the most effective trained model will be the one with the lowest MAE value.

$$MAE = \frac{1}{n}\sum_{i}^{n}(y_i - \hat{y}_i) \qquad (6)$$

The notation yi represents the true values from the dataset, and yˆi represents the values that the trained model anticipated.

4. Model Evaluation: After obtaining the ”best” trained model, it is used to predict values on the test dataset, and the performance of the model is evaluated using various evaluation metrics. We employ two widely used metrics in our experiments: mean absolute error (MAE) and the root-mean-square error (RMSE). A lower value reflects a better fit. The RMSE evaluates the difference between the original and projected values. It is a common statistical metric used in study on temperature, air quality, and weather predictions. MAE, on the other hand, is employed to strike a balance and increase trust in the values attained. Equation 6 has the formula for MAE, while the formula for RMSE is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad (7)$$

where yi denotes the actual values and yˆi denotes the predicted values from the trained model.

5. Model Prediction: Once the model is trained and the evaluation metric shows that it is functioning properly, it can be used to make forecasts on current, unforeseen data.

4 Experiments

We evaluate the predictive accuracy of our hybrid model BiCuDNNGRU- 1dCNN, drawing a comparison with four more GRU-based models in our experiments to assess its performance. These models include GRU-CNN, GRU- DNN, CuDNNGRU, and GRU. We evaluate these models on five different datasets, which are discussed in Section 4.1. The existing GRU-based models

have been previously used in stock price prediction studies (Dutta et al, 2020; Sheheryar et al, 2021).

According to the “Data Preprocessing” step as discussed in Section 3.3, we first pre-process the dataset by (1) transforming any NaN values and (2) normalizing the raw data, hoping to obtain better prediction results. For model training, verification and testing, we next divided the pre-processed dataset into training, verification and test datasets with a ratio of 90 % to 10 %.

Thirdly, we create five DL models, namely BiCuDNNGRU-1dCNN, GRU- CNN, GRU-DNN, CuDNNGRU, and GRU, with their respective hy- perparam- eters or parameters prepared for training. More details on these values can be found in Section 4.2. We choose a 50-step lookback window time step for our BiCuDNNGRU-1dCNN model. Fourthly, we train each DL model with the training dataset. Finally, we evaluate the performance of the trained model on the test dataset using two evaluation metrics, namely RMSE and MAE, after obtaining the best-trained model.

In this study, we trained and tested all five hybrid DL models using Keras in Python 3.8.5. To accelerate the processing speed and perform high-speed matrix calculations, we utilized the cuDNN library of the GPU-based Tensor- flow on Google Collaboratory. It is worth noting that Google's Collaboratory provided the computer hardware for the experiments, as it offers a free Python environment that requires minimal setup and runs entirely in the cloud. The experiments were performed on a desktop PC equipped with an Intel Core i7-8565U CPU, operating at a clock speed of 1.99 GHz, and 8 GB of RAM.

4.1 Datasets

In this study, we utilized five datasets for our experiments, which can be categorized into two groups. The first group consists of three datasets, each representing an individual stock item, while the second group contains two datasets representing the overall performance of the stock market. The stock price data used in these datasets is provided on a daily basis. The details of three datasets for the individual stock items are as follows:

1. Crude Oil: March 30, 1983 – August 15, 2018 (8884 data points).

2. Crude Oil (CL=F1): August 23, 2000 – January 15, 2021 (6263 data points).

3. Global X DAX Germany ETF (DAX1): October 23, 2014 – December 31, 2020 (1560 data points).

We have two datasets for the performance index of the stock market:

1. DAX Performance-Index (ˆGDAXI1): January 3, 2001 - July 16, 2021 (5508 data points).

2. Hang Seng Index (ˆHSI1): January 3, 2001 - July 16, 2021 (5390 data points).

The Crude Oil dataset used in this study was purchased through a paid sub- scription from Metastock.com. The remaining four datasets, namely CL=F,

Table (1): Sample Dataset of CL=F Dataset

Date	Open	High	Low	Close	Volume
09/07/2021	73.26	74.76	72.72	74.55	422154
12/07/2021	74.73	74.93	73.16	74.09	397931
13/07/2021	74.18	75.51	73.68	75.25	444078
14/07/2021	75.16	75.44	72.20	73.12	528360
15/07/2021	72.95	72.95	71.40	71.65	368679
16/07/2021	71.48	72.30	70.41	71.80	368679

DAX, ˆGDAXI, and ˆHSI, were obtained from YAHOO finance and are pub- licly available. All datasets contain information on daily stock prices, including open price, high price, low price, close price, and volume. Table 1 provides a sample dataset for Crude Oil (CL=F) from YAHOO Finance..

The collected dataset has five features, which are

1. Open: The first stock transaction after the stock exchange opens for that day's trading.

2. High: Highest price of a stock in entire trading day.

pISSN: 1684-0615 / eISSN: 2616-2814 AJRS

3. Low: Lowest price of a certain stock in entire trading day.

4. Close: Closing price of stocks for the specific trading day.

5. Volume: Volume of stocks traded in the specific trading day.

4.2 Experimental Processes and Settings

The five models that were examined in our experiments are shown in Table 3 along with the values assigned to each of their hyperparameters. Table 3 presents the five models evaluated in our experiments along with their respec- tive hyperparameters values. These models are trained first and then used to perform stock price prediction for the next trading day. This is a common practice to predict the price for the next trading day (Lu et al., 2021; Islam and Nguyen, 2020; Vogl et al., 2022).

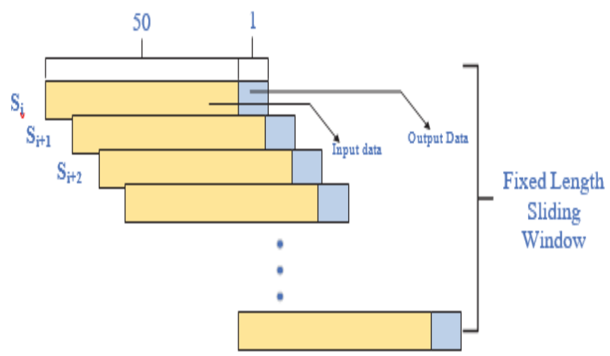
We set the same number of epochs, layers, and batch sizes for all five DL models during training to provide a fair comparison. Once the training is complete, we select the best trained model with the smallest MAE for each DL model and use them for comparison with each other.

4.3 Sliding Window

It is better to use the concept of sliding window in the training algorithm for prediction of long-term time series data, in which the data is split into several intervals (based on the sliding window) for training (Zivot and Wang, 2006). As seen in Figure 4, in this research, we use a constant length of 50 units as the time step of our sliding windows and the sliding windows are moving forward by one-unit step.

In other words, for the ith sliding window Si, the input data involve Di to Di+50 in which the first 50 data items are for ”training” whereas the 51-th data is used to compare the predicted result during the training. This series of sliding windows goes ahead one step at a time till the end of the time series.

Fig. (4): Sliding Window



5 Results and Comparisons

In this section, we compare our proposed model, BiCuDNNGRU-1dCNN, with other GRU based models, GRU-DNN, GRU-CNN, CuDNNGRU and GRU (Sethia and Raut, 2019; Zulqarnain et al, 2020). The purpose of this experiment is to depict that our proposed approach yields a better result that these GRU based models.

Individual stock price datasets and stock market performance index datasets are the two different types of stock price datasets we use in our study, as indicated in Section 4.1. The following two subsections, the first of which deals with specific stock items and the second with the stock market performance index, present our findings.

As discussed in Section 3.3, all five DL models, namely, BiCuDNNGRU- 1dCNN, GRU-DNN, GRU-CNN, CuDNNGRU and GRU, have been trained using each dataset, and the “best” trained models are returned for evaluation based on RMSE and MAE values.

5.1 Performance on individual stock item dataset (Crude Oil, CL=F, DAX)

Figures 5 and 6 respectively, exhibit the validation and test accuracy of the five DL trained models based on the Crude Oil, CL=F, and DAX datasets, demonstrating the RMSE and MAE values. We have drawn the following six key conclusions from these two figures:

1. From Figure 5, we have the following three observations related to their performances on the RMSE value.

(a) The DL model we proposed, BiCuDNNGRU-1dCNN, achieved the lowest RMSE values for both validation and test accuracy of the Crude Oil, CL=F and DAX datasets. Therefore, our model outperformed the four other DL models by using RMSE on these datasets to quantify prediction accuracy.

(b) The GRU-CNN trained model exhibits superior performance to the other three DL trained models (excluding our proposed GRU model) in terms of RMSE for the Crude Oil, CL=F, and DAX datasets' test and validation accuracy.

Fig. (5): Individual stock item: RMSE values of trained models

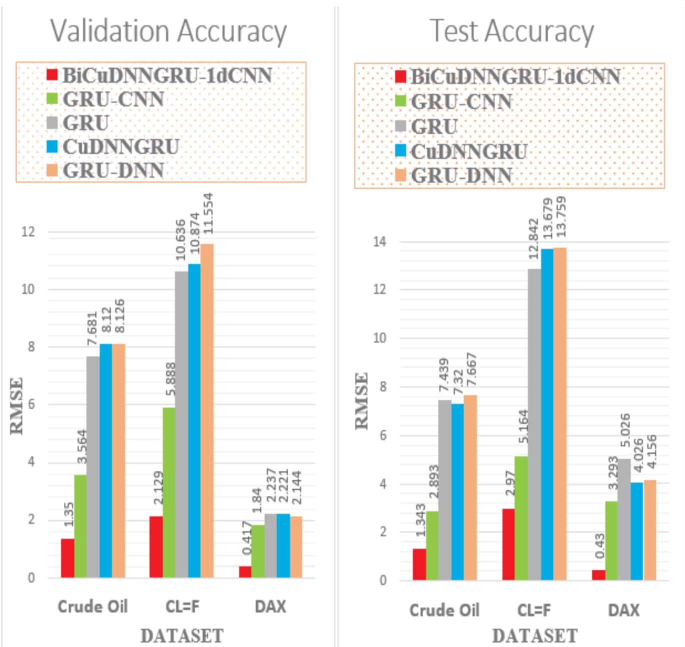
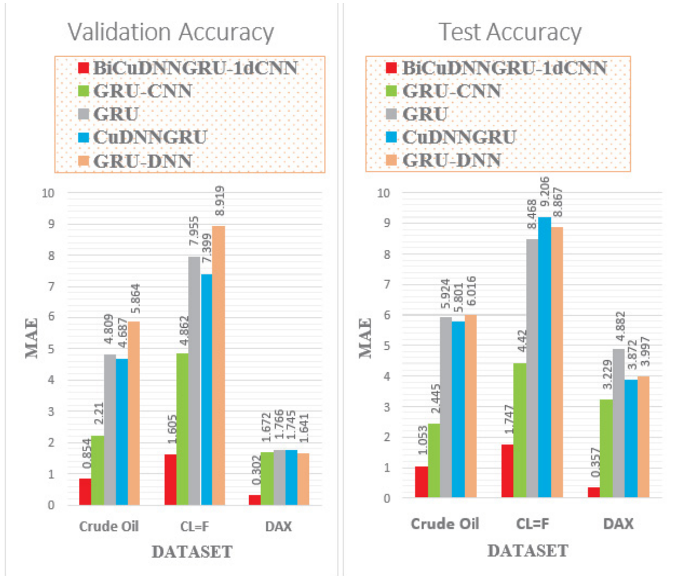


Fig. (6): Individual stock item: MAE values of trained models



(c) The GRU trained model has lower values for RMSE than the remaining CuDNNGRU and GRU-DNN trained models for both validation and test accuracy of the Crude Oil, CL=F, and DAX. For the Crude Oil and CL=F datasets, the RMSE value of the GRU-CNN model is around half that of the GRU model.

2. The following three observations, which are related to their performances on the MAE value and are comparable to the case of RMSE, can be drawn from Figure 6.

(a) Our proposed BiCuDNNGRU-1dCNN trained model outperforms the other four trained models in terms of having the lowest MAE values for both validation accuracy and test accuracy of the Crude Oil, CL=F, and DAX datasets. Therefore, our model exhibits its exceptional accuracy of forecast in relation to MAE compared to the other DL models on our datasets.

(b) From the remaining four DL-trained models, GRU-CNN performs better than the other three models nearly for all the cases except one. For the validation accuracy of DAX, the MAE value of GRU-CNN is 1.672 which is slightly bigger than that of GRU-DNN.

(c) The performance comparison of GRU-DNN, GRU, and CuDNNGRU models is not straightforward. For instance, in terms of validation accu- racy for Crude Oil dataset, the CuDNNGRU model outperforms the GRU-DNN and GRU models. However, for the CL=F test dataset, CuDNNGRU has a higher MAE value than the other two models. Addi- tionally, CuDNNGRU has a lower MAE value for Crude Oil and DAX datasets when compared to the other two models.

The Crude Oil, CL=F, and DAX datasets' original values as well as the predicted values of the five DL trained models are plotted against the dates in the Figures 7, 8 and 9 respectively. The following three observations are ours:

1. The red line in all of the figures closely follows the black line, which rep- resents the original values. This shows that the anticipated values of our proposed BiCuDNNGRU-1dCNN model have a high level of similarity with the original prices, demonstrating the model's excellent prediction accuracy.

2. The green line in each of these graphs represents the GRU-CNN trained model, which closely matches the original line however not exactly the same as our suggested GRU model. This clarifies why the performance of the GRU-CNN model is decent but not as exceptional as our proposed BiCuDNNGRU-1dCNN model.

3. The three other models, represented by their respective predicted value lines, deviate noticeably from the original values, more so than the lines for our proposed GRU model (red) and the GRU-CNN model (green) in each of these figures. Thus, the performance of the remaining three models is inferior to that of both our proposed GRU model and the GRU-CNN model.

To summarize, our study shows that the BiCuDNNGRU-1dCNN model we proposed outperforms the other four DL models, in terms of both RMSE and MAE metrics, on the Crude Oil, CL=F and DAX datasets used for validation and testing.

Fig. (7): Crude Oil (832 days): Original and predicted values of trained models

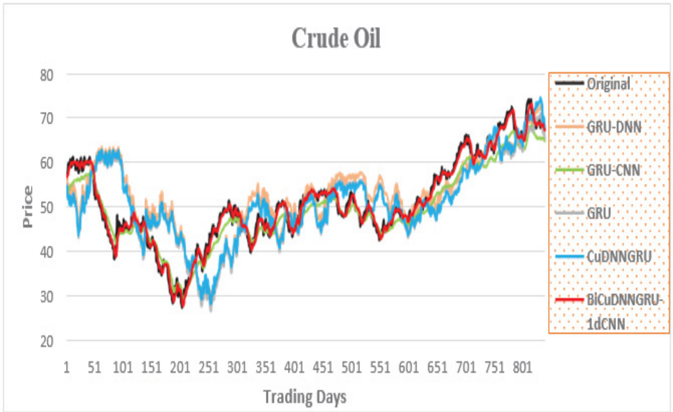


Fig. (8): CL=F (480 days): Original and predicted values of trained models

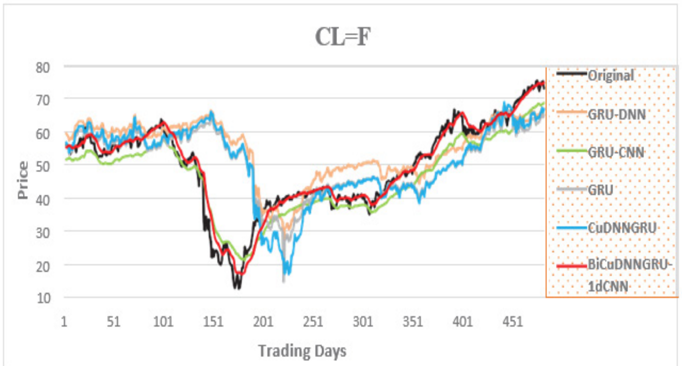
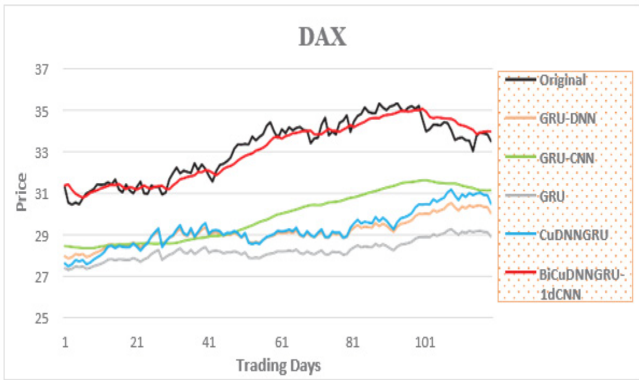


Fig. (9): DAX (119 days): Original and predicted values of trained models



5.2 Performance on stock market's performance index (ˆGDAXI and ˆHSI)

We will now present the outcomes of the five DL models in terms of their RMSE and MAE values, which are comparable to the study on individual stock prices.

The RMSE and MAE values of the validation and test accuracy for the five DL trained models on the ˆGDAXI and ˆHSI datasets are shown in Figures 10 and 11 respectively. These two figures allow us to draw the following two key conclusions about how well the models performed in terms of the RMSE value:

1. From Figure 10, the following two observations are ours:

(a) Outperforming all other DL models, our suggested BiCuDNNGRU- 1dCNN trained model exhibits the least RMSE values for both the validation and test accuracy of the ˆGDAXI and ˆHSI datasets.

(b) Performance of the other four DL models, excluding our suggested BiCuDNNGRU-1dCNN trained model is not consistent across the two datasets. For instance, the GRU based trained approach performs better than the remaining three DL models in terms of RMSE for the ˆGDAXI validation accuracy, but not for the ˆHSI validation accuracy, where it actually ranks second to last. Moreover, for the ˆHSI test accuracy, the GRU-CNN trained model appears to be the best among the four, while for the ˆGDAXI test accuracy, the GRU-CNN trained model performs better for validation accuracy but worse for test accuracy than the remaining three models.

2. From Figure 11, we have the following three observations related to their performances on the MAE values:

(a) In the ˆGDAXI and ˆHSI datasets, The MAE values for our suggested BiCuDNNGRU-1dCNN trained model's validation and test accuracy are the lowest as compared to other four DL trained models. Thus, regarding prediction accuracy in terms of MAE, our proposed model is the best performer among the five DL models investigated on these two datasets.

(b) The GRU-DNN trained model surpasses the other three models in terms of MAE for both the ˆGDAXI and the ˆHSI test accuracy, with the exception of the ˆHSI validation accuracy, where its performance is less obvious, among the other 4 DL trained models excluding our suggested BiCuDNNGRU-1dCNN.

(c) The performance of GRU-CNN, GRU and CuDNNGRU is not consis- tent across all datasets. For instance, for ˆGDAXI validation and test accuracy, the GRU trained model outperforms the GRU-CNN and CuD- NNGRU trained models in terms of MAE values. However, for ˆHSI test accuracy, the performance of the GRU-CNN model is worse compared to the other two models.

Fig. (10): Stock market performance index: RMSE values of trained models

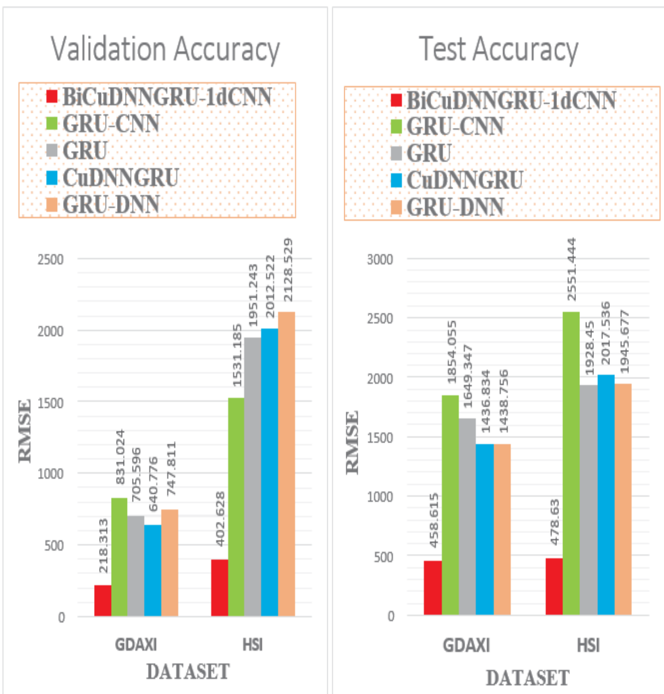


Fig. (11): Stock market performance index: MAE values of trained models

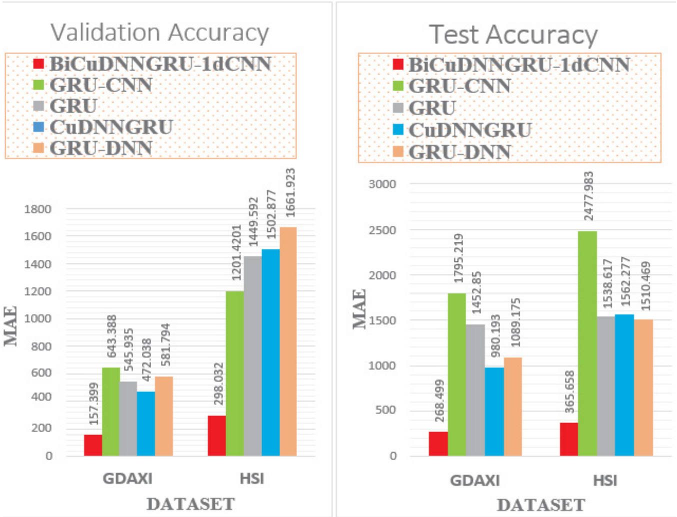


Fig. (12): ˆGDAXI (500 days): Original and predicted values of trained models

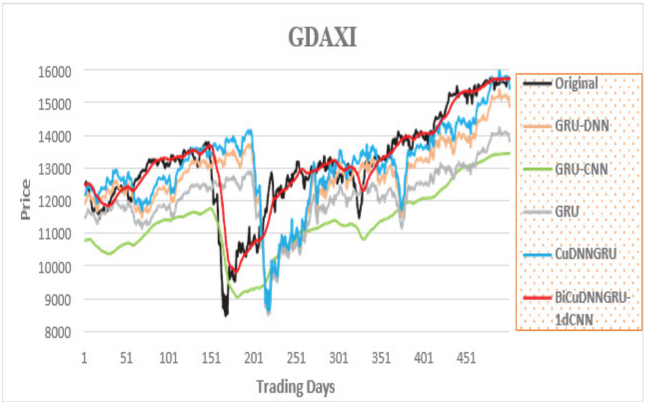
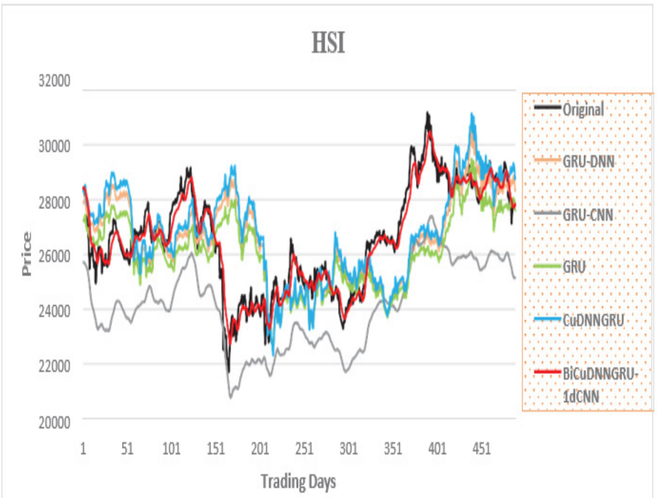


Fig. (13): ˆHSI (487 days): Original and predicted values of trained models



In a similar manner as with individual stock items, the predicted and orig- inal values of the five DL trained models for the ˆGDAXI and ˆHSI datasets are shown in Figures 12 and 13. We can infer the following four key findings from these figures:

1. We can deduce the following two findings about the ˆGDAXI dataset from Figure 12.

(a) In every point, the black line representing the original values are closely followed by the red line representing the anticipated values of our sug- gested BiCuDNNGRU-1dCNN model. The two lines almost completely overlap, showing how well the predicted values of our suggested GRU model match the initial prices. As a result, the GRU model we've suggested offers excellent prediction accuracy.

(b) The remaining four models' anticipated values, however, show greater divergence from the original (black) line. Their performance is therefore inferior to that of our suggested GRU model.

2. Regarding the ˆHSI dataset, we can note the two observations based on Figures 13.

(a) Our proposed BiCuDNNGRU-1dCNN model's projected values (red line) closely follow the original values (black line) throughout. In fact, the two lines are nearly identical in most places, which indicates that the model's predicted values are very close to the original prices. As a result, our proposed GRU model demonstrates excellent forecasting accuracy.

(b) Compared to the BiCuDNNGRU-1dCNN (red) line, the GRU-CNN model's line closely approximates the original line. This suggests that our suggested BiCuDNNGRU-1dCNN model outperforms the GRU-CNN model in performance. The three remaining models' predicted value lines also depart considerably more from the original (black) line and from the BiCuDNNGRU-1dCNN (red) and GRU-CNN lines. Therefore, these three models have poorer performance than our proposed GRU model.

In summary, our study reveals that the proposed BiCuDNNGRU-1dCNN model outperforms the other four DL models in terms of RMSE and MAE for both ˆGDAXI and ˆHSI datasets (validation and test accuracy). Additionally, the proposed GRU model consistently exhibits the best prediction accuracy among all models considered in this investigation.

The following table 4 shows a 42% to 91% reduction in RMSE value obtained by the proposed model.

The nine figures shown in this study show conclusively that our suggested BiCuDNNGRU-1dCNN framework can forecast stock prices with high accuracy, which can help traders make wise investment decisions. However, the performance of the other models is not consistent across different datasets.

To sum up, our suggested GRU based model produces the best outcomes for forecasting the price of both specific stocks and stock market indices, while the performance of the other models varies depending on the dataset.

6 Discussion: Threat to Validity

This section addresses some of the potential threats to the internal and external validity of our experiments.

One threat to external validity is that accurate predictions may require a sufficiently time series data in plenty, as evidenced by our findings for the DAX dataset. This is a limitation that is common to many DL models.

There are three risks to internal validity to take into account. In order to uncover historical patterns, it is crucial to carefully choose the dataset for training. To mitigate this, we experimented with different partition sizes ranging from 70% to 90% and selected the ideal choice for training our model. Secondly, Although training the models with various hyper-parameters can produce various results, choosing the best hyperparameters is a time-consuming and individualized procedure. To overcome this, we conducted various trial exper- iments using different combinations of hyperparameters (such as epoch size, batch size, dropout layer, number of hidden layers, learning rate, optimizer selection, and max trials) and selected the best model after many iterations. Thirdly, as we only used five datasets in our experiments, our results may not be generalizable to all stock prices. However, given the vast number of stock prices in the financial market, it is impractical to analyze all of them before proposing a model. Therefore, to show the broad application of our method- ology, we chose two types of stock prices: individual stock items and stock market performance indices.

The quality and quantity of the data utilised have a significant impact on the hybrid BiCuDNNGRU-1dCNN model's performance. Financial data can be noisy, and incomplete or low-quality data may cause the model to perform poorly. Furthermore, training the hybrid model uses a large amount of computational resources, particularly when GPU acceleration with CUDA is used. For academics or practitioners with restricted access to high-performance computer infrastructure, this could be a limitation.

In conclusion, although there are some potential threats to the validity of our experiments, we have taken steps to address them, and our proposed model has demonstrated promising results for stock price prediction.

7 Conclusion

Stock price forecasting has been very challenging due to the complex behaviour of stock price patterns, which affect the price prediction. In past decade, neural networks have shown promising outcomes in dealing with sequential data prediction. Many modifications of neural networks are proposed to anal- yse stock prices.To predict stock prices in this study, we suggest the hybrid BiCuDNNGRU-1dCNN model because the model (1) is good in handling time series data spatiotemporal matrix data and (2) can also extract the hid- den features, if any, in the datasets. Our proposed model exploits a hybrid model of BiCuDNNGRU and 1dCNN. Our proposed model achieved the low- est RMSE and MAE values for both training and test datasets. For example, it achieved an MAE value of 1.053 for the test dataset of Crude Oil, outperform- ing other models. Our experiments demonstrate a 42-91% reduction in RMSE compared to the standard CNN and GRU-based approaches, leading to more precise stock price predictions. In future research, we aim to expand our study by investigating the application of multivariate data to prediction. A more comprehensive knowledge of the determinants influencing stock prices can be obtained by integrating a number of variables, such as macroeconomic indica- tors, trading volumes, market sentiment, and geopolitical developments. Our goal is to improve financial forecasting and create more dependable research work includes the investigation of multivariate data for prediction decision.

References

Abu-Mostafa, Y. S., & Atiya, A. F. (1996). Introduction to financial forecasting. Applied intelligence, 6, 205-213. https://doi.org/10.1007/BF00126626

Agarwal, S., & Muppalaneni, N. B. (2022, April). Stock Market Price and Cryptocurrency Price Prediction. In 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) (pp. 1-6). IEEE. https://doi.org/10.1109/ICDCECE53908.2022.9793088

Aldhyani, T. H., & Alzahrani, A. (2022). Framework for predicting and modeling stock market prices based on deep learning algorithms. Electronics, 11(19), 3149. https://doi.org/10.3390/electronics11193149

Baek, Y., & Kim, H. Y. (2018). ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. Expert Systems with Applications, 113, 457-480. https://doi.org/10.1016/j.eswa.2018.07.019

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS one, 12(7), e0180944. https://doi.org/10.1371/journal.pone.0180944

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of machine learning research, 13(2).

Chhajer, P., Shah, M., & Kshirsagar, A. (2022). The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction. Decision Analytics Journal, 2, 100015. https://doi.org/ 10.1016/j.dajour.2021.100015

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078. https://doi.org/10.3115/v1/D14-1179

Dutta, A., Kumar, S., & Basu, M. (2020). A gated recurrent unit approach to bitcoin price prediction. Journal of risk and financial management, 13(2), 23. https://doi.org/ 10.3390/jrfm13020023

Eapen, J., Bein, D., & Verma, A. (2019, January). Novel deep learning model with CNN and bi-directional LSTM for improved stock market index prediction. In 2019 IEEE 9th annual computing and communication workshop and conference (CCWC) (pp. 0264-0270). IEEE. https://doi.org/10. 1109/ CCWC.2019.8666592

Fama, E. F. (1990). Stock returns, expected returns, and real activity. The journal of finance, 45(4), 1089-1108. https://doi.org/10.1111/j.1540-6261.1990. tb02428.x

Fang W, Zhang S, Xu C (2024) Improving prediction efficiency of chinese stock index futures intraday price by vix-lasso-gru model. Expert Systems with Applications 238. https://doi.org/10.1016/j. eswa.2023.12196

Fernandes, B., Silva, F., Alaiz-Moreton, H., Novais, P., Neves, J., & Analide, C. (2020). Long short-term memory networks for traffic flow forecasting: exploring input variables, time frames and multi-step approaches. Informatica, 31(4), 723-749. https://doi.org/10.15388/20-INFOR431

Gao, R., Zhang, X., Zhang, H., Zhao, Q., & Wang, Y. (2022). Forecasting the overnight return direction of stock market index combining global market indices: A multiple-branch deep learning approach. Expert Systems with Applications, 194, 116506. https://doi.org/10.1016/j.eswa.2022.116506

Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), Sardinia, Italy

Hastie, T. (2009). The elements of statistical learning: data mining, inference, and prediction. https://doi. org/10.1007/978-0-387-84858-7

Hiransha, M. E. A. G., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE stock market prediction using deep-learning models. Procedia computer science, 132, 1351-1362. https://doi. org/10.1016/j.procs.2018.05.050

Hochreiter, S. (1997). Long Short-term Memory. Neural Computation MIT-Press. https://doi.org/10.1162/ neco.1997.9.8.1735

Hossain, M. A., Karim, R., Thulasiram, R., Bruce, N. D., & Wang, Y. (2018, November). Hybrid deep learning model for stock price prediction. In 2018 ieee symposium series on computational intelligence (ssci) (pp. 1837-1844). IEEE. https://doi.org/10.1109/ SSCI. 2018.8628641

Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. Artificial Intelligence Review, 53(8), 5929-5955. https://doi.org/ 10.1007/s10462-020-09838-1

Hult, H. (2012). Risk and portfolio analysis: Principles and methods. Springer.

Huynh, H. D., Dang, L. M., & Duong, D. (2017, December). A new model for stock price movements prediction using deep neural network. In Proceedings of the 8th International Symposium on Information and Communication Technology (pp. 57-62). https://doi.org/10.1145/3155133.3155202

Islam, M. R., & Nguyen, N. (2020). Comparison of financial models for stock price prediction. Journal of Risk and Financial Management, 13(8), 181. https://doi.org/ 10.3390/jrfm13080181

Jaggi, M., Mandal, P., Narang, S., Naseem, U., & Khushi, M. (2021). Text mining of stocktwits data for predicting stock prices. Applied System Innovation, 4(1), 13. https://doi. org/10.3390/asi4010013

Kanwal, A., Lau, M. F., Ng, S. P., Sim, K. Y., & Chandrasekaran, S. (2022). BiCuDNNLSTM-1dCNN—A hybrid deep learning-based predictive model for stock price prediction. Expert Systems with Applications, 202, 117123. https://doi.org/10.1016/j.eswa.2022. 117123

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324. https://doi.org/10.1109/5.726791

Li, M., Zhu, Y., Zhao, T., & Angelova, M. (2022). Weighted dynamic time warping for traffic flow clustering. Neurocomputing, 472, 266-279. https://doi.org/10.1016/ j.neucom.2020.12.138

Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). A CNN-LSTM-based model to forecast stock prices. Complexity, 2020(1), 6622927. https://doi.org/10.1016/j.frl.2021.102209

Lu, W., Li, J., Wang, J., & Qin, L. (2021). A CNN-BiLSTM-AM method for stock price prediction. Neural Computing and Applications, 33(10), 4741-4753. https://doi.org/10.1007/s00521-020-05532-z

Narayanadoss, A. R., Truong-Huu, T., Mohan, P. M., & Gurusamy, M. (2019, April). Crossfire attack detection using deep learning in software defined its networks. In 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring) (pp. 1-6). IEEE. https://doi.org/10.1109/VTCSpring. 2019.8746594

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2021). A novel multi-source information-fusion predictive framework based on deep neural networks for accuracy enhancement in stock market prediction. Journal of Big data, 8(1), 17. https://doi.org/10.1186/s40537-020-00400-y

Passricha, V., & Aggarwal, R. K. (2019). A hybrid of deep CNN and bidirectional LSTM for automatic speech recognition. Journal of Intelligent Systems, 29(1), 1261-1274. https://doi.org/10.1515/jisys-2018-0372

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., ... & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. ACM computing surveys (CSUR), 51(5), 1-36. https://doi.org/10.1145/3234150

Sajjad, M., Khan, Z. A., Ullah, A., Hussain, T., Ullah, W., Lee, M. Y., & Baik, S. W. (2020). A novel CNN-GRU-based hybrid approach for short-term residential load forecasting. Ieee Access, 8, 143759-143768. https://doi.org/10.1109/ACCESS.2020.3009537

Saud, A. S., & Shakya, S. (2020). Analysis of look back period for stock price prediction with RNN variants: A case study on banking sector of NEPSE. Procedia Computer Science, 167, 788-798.
https://doi.org/10.1016/j.procs. 2020.03.419

Schumaker, R. P., & Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The AZFin text system. ACM Transactions on Information Systems (TOIS), 27(2), 1-19. https://doi.org/10.1145/1462198.1462204

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE transactions on Signal Processing, 45(11), 2673-2681. https://doi.org/10.1109/78.650093

Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017, September). Stock price prediction using LSTM, RNN and CNN-sliding window model. In 2017 international conference on advances in computing, communications and informatics (icacci) (pp. 1643-1647). IEEE. https://doi.org/10.1109/ICACCI. 2017.8126078

Senapati, M. R., Das, S., & Mishra, S. (2018). A novel model for stock price prediction using hybrid neural network. Journal of the Institution of Engineers (india): Series B, 99(6), 555-563. https://doi.org/10.1007/s40031-018-0343-7

Sethia, A., & Raut, P. (2019). Application of LSTM, GRU and ICA for stock price prediction. In Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2018, Volume 2 (pp. 479-487). Springer Singapore. https://doi.org/10.1007/978-981-13-1747-7-46

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

Shah D, Campbell W, Zulkernine FH (2018) A comparative study of LSTM and DNN for stock market forecasting. In: 2018 IEEE International Confer- ence on Big Data (Big Data), Seattle, Washington, USA, https://doi.org/ 10.1109/BigData.2018.8622462

Shinde, P. P., & Shah, S. (2018, August). A review of machine learning and deep learning applications. In 2018 Fourth international conference on computing communication control and automation (ICCUBEA) (pp. 1-6). IEEE. https://doi.org/10. 1109/ICCUBEA.2018.8697857

Singh R, Srivastava S (2017) Stock prediction using deep learning. Multime- dia Tools and Applications 76(8):18,569–18,584. https://doi.org/10.1186/s40537-020-00333-6

Thakkar, A., & Chaudhari, K. (2021). A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. Expert Systems with Applications, 177, 114800. https://doi.org/10.1016/j.eswa.2021. 114800

Thi Kieu Tran, T., Lee, T., Shin, J. Y., Kim, J. S., & Kamruzzaman, M. (2020). Deep learning-based maximum temperature forecasting assisted with meta-learning for hyperparameter optimization. Atmosphere, 11(5), 487. https://doi.org/10.3390/atmos11050487

Vogl, M., Rötzel, P. G., & Homes, S. (2022). Forecasting performance of wavelet neural networks and other neural network topologies: A comparative study based on financial market data sets. Machine Learning with Applications, 8, 100302. https://doi.org/10.1016/j.mlwa.2022.100302

Yadav, A., Jha, C.K., & Sharan, A. (2020). Optimizing LSTM for time series prediction in Indian stock market. Procedia Computer Science, 167, 2091-2100.

Yan, H., & Ouyang, H. (2018). Financial time series prediction based on deep learning. Wireless Personal Communications, 102, 683-700. https://doi.org/10.1007/s11277-017-5086-2

Yang C, Zhai J, Tao G (2020) Deep learning for price movement prediction using convolutional neural network and long short-term memory. Mathe- matical Problems in Engineering 2020:1–13. https://doi.org/10.1155/2020/ 2746845

Yong, B. X., Abdul Rahim, M. R., & Abdullah, A. S. (2017). A stock market trading system using deep neural network. In Modeling, Design and Simulation of Systems: 17th Asia Simulation Conference, AsiaSim 2017, Melaka, Malaysia, August 27–29, 2017, Proceedings, Part I 17 (pp. 356-364). Springer Singapore. https://doi.org/ 10.1007/978-981-10-6463-0 31

Yu, J., Zhang, X., Xu, L., Dong, J., & Zhangzhong, L. (2021). A hybrid CNN-GRU model for predicting soil moisture in maize root zone. Agricultural Water Management, 245, 106649. https://doi.org/10.1016/j.agwat.2020.106649

Yu, P., & Yan, X. (2020). Stock price prediction based on deep neural networks. Neural Computing and Applications, 32(6), 1609-1628. https://doi.org/10.1007/s00521-019-04212-x

Zhan, X., Li, Y., Li, R., Gu, X., Habimana, O., & Wang, H. (2018). Stock price prediction using time convolution long short-term memory network. In Knowledge Science, Engineering and Management: 11th International Conference, KSEM 2018, Changchun, China, August 17–19, 2018, Proceedings, Part I 11 (pp. 461-468). Springer International Publishing. https://doi.org/10.1007/978-3-319-99365-2 41

Zhang, S., Luo, J., Wang, S., & Liu, F. (2023). Oil price forecasting: A hybrid GRU neural network based on decomposition–reconstruction methods. Expert Systems with Applications, 218, 119617. https://doi.org/10.1016/j.eswa.2023.119617

Zhen, T., Yan, L., & Yuan, P. (2019). Walking gait phase detection based on acceleration signals using LSTM-DNN algorithm. Algorithms, 12(12), 253. https://doi.org/ 10.3390/a12120253

Zhou, Y., & Yan, J. (2020, July). A Hybrid Deep Learning Approach for Systemic Financial Risk Prediction. In International Conference on Computational Science and Its Applications (pp. 859-874). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-58799-4 62

Zivot, E., & Wang, J. (2006). Modeling financial time series with S-PLUS (Vol. 2). New York: Springer. https://doi.org/10.1007/978-0-387-32348-0 9

Zou, H. F., Xia, G. P., Yang, F. T., & Wang, H. Y. (2007). An investigation and comparison of artificial neural network and time series models for Chinese food grain price forecasting. Neurocomputing, 70(16-18), 2913-2923. https://doi.org/10.1016/j.neucom.2007.01.009

Zulqarnain, M., Ghazali, R., Ghouse, M. G., Hassim, Y. M. M., & Javid, I. (2020). Predicting financial prices of stock market using recurrent convolutional neural networks. International Journal of Intelligent Systems and Applications, 13(6), 21. https://doi.org/10. 5815/ijisa.2020.06.02

Declaration of conflict of interest:

The researchers declare and undertake that there is no conflict of interest with any person or institution. This research has not been previously published in any way, whether written, read, published, visual or audio.

Funding statement:

The Malaysia Ministry of Higher Education's Fundamental study Grant Scheme is helping to fund some of this study (FRGS/1/2015/ICT04/SWIN/02/1).

Biography of researchers

Anika Kanwal

Anika Kanwal earned her PhD in 2023, specializing in stock price prediction using neural networks. Since 2019, she has been serving as a sessional academic in the School of Science, Computing, and Engineering Technologies at Swinburne University of Technology, Melbourne, Australia. Her research primarily focuses on developing and optimizing machine learning and deep learning-based predictive models. She is particularly interested in exploring advanced neural network architectures and large language models (LLMs) to enhance predictive accuracy and improve interpretability in financial forecasting.

Email: anikakanwaal@gmail.com
ORCID: https://orcid.org/0000-0001-5495-4799

Man Fai Lau

Man Fai Lau received his BSc(Hons) from The University of Hong Kong and PhD from The University of Melbourne. He joined Swinburne University of Technology in 2000. He received 2 ARC DP research grants in 2005 (a 3-year ARC DP project) and 2007 (a 5-year ARC DP project). His currently research interests are on Data Mining and Testing Prediction Models. Currently, he is working on a Big Data project on testing Optimal Oil Drilling Prediction Models based on past 15+ years of oil drilling data.

Email: elau@swin.edu.au
ORCID: https://orcid.org/0000-0002-2271-6259

AhmadSamiAl-Shamayleh

Ahmad Sami Al-Shamayleh received the master's degree in Information Systems from The University of Jordan, Jordan, in 2014, and the Ph.D. degree in Artificial intelligence from University of Malaya, Malaysia, in 2020. He is currently an Assistant Professor with the Faculty of Information Technology, Al-Ahliyya Amman University, Jordan. His research interests include: Artificial Intelligence, Human Computer Interaction, IoT, Arabic NLP, Arabic sign language recognition, language resources production, the design and evaluation of interactive applications for handicapped people, multimodality, and software engineering.

Email: a.alshamayleh@ammanu.edu.jo
ORCID: https://orcid.org/0000-0002-7222-2433

Adnan Akhunzada

Adnan Akhunzada, a distinguished Senior Member of IEEE and Professional Member of ACM, brings 15 years of expertise in Research and Development (R&D) at the nexus of the ICT industry and academia. Renowned for his high impact publications, US patents, and commercial products, Dr. Akhunzada's patented innovations in cybersecurity and AI have secured multi-million-dollar projects with global entities such as Vinnova and EU Horizon. In 2023, Stanford University recognized him as one of the top 2% scientists globally for his outstanding scholarly contributions. Leveraging his robust cybersecurity skills and advanced technological knowledge, Prof. Akhunzada excels in solving industrial challenges and developing state-of-the-art security tools and frameworks. His expertise spans Cybersecurity & AI, Secure Future Internet, Secure & Dependable Software Defined Networks, and Large-Scale Distributed Systems (including Cloud, Fog, Edge, IoT, IoE, IIoT, CPS). Additionally, his work on Lightweight Cryptographic Communication Protocols, QoS/QoE, and Adversarial Machine Learning is shaping the future of secure and dependable systems.

Email: adnan.adnan@udst.edu.qa
ORCID: https://orcid.org/0000-0001-8370-9290

SebastianP.H. Ng

NG S.P.H. is an associate professor at Swinburne, specializing in Computer Science and Software Engineering. His research centres on Software Engineering Education and Software Testing. He's well-qualified with a PhD, Master's, PG Diploma, and bachelor's degrees spanning Information Technology, Computing, Science, and Education. He also holds senior membership in the Australian Computer Society.

Email: sng@swin.edu.au
ORCID: https://orcid.org/0000-0002-2074-6673

Siva Chandrasekaran

Siva Chandrasekaran is a Senior Lecturer in Computer and Software Engineering at Swinburne University of Technology. He is an elected member of the Academic Senate for 2021-2022 and 2023-2024 and serves as the Major Discipline Coordinator for Software Engineering. He holds the title of Fellow of the Institution of Engineers Australia and is a Chartered Professional Engineer in various areas. Siva Chandrasekaran has been bestowed as a senior member (SMIEEE) of the Institution of Electrical and Electronics Engineers (IEEE) and a member of the Australian Computer Society (ACS). Siva is an Associate Editor of the IEEE Education Society IEEE Access Section. He is the lead of the Internet of Things (IoT) Training Academy at Swinburne. Siva has made significant contributions to engineering education research and published more than 140 scholarly articles in national/international peer-reviewed journals and conference proceedings. His research focuses on AI, ML, cybersecurity and Industrial IoT.

Email: schandrasekaran@swin.edu.au
ORCID: https://orcid.org/0000-0003-2871-880X

Kwan Yong Sim

Kwan Yong Sim is Deputy Pro Vice-Chancellor (Academic), Professor at Swinburne University of Technology Sarawak Campus, Malaysia. is Deputy Pro Vice Chancellor (Aca- demic) at Swinburne University of Technology Sarawak Campus. His previous appointments include Executive Dean (Academic) – Interim, Head of School for School of Engineering, Associate Dean at the Faculty of Engineering, Computing and Science, Swinburne Sarawak, Malaysia Re- search Assessment (MyRA) Lead Auditor, Engineering Accreditation Council Panel of Evaluator. He is a registered Professional Engineer (P.Eng.) with the Board of Engineer Malaysia (BEM), Chartered Engineer (CEng) with the U.K. Engineering Council. His current research interests include Data Mining and Testing Prediction Models.

Email: ksim@swinburne.edu.my

Table (2): Description of system model

Algorithm	Layers	Neurons / Filter Kernal / Pool Size
BicuDNNGRU-1dCNN	Bidirectional cuDNNGRU	(512)
	Dropout Rate (2)	(0.2)
	Conv Layer	(64, 1)
	MaxPool Layer	(1)
	Flatten Layer	default
	Dense Layer	(32 – 512)
	Output Layer	(1)
GRU-DNN	GRU Layer	(32)
	Dropout Rate (2)	(0.2)
	Dense Layer (2)	(32, 8)
	Output Layer	(1)
GRU–CNN	GRU Layer	(32)
	Dropout Rate (2)	(0.2)
	Conv Layer	(64, 1)
	MaxPool Layer	(1)
	Flatten Layer	default
	Dense Layer	(50)
	Output Layer	(1)
CuDNNGRU	Bidirectional CuDNNGRU	(512)
	Dropout Rate	(0.2)
	Dense Layer	(32 – 512)
	Output Layer	(1)
GRU	GRU Layer (2)	(32, 32)
	Dropout Rate (2)	(0.2)
	Dense Layer	(50)
	Output Layer	(1)

All models employ 64 Batch Size, 32 as the Epoch, Adam as the Optimizer, ReLU as the Activation Function, MAE as the Loss Function, and Adam as the Optimizer.

Table (3): Test Accuracy of the Proposed Model and State-of-the-Art Models

Models	BiCuDNNGRU- 1dCNN vs GRU-CNN	BiCuDNNGRU- 1dCNN vs GRU-DNN	BiCuDNNGRU- 1dCNN vs CuD- NNGRU	BiCuDNNGRU- 1dCNN vs GRU
Crude Oil	53.58%	81.95%	81.65%	82.48%
CL=F	42.49%	76.87%	78.29%	78.41%
DAX	86.94%	91.44%	89.32%	89.65%
ˆGDAXI	85.04%	81.52%	72.61%	75.35%
ˆHSI	85.24%	76.23%	76.59%	75.79%