Information Sciences 594 (2022) 1–19

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

An ensemble of a boosted hybrid of deep learning models and technical analysis for forecasting stock prices

Amadu Fullah Kamara [a,b,*], Enhong Chen [a], Zhen Pan [a]

[a] School of Computer Science and Technology, University of Science and Technology of China, China
[b] Department of Mathematics & Statistics, Fourah Bay College, University of Sierra Leone, Sierra Leone

ARTICLE INFO

Article history:
Received 25 September 2020
Received in revised form 22 December 2021
Accepted 9 February 2022
Available online 16 February 2022

Keywords:
Attention-based convolutional neural network
Contextual bidirectional long short-term memory
Moving average
Moving average convergence-divergence curve
Moving average convergence-divergence histogram
Relative strength index

ABSTRACT

For several years the modeling as well as forecasting of the prices of stocks have been extremely challenging for the business community and researchers as a result of the existence of noise in samples and also the non-stationary behaviour of information samples. Notwithstanding these drawbacks with improved deep learning, it is now possible to design schemes that will efficiently perform the feature learning task. For this work, we proposed a brand-new end to end algorithm labeled EHTS toward solving the stock price forecasting problem. The $AB-CNN$ and $CB-LSTM$ modules extract features from the stock price dataset and soon after amalgamating the results. Thus, the output of the concatenation stage was feed into the concluding stage which is a stand-alone MLP module. The inclusion of the LSTM and Attention Mechanism in our architecture is to extract long-range and exceptionally long-term stock price information. We experiment the proposed algorithm on two popular stocks both from the NYSE stock market namely "Johnson & Johnson" code-named, "*JNJ*" and the Bank of America (BAC). In terms of the rMSE, MAE and MAPE error metrics, our proposed scheme gives the lowest error value in all for all datasets. Also, five percentage training window sizes are experimented and EHTS outperforms all the baseline schemes for the different window sizes in all the two datasets with the 70% window size having the highest performance. In terms of number of epochs, EHTS uses the lowest number of epochs for training than the other schemes in all the datasets. Finally, we as well study our stock's information to point out short-range trading opportunities by performing simulations on our stock price data. The metrics considered in the simulation are as follows: Moving Average (MA), Moving Average Convergence Divergence (MACD) curve, MACD histogram, Signal line, Relative Strength Index (RSI), Returns (R), Annual Returns (AR), Sharpe Ratio (SR), Annual Volatility (V), Maximum DrawDown (MDD) and Daily WinningRate (DWR). For all the aforementioned metrics, EHTS performs better than the baselines. Experimental results revealed that our proposed scheme outperformed the stand-alone deep learning schemes, statistical algorithms, and machine learning models from the beginning to the end.

* Corresponding author at: Department of Mathematics & Statistics, Fourah Bay College, University of Sierra Leone, Sierra Leone.
E-mail address: kamara@mail.ustc.edu.cn (A.F. Kamara).

https://doi.org/10.1016/j.ins.2022.02.015
0020-0255/

1. Introduction

Stock Markets, by definition, are communities that bring together traders that may want to buy or sell stocks. Stock markets are of two types, namely: the primary market and the secondary market. A primary market is a market where the introduction of new stocks takes place. A secondary market is one in which shareholders will trade with bonds that are already in possession. The Stock Exchange is always known for its explosive and unreliable nature in behaviour. An individual stock could be booming one day and struggling another day. The realisation of massive profit from selling stocks occurs when their prices are at their highest peaks and buying when their prices are at their lowest heights. A stock is simply a block of a company. The more one acquires stocks, the greater his/her ownership stack in the company. It is also interesting to know that stock price is a metric that measures a company's performance [1]. Time series are sequences of numerical information points of a particular variable that are measurable over some time [2]. In this paper, our primary focus is on the day-wise closing price variable. Nonetheless, time-series data are of two groups, namely, univariate and multivariate. A time series data that contains or considers only one variable is known to be univariate, while that which includes more than one variables is called multivariate.

However, there are a lot of existing linear models that have produced outstanding results for forecasting the securities market, including; Autoregression (AR), Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA) [3]. The main drawback of the models mentioned above is that they are dataset specific, i.e., a model that yields outstanding performance results for a particular stock will not perform well on another stock. Therefore, because of the above reason, there is no possibility for the identification of patterns and dynamics present in an entire stock market.

Developing accurate models is tough since the difference in stock price depends not only on a single factor but on several factors which include, including news, data from social media, fundamentals, company's production, state securities, historical price, as well as country's economics [4]. A forecast model that regards only one characteristic might not produce accurate results. Deep neural networks are excellent at approximating non-linear functions and can map non-linear functions better [5,6]. Deep learning algorithms have a kind of self-learning mechanism that helps identify both latent patterns and underlying data dynamics [7]. For the past few decades, the performance of deep learning-based models for stock price prediction has been outstanding. For [8,9], the authors showcased the applications of various deep learning models. Similar research of the disparate deep-learning benchmark algorithms to predict the securities market has been done already [10]. Also, a group of authors introduced Back propagation neural network into Log-periodic power law scheme to efficiently capture and forecast the volatility of stock market during crashes [11]. Artificial Intelligence (AI) has offered new methods to finance via its implementation in the forecasting of financial stock markets. A bulk of literature has tried to use AI and Machine Learning for the prediction of the returns and volatilities of the stock market. The investigations on the use of AI in financial stock market have a host of issues such as different investigation direction, findings, methodology, etc which were addressed by connecting the existing literature in a standardised review to address the application of AI in the forecasting of stock prices [12]. The training of deeper networks with financial stock market data is not difficult, the most important issue is, how efficient is the prediction of the actual data by a trained network. A group of authors proposed a dynamic algorithm that dynamically appraise and picks the prediction scheme for the stock movement trend prediction. The first phase of the algorithm consists of a set of prospective candidate predictors which are built based on the convolutional long short-term memory network by utilising different parameter values. The second phase is made of a kernel time-weighted fuzzy c-means clustering algorithm to arrange the real samples in order of importance to the target samples i.e. historical and target samples that are closely related have more more impact on the predictors. Then, the arranged real samples are used to appraise the candidate predictors, and the predictor with the best accuracy is selected to perform the target sample prediction [13].

However, due to the inception of a unique kind of Recurrent Neural Network (RNN), i.e., Long Short Term Memory (LSTM) [14,15] model, the sequences with long-term features can now be learned. The LSTM models possess the ability to hold previous information. Several such algorithms are currently in use for analysing financial time-series data [16,17].

For the past decade, a data mining technique named neural network has been one of the best methods used in various fields by researchers. A list of machine learning models such as multilayer perceptron, convolutional neural network, recurrent neural networks, and long short-term memory have previously been used to analyse the Indian stock market prediction. Two different stock market companies namely, New York Stock Exchange and National Stock Exchange were used to compare the above machine learning techniques with numerous existing schemes. In the end, neural network models outperformed the previous existing schemes [18]. The factors that influences a company's stock price are as follows: financial aspects, international policies, emerging news, government policies. These as well affects investors and the stock market. A multi-level machine learning model was proposed by a group of researchers by considering critical technical indicators. The first phase of the model analyses news sentiments using lexicon-based Natural Language Processing (NLP). The prediction of the stock price movement is done by Long Short-Term Memory-Recursive Neural Network (LSTM-RNN) model [19].

In [20], the authors present a new scheme to forecast both multivariate and univariate time series which depends on the combination of some techniques such as clustering, classification and prediction. The proposed algorithm has a frame that is flexible in order to be able to allow any combination of machine learning techniques to generate the model. The new scheme is seen to outperform both classical and machine learning models when compared.

This paper aims to propose a hybrid stock price prediction algorithm by combining standalone deep learning models. Namely, CNN (to capture short-term stock signals), LSTM (to capture long-term stock signals), and Attention Mechanism (to capture features of too long stock prices) to expertly tackle the complication of stock price forecasting in stock markets. Our proposed scheme utilises appropriate standalone models to adequately extract features in order to achieve maximum performance. Frankly speaking finances/resources are never enough for any business person or entrepreneur. A prior knowledge of the future market dynamics of the market of interest is very important. Also, profit maximisation is the key aim of every investor. Therefore, a novel model that can accurately forecast the future market dynamics of any market is the quest of any 21st century investor. The remainder of this work is arranged in the following manner: Section 2 proffers a short write-up of the keep going task on hybrid deep learning-based stock price prediction approaches. Section 3 introduces the framework of the advanced model. Section 4 describes our experiments, as well as evaluation metrics, Section 5 furnishes experimental outcomes and analysis, and Section 6 delivers concluding remarks.

2. Related work

The hybrid deep learning model in [21] presents recurrent networks together with Convolutional Neural Networks (CNNs) to learn the features for the stock price dataset adequately. Also, the authors integrated two Multi-Filter modules to solve the stock price forecasting problem effectively. Nonetheless, it is an obvious fact that CNNs are good at learning short-term dependencies. At the same time, recurrent networks are good at learning long-term dependencies but can not efficiently learn the dependencies of exceedingly long stock price datasets [22].

The paper by [23] introduces an amalgamation of LSTM and Attention Mechanism (AM) to learn the attributes of the stock prices of a dataset. As it is, LSTM can do better in learning long-term dependencies, and Attention Mechanism can do better in learning the dependencies of extraordinarily long stock prices-based time-series datasets [22]. This model leaves out a complete vacuum for the learning of short-term dependencies. Worthy to note is the fact that efficient extraction of attributes is a significant factor for the improvement of prediction accuracies [24].

In [25], the authors proposed a Hybrid Deep Learning architecture that amalgamates CNNs and LSTMs to effectively learn the temporal features for a dataset of stock prices. The above scheme is partitioned into three segments. The first segment concerns the extraction of different time-scales from stock price-based time series via separate layers of CNN also amalgamates them alongside the initial stock prices thus, reflecting changes in the dataset of stock prices, independently. The second portion deals with the utilisation of diversified LSTMs to grasp the dependencies of time for characteristics of distinct time-scales. However, in the terminal segment, what takes place is the amalgamation of all the attributes extracted by LSTMs via a completely linked neural network to estimate the future closing stock prices. Nevertheless, the above scheme can not adequately learn the dependencies of exceptionally long prices of stocks.

In the papers by [26,27], the authors present hybrid deep learning models to adequately perform the tasks of forecasting and/or analysis of time series datasets. Sometimes ago, stand-alone deep learning schemes were used to perform the above tasks while the utilisation of the above proposed algorithm to perform the same tasks yields higher success rate.

In [28], the authors proposed a hybrid network for forecasting the interaction biomolecule kinds for lncRNA (long noncoding RNA). Before applying the proposed model on the biomolecule data, an investigation of the main interaction properties of the molecular mechanisms is carried out on dataset. It is obvious that a well constructed hybrid network can perform the above task and at the same time predict the interaction biomolecule types efficiently.

In the paper by [29], a novel hybrid method for the automated diagnosis of breast cancer is presented by the authors. The above hybrid scheme consists of both machine learning and data mining approaches for the efficient execution of the aforementioned task. Efficient feature extraction is still an issue since appropriate deep learning schemes are not utilised in the ensemble.

The ensemble of Deep Q-Learning agents for predicting stock markets [30] is proposed to minimise some problems like overfitting which are faced by some of the early approaches. This approach employs reinforcement learning schemes that do not utilise annotations to learn features. However, what is learned here is the maximisation of a return function over the training step to achieve maximum success rate.

In [31], the authors proposed a stacked ensemble scheme to perform the task of multi-label classification, a subject that has always attracted attention in various disciplines. The aforementioned scheme was proposed to solve the following two issues to learn the classifier weights needed for the selection of classifiers, and to fully investigate the connection between multi-label performance and pair wise-label interdependence effectively.

In the paper by [32], the authors present a new nested hybrid named ‘‘Ensemble nu-Support Vectors Classification”, for short ‘‘NE-nu-SVC” scheme which blends several orthodox machine learning approaches and ensemble learning methods for efficient diagnosis of CAD. Here, in other to achieve high success rate, firstly, they used a genetic search scheme to choose attributes that are clinically important from the dataset, and secondly balanced the dataset via the multi-level fitting approach.

The paper by [33] introduces a new framework named ‘‘HealthFog” utilised in the analysis of Heart Diseases. For the above scheme, the inputs are heart patient data which is obtain from sensors. The algorithm performs a binary classification task, thus the output or result is whether a patient has heart disease or not. Although deep learning models are used in the ensemble, the issue of efficient attribute extraction is not properly addressed as the appropriate deep learning schemes are not utilised.

In the paper by [34] the authors proposed a novel hybrid model for the analysis of driving performance. It is clear that the level of alertness and attentiveness of drivers while driving are key factors worth considering in trying to minimise the number of road accidents. The above ensemble consists of statistical and machine learning schemes but no deep learning models.

In [35] a novel algorithm for the classification of text of inclusive policy was presented. In its first stage an ensemble convolutional neural network is used for efficient extraction of short-term features while the adequate extraction of long-term attributes remains unaddressed.

Nonetheless, our proposed scheme provides solutions to all the issues above to adequately forecast the prices of stocks. It is made up of CNN modules to effectively learn short-term dependencies and LSTM modules to effectively learn long-term dependencies. We as also introduce the Attention Mechanism to learn the dependencies of exceedingly long stock price datasets adequately.

3. Proposed model

Understandably, none of the proposed hybrid deep learning networks for the stock price forecasting problem can fully capture the following three features together: long-term features, short-term features, and features of very long stock prices. Therefore, this work projects a novel hybrid deep-learning algorithm to capture the above three features together efficiently. The above task is accomplished by combining CNN (captures short-term stock signals), LSTM (captures long-term stock signals), and Attention Mechanism (captures signals from too long stock price data) to address the stock price forecasting problem. Our proposed model is named "An Ensemble of a Boosted Hybrid of Deep Learning Models and Technical Analysis for Forecasting Stock Prices," in short, "EHTS.".

Our proposed model employs two networks, namely Attention-Based CNN (AB-CNN), and Contextual Bidirectional LSTM (CB-LSTM), to do the feature extraction task. Stock price data is simultaneously loaded into the AB-CNN and CB-LSTM arms. The input to the AB-CNN module is of univariate type stock price data with multiple timestamps, while the one for the CB-LSTM arm is multivariate with one timestamp.

Algorithm 1: Pseudocode for the proposed EHTS model

Input: Raw datasets of stocks
Output: Predict the prices of stocks and the accuracies achieved
1: Setup timestamps for each of the two feature extraction arms, i.e. AB-CNN and CB-LSTM.
2: Simultaneously feed data into the AB-CNN and CB-LSTM arms.
3: In the AB-CNN arm, Concat1 is calculated using Eq. 1.
4: The result of Concat1 is feed into the attention mechanism.
5: Concatenate the outputs of AB-CNN and CB-LSTM by using Eq. 6.
6: Feed the output of step 5 into the MLP stage.
7: Adjust the number of epochs and batch sizes.
8: Train model.
9: Forecast/Predict stock prices and give the accuracies obtained.

Deep Learning models are presently the benchmark methods for most of the problems in computer vision, natural language processing (NLP), and speech recognition [36], for extracting features. The CNN introduced in [37] has demonstrated marvelous achievements atop benchmark models. A unique kind of CNN has been successfully implemented in computer vision [38] and speech recognition [39], and has demonstrated robustness against noise in speech data than other deep learning models.

In recent times, the application of attention-based RNNs [40] to a variety of disciples like speech recognition, image caption generation, handwriting generation, and machine translation was actualised. In the area of NLP researchers have begun applying attention mechanism to CNNs, which appears to be meaningful when the input feature sequences are very complicated or too long [41].

This arm, i.e. the AB-CNN arm of our model, is introduced to capture short-term feature maps and feature maps of very long sequences of stocks. In the AB-CNN module, the CNN layers are connected in parallel. Let $C_1, C_2, C_3, \cdots, C_p$ be the outputs of the CNN layer1, layer2, layer3, ……., layerp respectively. Let C_T be the total output of all the CNN layers which can be evaluated as follows:

$$\frac{1}{C_T} = \frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} +, \cdots, + \frac{1}{C_p}. \tag{1}$$

Thus, C_T is the final output of "Concat1" and is downsampled by either Average or Maximum pooling technique before feeding into the Attention Mechanism. Let α denote a series of convolutional feature maps possessing vectors α_t, and let β_t denote the amplitude of the contribution made by feature maps towards the forecasting problem [40]. The above contributions are usually called feature map attention weight and are evaluated as follows [40]:

$$\beta_t = \frac{exp(\delta(\alpha_t))}{\sum_m exp(\delta(\alpha_m))}, \tag{2}$$

where $\delta(\alpha)$ denotes a scoring function, we now let $\lambda(\alpha)$ to denotes the out-run of the attention layer. Nonetheless, $\lambda(\alpha)$ which is the weighted sum of the input feature maps sequences, is evaluated as follows [40]:

$$\lambda(\alpha) = \sum_t \beta_t \alpha_t. \tag{3}$$

Intuitively, AB-CNN is introduced to adequately handle the following issues: extract short-term features; extract features from long sequences of stocks, and lastly to extract features from complex stock datasets.

The second arm of our proposed model is named CB-LSTM and its inputs are stock market datasets. RNN has produced excellent results in numerous disciplines. Namely: sentiment analysis; handwriting recognition, and speech recognition [39]. Moreover LSTM based models have been seen to outperform RNNs on various tasks such as context-free learning and contextually sensitive languages. Also, Bidirectional LSTM (BLSTM) networks, i.e. networks that produce decisions by using both directions of the input sequence, have been presented to forecast three features namely wind speed, load demand, and electricity price using the Ontario province dataset [42].

In this CB-LSTM arm of our proposed model, we let $x = \{x_1, x_2, \cdots\cdots, x_{n-1}, x_n\}$ be a sequence of stock market data and take $s = \{s_1, s_2, \cdots\cdots, s_{n-1}, s_n\}$ to be the bootsrap stock market dataset. Let $M = \{M_1, M_2, \cdots\cdots, M_{n-1}, M_n\}$ be the bootstrap means of stock market datasets. CB-LSTM considers the bootstrap means (M) of the stock market dataset as the contextual features. The CB-LSTM is very similar to the ordinary BLSTM, with some changes effected on the equations that mimic LSTM cell operations. We add the bootstrap means (M) to the following gates: input; forget, cell; and output. In the following equations, the vivid black terms ($\mathbf{U_{Mi}}M$) were the alterations effected on the ordinary LSTM equation [43].

$$\left.\begin{aligned} i_p &= \Psi(U_{xi}x_p + U_{hi}h_{p-1} + U_{ci}c_{p-1} + b_i + \mathbf{U_{Mi}}M), \\ f_p &= \Psi(U_{xf}x_p + U_{hf}h_{p-1} + U_{cf}c_{p-1} + b_f + \mathbf{U_{Mi}}M), \\ c_p &= f_p c_{p-1} + i_p \tanh\left(U_{xc}x_p + U_{hc}h_{p-1} + b_c + \mathbf{U_{Mi}}M\right), \\ o_p &= \Psi(U_{xo}x_p + U_{ho}h_{p-1} + U_{co}c_p + b_o + \mathbf{U_{Mi}}M), \\ h_p &= o_p \tanh(c_p), \end{aligned}\right\} \tag{4}$$

where i_p, f_p, o_p denote, input, forget, and output gates at time p respectively. x_p denotes input vector at time p. c_p and c_{p-1} represent the latest and old stock information, respectively. h_{p-1} denotes latent state at the previous timestep. U_{xi}, U_{xf}, U_{xc}, and U_{xo} are weighted matrices associated with the input vectors x_p. U_{hi}, U_{hf}, U_{hc}, and U_{ho} are weighted matrices associated with the hidden state (i.e., h_{p-1}) at the previous timestamp. U_{ci} and U_{cf} are weighted matrices associated with old stock price information (i.e. c_{p-1}). U_{co} is a weighted matrix associated with the latest stock price information (i.e., c_p). $\mathbf{M}$ represent vectors of bootstrap means, and $\mathbf{U_{Mi}}$ are the corresponding weighted matrices. h_p denotes either a forward or backward final result for the CB-LSTM module. Ψ denotes the activation function of hidden layers. The term $\mathbf{U_{Mi}}M$ is our context in the CB-LSTM module.

However, in CB-LSTM, the extraction of both short and long term features is effectively achieved. The CB-LSTM architecture comprises two sets of output layers: forward final output layer ($\overrightarrow{h_p}$) and backward final output layer ($\overleftarrow{h_p}$). The forward final output layer is the first to be evaluated, followed by the backward final output layer. And then combine them to yield the output y_p as follows [43]:

$$y_p = U_{\overrightarrow{h_p}\,y}\overrightarrow{h_p} + U_{\overleftarrow{h_p}\,y}\overleftarrow{h_p} + b_{\overrightarrow{h_p}} + b_{\overleftarrow{h_p}}, \tag{5}$$

where $U_{\overrightarrow{h_p}\,y}$ and $U_{\overleftarrow{h_p}\,y}$ are weighted matrices associated with the forward and backward final output layers, respectively. $b_{\overrightarrow{h_p}}$ and $b_{\overleftarrow{h_p}}$ denote the bias vectors for the forward and backward final output layers, respectively.

The next stage is the concatenation process (i.e., Concat2). In this stage, the final outputs of the AB-CNN and CB-LSTM arms of the feature extraction process are merged. Mathematically, "Concat2" can be expressed as follows [43]:

$$\begin{aligned} \text{Concatenation} &= \text{merge}(\lambda(\alpha), y_p) \\ &= \Omega\left(U^{\lambda} \cdot \lambda(\alpha) + b^{\lambda} + U^{y} \cdot y_p + b^{y}\right), \end{aligned} \tag{6}$$

where $\Omega(\cdot)$ represents element-wise ReLU activation function, + indicates element-wise addition. $\lambda(\alpha)$ is the final output of the AB-CNN block, while y_p is the final output of the CB-LSTM arm. U^{λ} and b^{λ} are respectively the weighted matrices and bias vectors from the AB-CNN training while U^{y} and b^{y} are also weighted matrices and bias vectors, respectively, obtained from the CB-LSTM training.

The last stage of our proposed model is the multilayer perceptron (MLP). MLP is an example of a simple neural network also called Feed Forward Network with input-neurons being associated with their superseding hidden layer neurons by a matrix of constituents weights. Three layers make up an MLP: input layers, hidden layers and output layers. The neurons within a layer of the MLP do not interconnect, yet they link with neurons of the afterward layer. In this paper, the MLP stage input is the concatenation stage output. The problem setting requires the placing of a batch normalisation layer after the first MLP layer. The predictions of the last layer (i.e., output layer) of the MLP can be mathematically expressed as follows [43]:

$$\begin{aligned}\text{Predictions} &= \Theta(\lambda(\alpha), y_p)\\ &= U^{m}\cdot\Omega\left(U^{\lambda}\cdot\lambda(\alpha)+b^{\lambda}+U^{y}\cdot y_p+b^{y}\right)+b^{m},\end{aligned} \tag{7}$$

where U^{m} denotes the weighted matrices associated with the final output of MLP and b^{m} is a corresponding bias vector term.

4. Experimental results

4.1. Dataset and evaluation metrics

In this paper, we used sure-enough stock market datasets collected from a prominent stock code-named "Johnson & Johnson," in short "JNJ," a leading stock in a well-known stock market called NYSE. The dataset used in this experiment is freely acquired from the yahoo finance website.

For this paper, the entire features of our dataset which include the following; Open price (Open), High price (High), Low price (Low), Closing price (Closing), Adjusted Close price (Adj Close) and Volume were utilised. Owing to immensely high values we standardised our dataset to lie between the range 0 and 1 inclusively. We denote our attribute vector as P_i, and thus expressed it as $P_i = \{p_1, p_2, p_3, p_4, p_5, p_6\}$, where i is a natural number that lies in the range $1 \leqslant i \geqslant 6$, while p_i serves as a feature column vector. Consequently, the standardized value $(\Pi_{i,j})$ of an information instance in our dataset is given as follows [21].

$$\Pi_{i,j} = \frac{P_i - \text{mean}(P_i)}{\text{std}(P_i)}, \tag{8}$$

where j denotes the j-th element of the i-th feature vector, mean(P_i) as well as std(P_i) respectively represents the average and standard deviation of an attribute column vector.

We used four metrics to evaluate our proposed scheme's performance adequately. These metrics appear as follows: Root Mean Square Error (rMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) [44], along with Time Complexity [22]. The rMSE metric is given by [44]:

$$\text{rMSE} = \sqrt{\frac{1}{N}\sum_{r=1}^{N}(\xi_r - \kappa_r)^2}, \tag{9}$$

Likewise, the MAE metric is evaluated by utilising the following formula [44]:

$$\text{MAE} = \sqrt{\frac{1}{N}\sum_{r=1}^{N}|\xi_r - \kappa_r|}, \tag{10}$$

The MAPE, is obtained by using the expression [44]:

$$\text{MAPE} = \frac{1}{N}\sum_{r=1}^{N}|\frac{\xi_r - \kappa_r}{\xi_r}|. \tag{11}$$

wherein ξ_r and κ_r stands for the real as well as predicted values respectively for information sample r.

The time complexity metric measures the length of time taken to get a quick solution for a particular problem.

We further examine the effectiveness of the proposed model by using a rank based statistics named Kruskal–Wallis Test [45]. The Null and Alternative hypothesis of the Kruskal–Wallis Test are stated as follows [45]:

$$H_0: \textit{The probability distributions of EHTS and a baseline scheme are the same}$$

$$H_1: \textit{The probability distributions of EHTS and a baseline scheme are not the same}$$

We experiment different possible arrangements of CNN layers and Attention Mechanism. The arrangement that yields the lowest training and testing errors was selected. For the CB-LSTM arm, we also experiment different parameters to select the arrangement that yields the lowest training and testing errors. Also, parameters of an arrangement that gives the lowest training and testing errors are optimised. The optimisation of the proposed method takes on average 30 and 32 s respectively to train and predict a single feature of the stock data.

4.2. Experimental setup

For the settings of this work, we choose both regression and deep learning algorithms to serve as baselines. We compare the performance of our projected model with eleven benchmark models. The eleven baseline models considered are as follows: Predicting the Trend of Stock Market Index Using the Hybrid Neural Network Based on Multiple Time Scale Feature Learning (PBMT) [25], DGHNL: A new deep genetic hierarchical network of learners for prediction of credit scoring (DGHNL)[46], ResNet-Attention model for human authentication using ECG signals (RAHA) [47], Novel deep genetic ensemble of classifiers for arrhythmia detection using ECG signals (NGEC)[48], RNN, CNN, LSTM, SVM, Logistic Regression (Logistic Regr), Random Forest (Random Fore) and Linear Regression (Linear Regr).

The EHTS algorithm was trained through the well-known Adam optimiser. The Adam optimiser utilises the following settings: 0.003 used as learning rate; 1e-6 used as learning rate decay; 0.9 utilised as momentum and the other parameters retained their conventional values for training the EHTS model. The loss functions of our proposed scheme include; rMSE, MAE and MAPE.

Implementation of this project was done via the python programming language by utilising the jupyter notebook environment. We use the following libraries: keras, numpy, pandas, and matplotlib during implementation. Our proposed model seems to overfit easily which is a significant setback encountered not only by our scheme but by all Neural Network-based models. This setback is cause by exterior parameters. For every 140 epochs without a change in the validation scores we reduce the learning rate by a fraction of 0.7 until the appropriate learning rate is attained. We evaluate our proposed model by observing the performance of the models and select the model which acquires the minimal training error. For the AB-CNN arm we utilised 3 CNN layers with ten neurons each. Dropout layers are used after every BatchNormalisation of a CNN layer to overcome overfitting.

Our Attention Mechanism contains one layer that accepts only 8 neurons. The CB-LSTM arm is made up of 3 layers: the first layer possesses 10 neurons; the second layer has 30 neurons and the third layer uses 20 neurons. For us to overcome overfitting and salvage computational resources [49] we apply dropout layers after each of the last two layers of the CB-LSTM module. The predicting block comprises of 4 layers. The last layer is the output layer and responsible for making predictions. The first three layers possesses 20, 30 and 40 neurons respectively.

Furthermore, we investigate our proposed algorithm's performance by using specific number of CNN layers and four optimisers. The optimisers considered are as follows: Adam, Stochastic Gradient Descent (SGD), Adaptive Gradient (AdaGrad) and Root Mean Square Propagation (RMSprop).

4.3. Stock simulation

We further attempt to study our stock's dataset to determine trading possibilities through the application of the indicators of Technical Analysis (TA). TA is a trading concept that is routinely employed to achieve short-range market signals. TA indicators are used to evaluate the weakness and strength of a stock market. Thus, we use the following technical analysis indicators: Moving Average (MA), Moving Average Convergence-Divergence (MACD) and Relative Strength Index (RSI) [50] to evaluate the weakness and strength of our stock market.

4.3.1. Moving average (MA)

MA can be defined as the average price of a stock for a definite period. However, three of the most utilised moving averages are: simple MA, weighted MA and exponential MA. In this paper, we utilised both the MA and exponential MA (EMA). The latter (i.e. EMA) is mathematically expressed as [50]:

$$\mathrm{EMA}_i = P_i * w + \mathrm{EMA}_{i-1} * (1 - w), \quad w = \frac{2}{N+1} \tag{12}$$

where P_i represents the prices of stocks for period i, w denotes the smoothing factor and N represents the number of periods in the EMA.

4.3.2. Moving average convergence-divergence (MACD)

The Moving Average Convergence-Divergence is one of the most important indicator of TA which assesses the variance of the short-term (s), long-term (l) and medium-term (m) EMAs for the closing prices of stocks. The MACD indicator is given by [50]:

$$\mathrm{MACD}_i = \mathrm{EMA}_{s,i} - \mathrm{EMA}_{(l,m),i} \tag{13}$$

where $MACD_i$ denotes the MACD of stock at point $i, EMA_{s,i}$ denotes the short-term EMA of stock at point i and $EMA_{(l,m),i}$ denotes the EMA of either the medium or long-term closing prices of stock at point *i*.

4.3.3. Relative strength index (RSI)

This indicator estimates the span of oscillations for the prices of stocks in the stock markets and is given by [50]:

$$\mathrm{RSI}_i = 100 - \frac{100}{1 + \mathrm{RS}_i}, \tag{14}$$

where [50]:

$$\mathrm{RS}_i = \frac{\sum_{m=i}^{i-d} \max\left(0, P_m - P_{m-1}\right)}{\sum_{m=i}^{i-d} \left| \min\left(0, P_m - P_{m-1}\right)\right|}. \tag{15}$$

where RS_i denotes the average gain or loss of a stock at point i, P_{m-1} and P_m represent the closing buying and selling prices of a stock at point *i*, respectively.

4.3.4. Volume

The volume of a stock is defined as the number of traded shares in a stock per day or period [50].

4.3.5. Returns, Annual Returns, Sharpe Ratio, Volatility, Maximum Drawdown, Daily WinningRate

We also investigated the performance of the EHTS model by performing forecast-based trading to assess whether the forecasts done by the proposed model yield profit or not. The following metrics: Return (R), Annual Return (AR), Sharpe Ratio (SR), Volatility (V), Maximum drawdown (MDD) and Daily winning rate (DWR) are employed to accomplish the above task. The accuracy of schemes will only evaluate the capability of the classification-based -forecasts that corresponds to the span of future returns. In stock market practices the profit that is connected to the estimated fall or rise is very important.

In our settings, the total returns (R) of a stock is the predicted returns values over the entire period of simulation. Total returns is given as follows [21]:

$$R = \frac{(P_f - P_i) + r_t}{P_i}, \tag{16}$$

where P_i and P_f represent the initial and final closing prices from the begin to the end of the simulation respectively.

$$r_t = \ln\left(\frac{\mathrm{Closing}_t + t_{\mathrm{forward}}}{\mathrm{Closing}_t}\right), \tag{17}$$

where r_t represents the logarithmic returns of the stock prices in t_{forward} minutes. For example: $t_{\mathrm{forward}} = 5$ implies a model will predict future-5-min returns for the stock under consideration. A prediction with high returns value means the model has outstanding performance in terms of profitability.

The annual return ratio (AR) is evaluated by transforming the total profit as follows [21]:

$$AR = (1 + R)^{\frac{244}{T_s}} - 1, \tag{18}$$

where T_s denotes the time used to simulate the scheme and 244 is the mean number of business days per year.

Furthermore, we used the daily winning rate (DWR) [21] metric to assess model stability on forecast-based businesses of the stock market. In portfolio theory the risk-adjusted profit is always used to assess the stability of the business systems. The sharpe ratio (SR) is another metric that has been widely used to assess algorithm performances in various business related tasks. The SR metric is given by [21]:

$$SR = \sqrt{244} \times \frac{\overline{r_e}}{\sigma_e}, \tag{19}$$

where

$$\overline{r_e} = \frac{1}{m}\sum_{j=1}^{m}\left(r_p^j - r_f^j\right), \tag{20}$$

and

$$\sigma_e = \sqrt{\frac{1}{m-1}\sum_{j=1}^{m}\left(r_p^j - r_f^j - \overline{r_e}\right)}, \tag{21}$$

where $\overline{r_e}$ denotes the mean daily excess returns/income during the period of simulation and σ_e represents the volatility of every-day excess returns for a given period of the simulation. r_p^j and r_f^j represent the income from the business strategy and the risk-free tax of interest on the jth business day respectively. m denotes the time in days of the simulation period. High SR value is an indication of high profit under a component risk and soaring stability. Another metric that is used to assess the risk of business systems is the maximum drawdown (MDD). MDD measures the way a business account is drawn down from the peak to the trough before attaining another peak. Annual volatility (V) is another metric that is used to assess risk and is evaluated as follows [21]:

$$V = \sqrt{\frac{244}{m-1}\sum_{j=1}^{m}\left(r_p^j - \overline{r_p}\right)^2}, \tag{22}$$

where $\overline{r_p}$ represents the mean returns of the business system.

5. Discussion

Fig. 1 and Algorithm 1 present the general overview of the EHTS scheme. Tables 1 and 2 give a summary of the performance of the proposed and the eleven baseline models on the JNJ dataset using the train-test-split and cross validation methods respectively. We observe that the proposed and all eleven benchmark schemes perform better under the train-test-split approach (i.e. in Table 1) than the cross validation method (i.e. in Table 2). Thus the motivation for using the train-test-split method in our experiments (Table 4).

In Table 3 a summary of the hyperparameters with their possible ranges for the various stages of the EHTS model are presented.

Table 3 gives the results of hyperparameter optimisation for our proposed scheme via the rMSE metric. For every hyperparameter the range that offers the lowest training and testing errors is selected for the model setup. Thus, the bold black decimal numbers in the table corresponds to the selected hyperparameters.

This research commences by exploring the optimiser type and number of CNN layers that are worthy for our experiment. For all the four optimisers, we examine three CNN layers combination: two layers, three layers and four layers in the AB – CNN arm. We use errors of training and testing as metrics and report on the model that gives the minimum for the two errors. In Table 5 it is seen that the Adam optimiser that has 3 CNN layers in the AB-CNN module gives the lowest training and testing errors 0.047 and 0.039 respectively. Thus the motivation for selecting the Adam optimiser for our model setup.

In the areas of machine and deep learning datasets are divided into two parts: training and testing before feeding them into the algorithm. If an algorithm utilises 80% of its dataset for training then the remaining 20% will be used for testing. In Table 6, we made a comparison of the performances of the proposed and the eleven benchmark models on five training win-

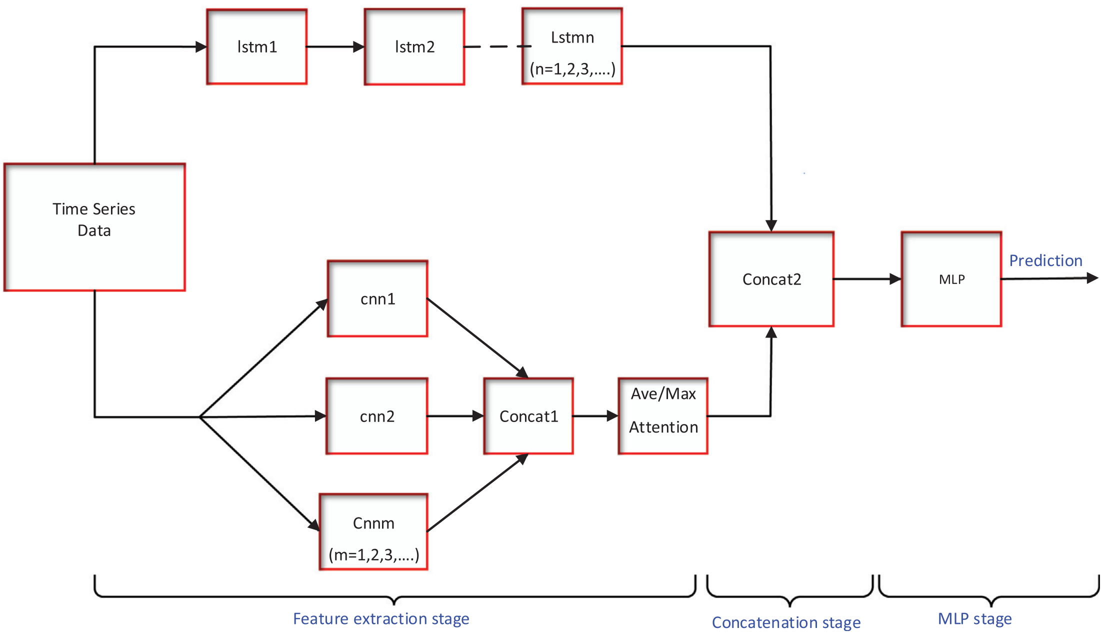

Fig. 1. The architecture for EHTS neural network.

Table 1
Training errors for the JNJ dataset using the train-test-split method.

Models	rMSE	MAE	MAPE
EHTS	**0.039**	**0.404**	**0.694**
PBMT	0.056	0.487	1.001
DGHNL	0.100	0.527	1.282
RAHA	0.104	0.529	1.321
NGEC	0.114	0.530	1.385
RNN	0.128	0.532	1.401
CNN	0.139	0.551	1.550
LSTM	0.111	0.519	1.301
SVM	0.147	0.764	1.771
Logistic Regr	0.107	0.614	2.101
Random Fore	0.142	0.871	1.802
Linear Regr	0.153	1.018	1.922

Table 2
Training errors for the JNJ dataset using cross validation method.

Models	rMSE	MAE	MAPE
EHTS	**0.041**	**0.516**	**0.722**
PBMT	0.062	0.501	1.042
DGHNL	0.104	0.548	1.390
RAHA	0.118	0.552	1.354
NGEC	0.135	0.541	1.404
RNN	0.139	0.546	1.441
CNN	0.152	0.610	1.571
LSTM	0.132	0.542	1.322
SVM	0.162	0.773	1.787
Logistic Regr	0.109	0.644	2.142
Random Fore	0.241	0.884	1.832
Linear Regr	0.174	1.029	1.962

Table 3
Description of hyperparameters and their allowed ranges for AB-CNN only scheme, CB-LSTM only scheme, and EHTS scheme using the JNJ dataset are given below.

Networks	Hyper parameters	Range	CNN	LSTM	CNN-LSTM (SMF)
	No. of layers	1–3	–	3	2
LSTM	Uni/bi-directional	{uni,bi}	–	uni	uni
	Apply contextual features	{yes,no}	no	yes	yes
	Name of contextual features	–	–	bootstrap means	bootstrap means
	Filter shape or Kernel size	3–4	3	–	3/4
	No. of hidden layers	1–4	4	–	5
	No. of layers	3–6	6	–	4
CNN	Downsampling technique	{Maximum, Average}	Average	–	Maximum
	No. of Concatenations	0–2	1	–	2
	Apply attention mechanism	{yes,no}	yes	no	yes
	Concatenation strategy	{flattening, broadcast}	flattening	–	flattening
	Concatenation strategy	{flattening, broadcast}	broadcast	–	broadcast
MLP	No. of layers including the output layer	1–3	2	2	3
	No. of units in first layer	1–510	421	340	504

dow dimensions using the JNJ dataset. Results from Table 6 revealed that the EHTS model outperforms all benchmark schemes in all the five different training window sizes. The 70% training window size gives the best performance. Hence the reason for using the 70% training window size in our experiments.

Table 7 presents results of the performance of EHTS and benchmark models on the BAC dataset using different training window sizes. However, it is seen that the proposed model outperforms the baseline schemes.

In this experiment we randomly selected five t_{forward} values: 7, 14, 21, 28, 35 and select the values that yield outstanding model performance. In Table 8 we observe that all hybrid deep learning -based model simulations are more profitable than the standalone deep and machine learning models. This means that hybrid deep learning models can capture suitable business features that leads to profit realisation. All simulations that are forecast-based hybrid deep learning schemes yield superior returns and annual returns than the standalone machine learning models.

Table 4
Results for the optimisation of hyperparameters of our proposed model via the rMSE metric using the JNJ dataset.

Hyper parameters	Range	Training errors	Testing errors
Downsampling Technique	Maximum	**0.038**	**0.034**
	Average	0.044	0.047
CNN layers arrangement	parallel	**0.039**	**0.038**
	series	0.045	0.047
CNN	with Attention	**0.040**	**0.039**
	no Attention	0.049	0.047
LSTM direction	uni	**0.037**	**0.032**
	bi	0.042	0.045
Concatenation	broadcast	0.045	0.048
	flattening	**0.036**	**0.031**
LSTM	with contextual features	**0.035**	**0.033**
	no contextual features	0.048	0.041

Table 5
Performance of EHTS for different optimisers and a specific number of CNN layers in the AB-CNN arm, using the JNJ dataset.

Optimisers	#CNN layers	rMSE	
		Training	Testing
Adam	2	0.087	0.065
	3	**0.041**	**0.039**
	4	0.074	0.091
SGD	2	0.056	0.068
	3	0.084	0.078
	4	0.069	0.067
AdaGrad	2	0.110	0.090
	3	0.092	0.088
	4	0.077	0.069
RMSprop	2	0.102	0.100
	3	0.107	0.099
	4	0.084	0.098

Table 6
Performance of EHTS and eleven baseline models on the JNJ dataset for different percentage training window sizes.

Models	40%	50%	60%	70%	80%
EHTS	**85.04**%	**86.11**%	**88.24**%	**94.12**%	**87.41**%
PBMT	84.12%	82.47%	87.03%	92.22%	85.71%
DGHNL	79.01%	78.02%	86.04%	89.41%	79.02%
RAHA	78.10%	80.51%	86.78%	89.24%	79.77%
NGEC	78.79%	81.30%	87.85%	89.97%	78.00%
RNN	77.10%	79.42%	87.46%	88.52%	78.11%
CNN	76.51%	78.56%	85.50%	89.32%	77.71%
LSTM	75.04%	76.28%	85.71%	87.05%	78.01%
SVM	73.21%	74.81%	86.80%	85.16%	76.42%
Logistic Regr	72.42%	72.02%	87.01%	84.42%	75.51%
Random Fore	70.07%	71.44%	86.10%	83.51%	74.61%
Linear Regr	69.09%	71.54%	84.40%	83.80%	74.16%

We also observed that the logistic regression shows outstanding performance with regards the risk control metric with an annual volatility of 4.71%. The second-best scheme is the NGEC model. It is seen that simulations that are based on logistic regression yield lower MDD values than the rest. It is a conservative strategy that might result in secondary returns.

For stability, we see that the most stable results are from PBMT with a Sharpe Ratio (SR) and daily winning rate (DWR) of 8.31% and 77.34% respectively. The EHTS model produces the second-best result. Therefore, from the results in Table 8 we can authoritatively conclude that hybrid deep learning models possess the capability to capture both profitable and stable signals than the rest. Our proposed model (i.e.,EHTS) is seen to be the second-best.

Table 7
Performance of EHTS and eleven baseline models on the BAC dataset for different percentage training window sizes.

Models	40%	50%	60%	70%	80%
EHTS	**89.02**%	**90.10**%	**92.21**%	**98.13**%	**91.39**%
PBMT	87.15%	86.44%	91.06%	96.31%	89.61%
DGHNL	83.54%	84.09%	90.10%	93.21%	85.01%
RAHA	82.20%	84.42%	90.81%	93.02%	85.80%
NGEC	82.80%	85.13%	91.90%	93.87%	84.04%
RNN	81.14%	84.52%	91.62%	91.46%	84.22%
CNN	80.54%	83.60%	88.62%	92.40%	83.61%
LSTM	79.12%	81.24%	88.89%	90.07%	84.10%
SVM	76.18%	78.49%	89.90%	88.21%	82.51%
Logistic Regr	76.52%	76.07%	90.08%	87.32%	81.64%
Random Fore	74.11%	75.38%	89.20%	86.70%	80.63%
Linear Regr	73.04%	75.61%	87.30%	86.91%	80.24%

Table 8
Stock price simulation results for the JNJ dataset.

Models	$t_{forward}$	R	AR	SR	V	MDD	DWR
EHTS	7	**30.40**%	**55.32**%	6.12	10.42%	−4.42%	74.22%
PBMT	7	25.08%	43.20%	**8.31**	7.33%	−3.02%	**77.34**%
DGHNL	7	21.11%	38.09%	4.04	10.21%	−4.24%	61.78%
RAHA	7	19.12%	24.07%	5.04	12.10%	−6.11%	70.04%
NGEC	7	18.41%	20.72%	4.18	4.72%	−2.23%	69.71%
RNN	14	17.01%	19.60%	3.31	10.14%	−5.47%	59.11%
CNN	14	16.24%	18.06%	3.10	8.21%	−4.70%	55.32%
LSTM	14	15.80%	16.89%	1.89	11.28%	−3.91%	51.49%
SVM	21	16.08%	17.23%	2.18	10.04%	−5.67%	46.04%
Logistic Regr	14	16.10%	18.29%	1.81	**4.71**%	−**2.17**%	60.13
Random Fore	21	12.44%	14.41%	1.52	6.06	−1.98%	40.77%
Linear Regr	14	10.11%	11.81%	1.15	8.49%	−3.77%	38.05%

Moreover, we assess the performance of our projected scheme via the three metrics: rMSE, MAE and MAPE. Results from Table 9 revealed that the EHTS scheme outperforms all the benchmark schemes on the JNJ dataset. PBMT, DGHNL and RAHA are appraised number-two, number-three and number-four respectively in performance. Thus, hybrid deep learning models are seen to outperform the other schemes entirely. The standalone LSTM scheme is seen to be number-five in terms of performance. Also, Table 10 gives results of the performance of EHTS and baseline schemes on the BAC dataset. It is observed that our proposed model performs better than benchmark schemes for all the three metrics.

Furthermore, we investigate the six attributes in our datasets via the following statistics: Minimum (MIN), Maximum (MAX) and two measures of central tendency (i.e, MEAN, and MEDIAN). Results from Table 11 disclosed that the Volume attribute gives the maximum value thus, excluded from the subsequent analysis. In Table 11 we also observed that for the MIN statistic, Adj Close attribute yields the lowest price (85.94) whilst the High attribute yields the maximum price (96.64). For the MAX statistic, we observed that the Adj Close feature produces the minimum price (146.44) whilst the High feature produces the maximum price (149.00). And for the MEAN statistic we observed that the Adj Close feature yields the

Table 9
Performance comparison of EHTS with eleven benchmark models on the JNJ dataset in terms of training errors via three metrics.

Models	rMSE	MAE	MAPE
EHTS	**0.039**	**0.404**	**0.921**
PBMT	0.056	0.487	1.001
DGHNL	0.100	0.527	1.282
RAHA	0.104	0.529	1.321
NGEC	0.114	0.530	1.385
RNN	0.128	0.532	1.401
CNN	0.139	0.551	1.550
LSTM	0.111	0.519	1.301
SVM	0.147	0.764	1.771
Logistic Regr	0.107	0.614	2.101
Random Fore	0.142	0.871	1.802
Linear Regr	0.153	1.018	1.922

Table 10
Performance comparison of EHTS with eleven benchmark models on the BAC dataset in terms of training errors via three metrics.

Models	rMSE	MAE	MAPE
EHTS	**0.029**	**0.394**	**0.911**
PBMT	0.045	0.476	0.991
DGHNL	0.090	0.518	1.271
RAHA	0.095	0.519	1.310
NGEC	0.103	0.520	1.374
RNN	0.117	0.523	1.391
CNN	0.129	0.540	1.540
LSTM	0.101	0.517	1.291
SVM	0.138	0.752	1.762
Logistic Regr	0.096	0.604	2.089
Random Fore	0.133	0.860	1.790
Linear Regr	0.143	1.006	1.913

Table 11
Feature assessment on the JNJ dataset via four statistics; minimum, maximum and two measures of central tendency.

Features	MIN	MAX	MEAN	MEDIAN
Open	95.77	147.84	127.81	129.90
High	96.64	149.00	128.63	130.91
Low	94.28	147.00	126.96	128.80
Close	95.75	148.14	127.83	129.90
Adj Close	85.94	146.44	121.33	124.18
Volume	2469500	58140200	7040843	6399000

minimum price (121.33) whilst the maximum price is observed for the High feature. Ultimately, it is seen under the MEDIAN statistic that the High attribute gives the maximum price (130.91) and the Adj Close feature yields the least price (124.18).

Table 12 presents the mean training time per epoch and the number of epochs required by our proposed scheme against 4 benchmark models using JNJ data. The outcomes from Table 12 disclosed that our projected model needs 31 s for a single epoch's training but requires a minimal number of epochs (98). Thus, the CNN model requires the minimum training time for one epoch (11 s) and needs 100 epochs. In Table 13, we observe that the proposed scheme has a mean training time of 33s with the lowest number of epochs (96) using the BAC data. The CNN model is seen to have the lowest mean training time (10 s).

Table 14 presents the sums of ranks (SR) of four baseline and EHTS model. In Table 14, the H-value of the Kruskal–Wallis Test far exceeds that of the 5% confidence level which means that we reject H_0 and accept H_1. The interpretation of the above outcome is that the schemes are taken from different populations. A scheme with low SR indicates excellent performance and from the table it is seen that our proposed scheme (i.e., EHTS) which has a SR of 1392 outperforms the four baseline schemes. The NGEC model has the highest SR value which implies that it has the worst performance.

In Table 15, EHTS model gives a SR value of 1392, the lowest in the table. This implies the EHTS scheme outperforms all the five baselines on the BAC dataset.

Fig. 1 presents an ensemble of deep learning models which can efficiently capture both profitable and stable trading signals to address the stock price prediction problem.

Table 12
Comparative results for EHTS against seven benchmark models on the JNJ dataset via mean training time for each epoch as well as the number of epochs.

Scheme	*Training time(s)*	*#Epochs*
EHTS	31	**98**
PBMT	38	135
DGHNL	19	124
RAHA	29	210
NGEC	40	190
RNN	16	110
CNN	**11**	100
LSTM	18	140

Table 13
Comparative results for EHTS against seven benchmark models on the BAC dataset via mean training time for each epoch as well as the number of epochs.

Scheme	*Training time(s)*	*#Epochs*
EHTS	33	**96**
PBMT	39	138
DGHNL	21	125
RAHA	42	218
NGEC	15	197
RNN	18	116
CNN	**10**	109
LSTM	20	151

Table 14
The sums of ranks of EHTS and four baselines along with the Kruskal–Wallis Test H-value for the JNJ dataset.

SR_{EHTS}	SR_{PBMT}	SR_{DGHNL}	SR_{RAHA}	SR_{NGEC}	*H*-value
1392	1474	1495	1548	1587	7.132

Table 15
The sums of ranks of EHTS and four baselines along with the Kruskal–Wallis Test H-value for the BAC dataset.

SR_{EHTS}	SR_{PBMT}	SR_{DGHNL}	SR_{RAHA}	SR_{NGEC}	*H*-value
1541	1984	1998	2152	2461	7.341

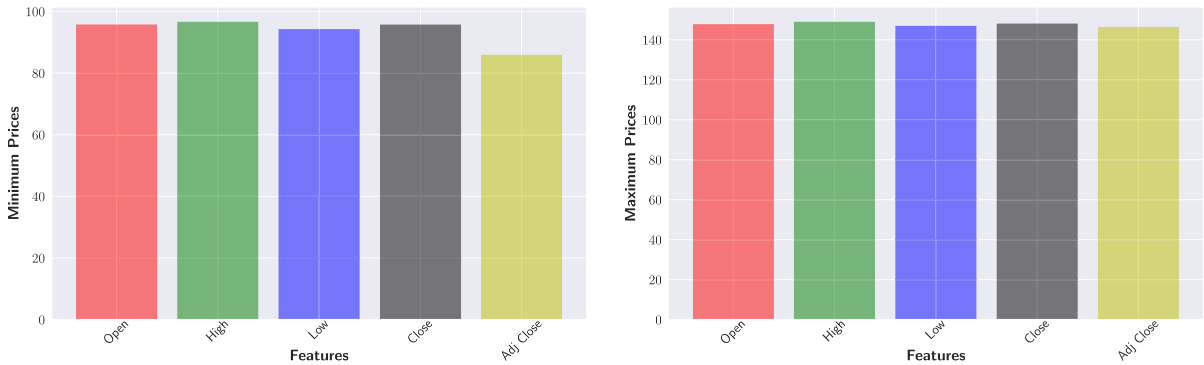

Fig. 2. (a) Bar Chart showing minimum prices for 5 of our features in the JNJ dataset; (b) Bar Chart showing maximum prices for 5 of our features in the JNJ dataset.

SubFig. 2(a) is a bar chart for the minimum prices of all the features. In subFig. 2(a) we see that the Adj Close feature gives the least price whilst the High attribute gives the maximum price. SubFig. 2(b) is a bar chart for the maximum prices of all the features. It revealed that the Adj Close feature yields the least price whilst the High gives the maximum prices.

SubFig. 3(a) is a bar chart for the mean prices of all the features. In subFig. 3(a) we perceived that the Adj Close and High features gives the minimum and maximum prices respectively. In subFig. 3(b) we present a bar chart for median prices of all the features. It is also observed that both the least and maximum prices are from the Adj Close and High attributes respectively.

Fig. 4 revealed that the use of 3 CNN layers with the Adam optimiser gives the least error and thus best performance. It is also seen that the Adam optimiser yields the least total error for the entire CNN layer combinations. The RMSprop optimiser is seen to give the maximum total error for the entire CNN layer combinations.

In Fig. 5 we present the outcomes of an experiment carried out to assess our proposed model's performance against the baseline schemes by using three metrics. We also observe that the EHTS model outruns all the benchmark algorithms. Also, it is seen that the PBMT algorithm emerges to be the second-best. A linear regression algorithm emerges as the worst scheme.

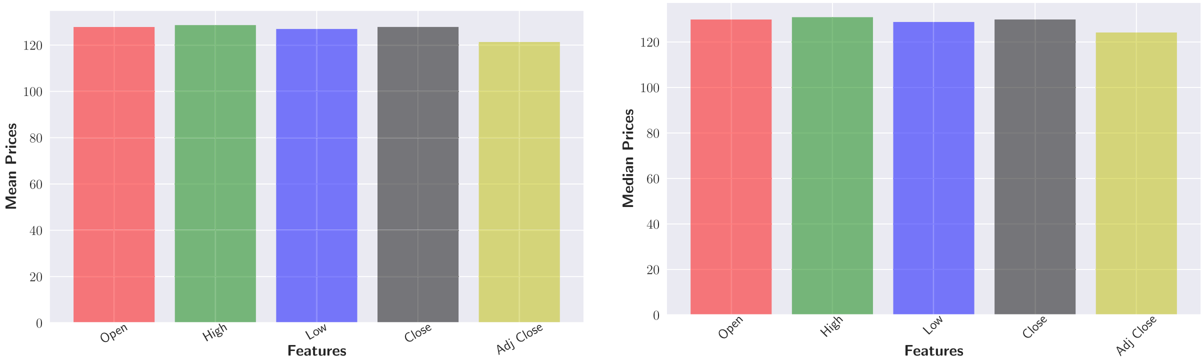

Fig. 3. (a) Bar Chart showing mean prices for 5 of our features in the JNJ dataset; (b) Bar Chart showing median prices for 5 of our features in the JNJ dataset.

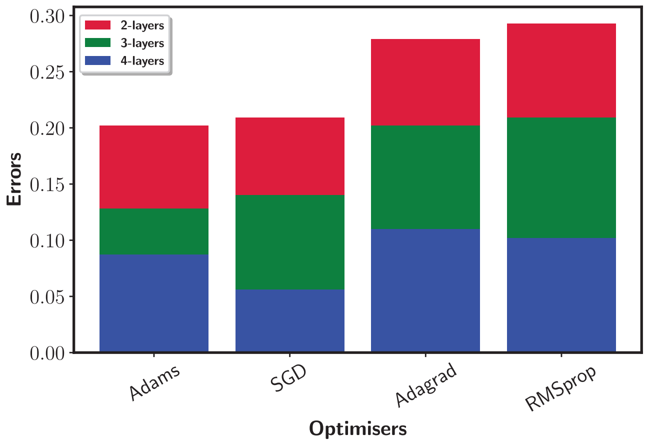

Fig. 4. Bar Chart showing the performances of three sets of CNN layer combination for 4 optimisers in terms of training errors for the JNJ stock dataset.

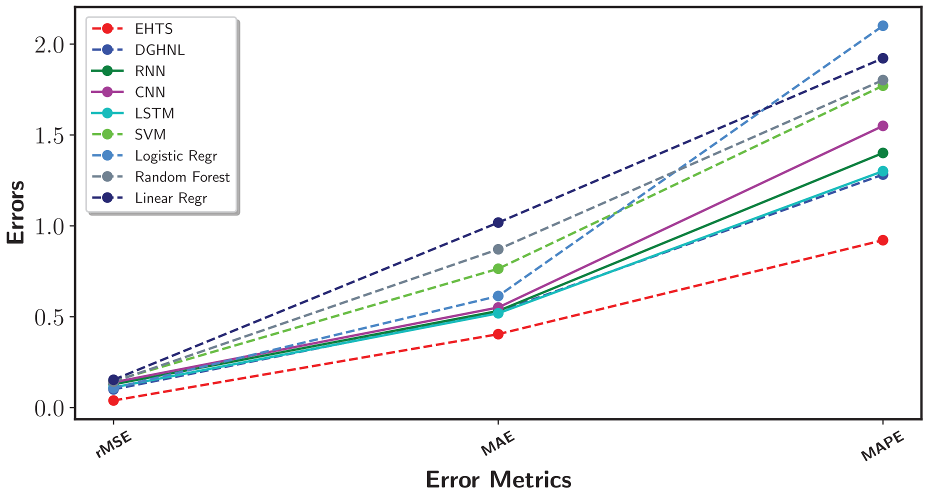

Fig. 5. A plot showing the performance of EHTS and entire baseline schemes on the JNJ dataset via three error metrics.

With regards to the MA indicator the following rationale is utilise. The intersection points for short and long-term MAs will emerge as potential buying signal points if only its numerical value is smaller than the immediate preceding maximum value of the short-term MA. A possible selling signal point crops up when the value of the point of intersection of a short and long-term MA is greater than the immediate preceding minimum value of the short-term MA.

From Fig. 6 we observed a significant number of buying and selling signal points. However, we have selected only 4 points for every scenario to demonstrate the applied rationale. In Fig. 6 it is seen that the potential buying points are denoted by B_0, B_1, B_2 and B_3 whilst the selling signal points are represented by S_0, S_1, S_2 and S_3.

For the MACD scenario a short-term analysis is performed on our stock market via the following rationale. The intersection points of the MACD curve and signal line only emerges to be a potential selling signal point if it is above the zero line. Secondly, its value must be lower than the immediate preceding maximum value of the MACD curve starting from the left. However, the intersection point of the MACD curve and signal line can also crops up to be a potential buying signal point if its occur below the zero line. And also the value at the intersection point must be greater than the immediate preceding minimum value of the MACD curve. In Fig. 7 we observed many buying and selling signal points but we have hand-picked a few for demonstrative purposes. In Fig. 7 we let $S_0, S_1, S_2, S_3, S_4, S_5$ and S_6 denote potential selling signal points and let $B_0, B_1, B_2, B_3, B_4, B_5$, and B_6 denote the possible buying signal points. Selling signal points represent the most suitable periods for traders to trade stocks and realise sufficient profit whilst potential buying signal points are appropriate periods for traders to buy stocks in order make maximum profit in the future.

In the MACD histogram layout we consider the above rationale to analyse our stock market. In this situation we will employ the frequency polygon created by the whole MACD histogram. Thus, we define a frequency polygon to be a plot generated by joining the midpoints of the MACD histogram. Potential selling regions will emerge when the frequency polygons occur above the zero line and possess negative gradients. These regions indicate appropriate selling periods for traders to realise maximum profit. In a similar manner, potential buying areas will crop up when frequency curves reside on the bottom of the zero line and possess positive gradients. These regions are signals to traders telling them to buy stocks. From Fig. 7 we observed that s_0, s_1, and s_2 denote potential selling regions/periods whereas b_0, b_1, and b_2 denote possible buying regions/periods.

Ultimately, for the RSI metric we employed the following logic to analysis our stock market. The point of intersection of the RSI curve and overbuy line will crops up to be a potential selling signal point if only it's value is lower than the immediate preceding maximum value from left. The above rationale is a signal for traders to sell and realise maximum profit. In Fig. 8 we let the points S_0, S_1, S_2, S_3 and S_4 denote the prospective selling signal points. A future buying signal point will emerges when the value of the intersection point for an oversell and RSI curve seems more prominent than the immediate preceding RSI minimum value from the left. These are appropriate periods for traders to buy stocks and realise maximum profit in the future. In Fig. 8 we let the points B_0, B_1, B_2 and B_3 denote prospective buying signal points.

From Fig. 9 we observed periods of low traded volumes (i.e., V_0 and V_1) and periods of high traded volumes (i.e., V_2 and V_3). However, the periods associated with V_0, V_1, V_2 and V_3 are not the only times of low and high traded volumes but we have randomly selected them for illustrative purposes. Under typical market situations V_0 and V_1 occur due to high prices of stocks while V_2 and V_3 occur due to low stock prices. Although V_2 and V_3 sometimes occur at high stock prices. These are the influences of psychological factors on the costs of stocks, and one such factor is news. For example if there is news that the price of a stock that is too high at the moment will further rise shortly then the trade volume of the stock will increase immediately.

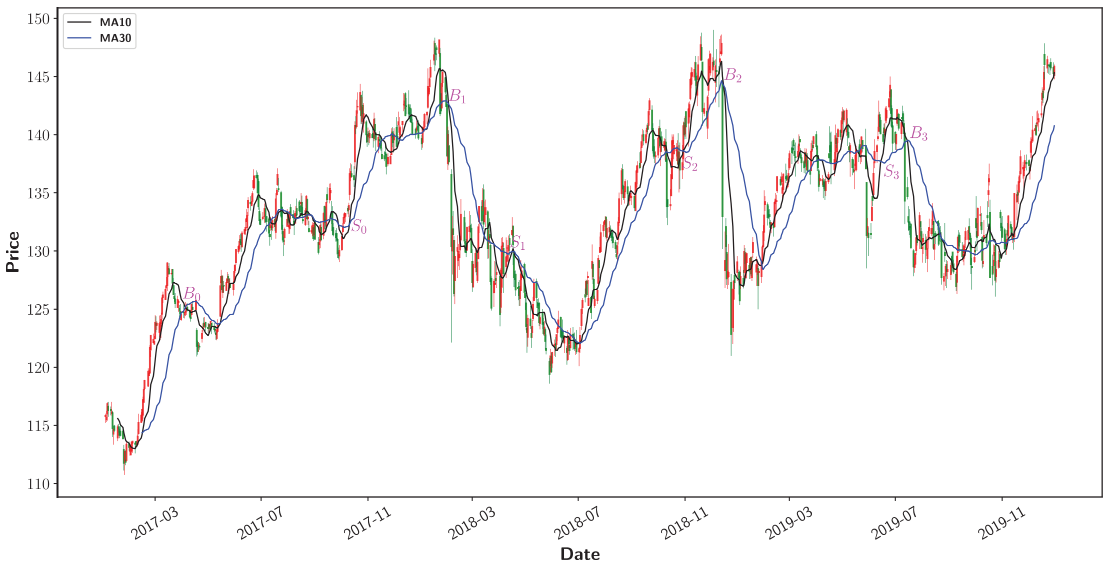

Fig. 6. A plot showing two simple MAs for the JNJ stock prices.

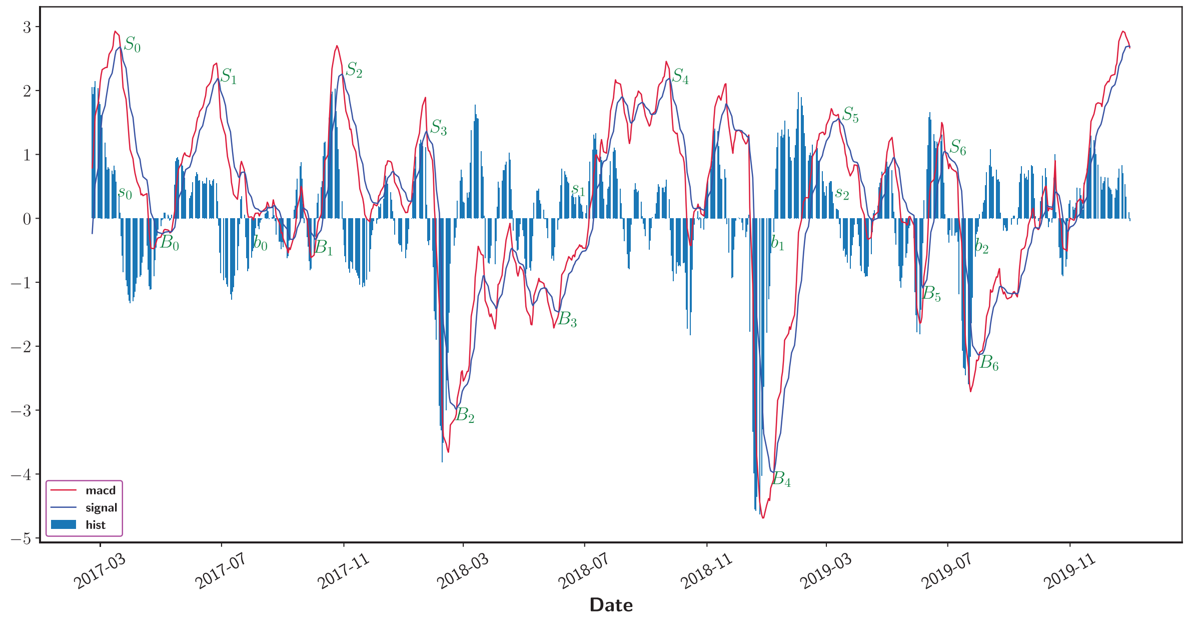

Fig. 7. A plot showing MACD line, signal line, and MACD histogram for the JNJ stock.

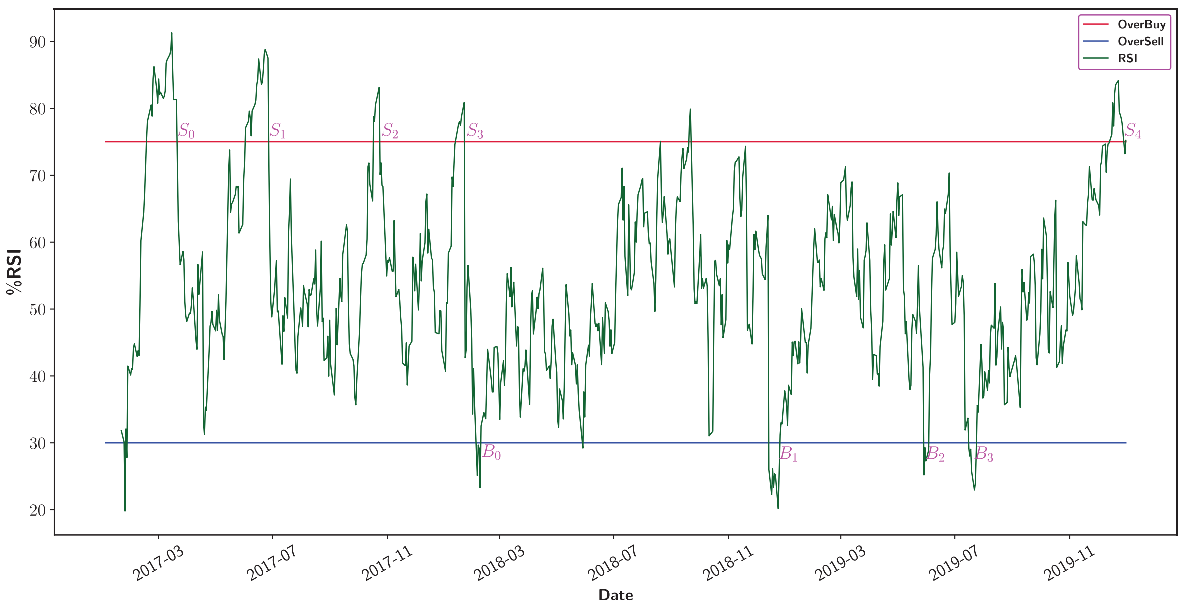

Fig. 8. A plot showing the RSI, OverSell, and Overbuy lines for the JNJ stock.

6. Conclusions

In this work, we present EHTS a unique scheme that has the ability to produced superb results over the existing baseline models. Our focus in this paper is to investigate the performance of the projected scheme on two popular datasets namely: $J\&J$ and BAC datasets. Our approach essentially standardised the data before feeding in the algorithm due to high data values especially from the attribute volume. Moreover, our scheme captures attributes from the data in a self-supervised modus operandi through $AB-CNN$ and $CB-LSTM$ arms, concatenate the outcomes and later feed them into the last stand-alone MLP block to perform the forecasting job. The model as mentioned earlier is well-tailored to forecast the stock prices of our data effectively. After the standardisation of our data we thoroughly trained our proposed model and made a breakthrough over the existing baseline schemes. The vast difference between our proposed and the eleven benchmark schemes

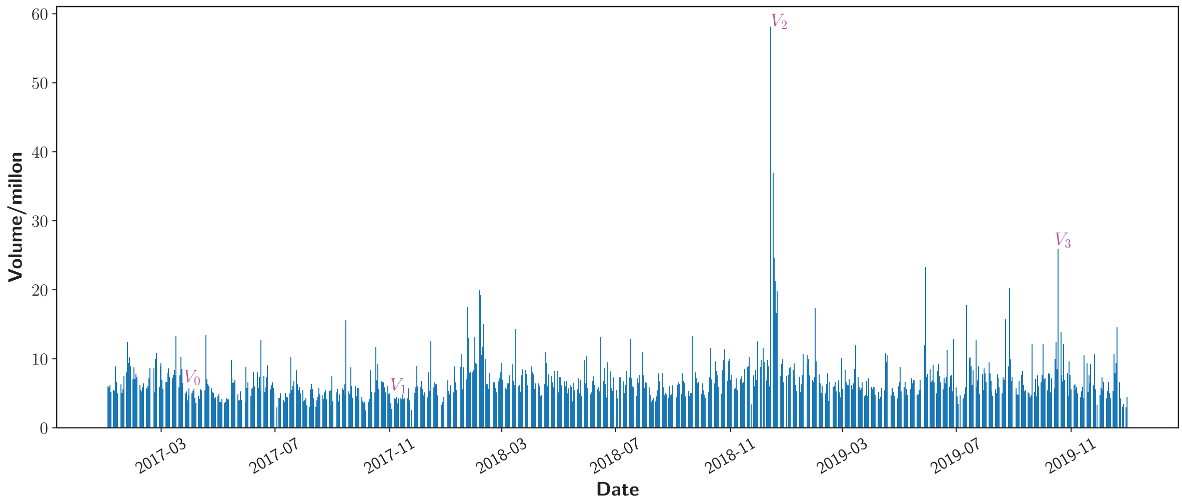

Fig. 9. A plot showing the traded Volumes for the JNJ stock.

revealed that EHTS can augment the performance of the existing methods in tackling the stock price forecasting problem. The merits of our proposed scheme are as follows: 1.) no manual extraction of features and 2.) the use of appropriate standalone deep learning models to efficiently extract stock market features of all types. One of the demerits of the EHTS scheme is that it can only be train on a GPU due to its complexity. Another disadvantage is that it requires large amount of datasets. Our future research will endeavour to apply this algorithm on heterogeneous datasets that is a particular dataset will be utilised for training while we do the testing on another dataset. Secondly, we will endeavour to increase scheme performance by introducing context in the $AB - CNN$ module. Lastly, future research will consider the merits of both numerical and textual data and utilise natural language processing approaches to boost the prediction performance of our model further

CRediT authorship contribution statement

Amadu Fullah Kamara: Conceptualization, Methodology, Software, Writing – original draft, Visualization. **Enhong Chen:** Supervision, Data curation, Investigation. **Zhen Pan:** Software, Validation, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

I want to thank the National Key Research and Development Program of China (No. 2016YFB1000904) and the National Natural Science Foundation of China (Grants No. U1605251, 61727809) for partially aiding my research via grants.

References

[1] D. Song, A.M.C. Baek, N. Kim, Forecasting stock market indices using padding-based fourier transform denoising and time series deep learning models, IEEE Access.
[2] D. Cheng, F. Yang, S. Xiang, J. Liu, Financial time series forecasting with multi-modality graph neural network, Pattern Recogn. 121 (2022) 108218.
[3] E. Chodakowska, J. Nazarko, Ł. Nazarko, Arima models in electrical load forecasting and their robustness to noise, Energies 14 (23) (2021) 7952.
[4] H. Gujral, A.K. Kushwaha, S. Khurana, Utilization of time series tools in life-sciences and neuroscience, Neurosci. Insights 15 (2020), 2633105520963045.
[5] Y. Xing, Z. Hu, Z. Huang, C. Lv, D. Cao, E. Velenis, Multi-scale driver behaviors reasoning system for intelligent vehicles based on a joint deep learning framework, in: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2020, pp. 4410–4415.
[6] Z. Hu, Y. Xing, C. Lv, P. Hang, J. Liu, Deep convolutional neural network-based bernoulli heatmap for head pose estimation, Neurocomputing 436 (2021) 198–209.
[7] J. Zhang, Y. Li, W. Xiao, Z. Zhang, Non-iterative and fast deep learning: Multilayer extreme learning machines, J. Franklin Inst. 357 (13) (2020) 8925–8955.
[8] V. Ingle, S. Deshmukh, Spdl-stock price prediction with deep learning, ICCIP-2021.
[9] J. Wu, Y. Ji, S. Li, Genetic-algorithm-based convolutional neural network for robust time series classification with unreliable data, Sens. Mater. 33 (4) (2021) 1149–1165.
[10] T. Damrongsakmethee, V.-E. Neagoe, Stock market prediction using a deep learning approach, in: 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), IEEE, 2020, pp. 1–6.
[11] H. Wang, D. Liu, J. Zhang, J. Wang, Y. Ma, Y. Lian, Log-periodic power law hybrid model based on bp neural network, Evol. Intel. (2021) 1–9.

[12] G.D. Sharma, B. Erkut, M. Jain, T. Kaya, M. Mahendru, M. Srivastava, R.S. Uppal, S. Singh, Sailing through the covid-19 crisis by using ai for financial market predictions, Math. Problems Eng. (2020).
[13] S. Dong, J. Wang, H. Luo, H. Wang, F.-X. Wu, A dynamic predictor selection algorithm for predicting stock market movement, Expert Syst. Appl. 186 (2021) 115836.
[14] K. Khalil, B. Dey, A. Kumar, M. Bayoumi, A reversible-logic based architecture for long short-term memory (lstm) network, in: 2021 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, 2021, pp. 1–5.
[15] M.S. Roodsari, M.A. Saber, Z. Navabi, Diba: n-dimensional bitslice architecture for lstm implementation, in: 2020 23rd International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), IEEE, 2020, pp. 1–6.
[16] J.Y. Lim, K.M. Lim, C.P. Lee, Stacked bidirectional long short-term memory for stock market analysis, in: 2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), IEEE, 2021, pp. 1–5.
[17] S. De, A.K. Dey, D.K. Gouda, Construction of confidence interval for a univariate stock price signal predicted through long short term memory network, Ann. Data Sci. (2020) 1–14.
[18] K. Karuppiah, N. Umamaheswari, R. Venkatesh, Analysis on indian stock market prediction using deep learning models, in: Challenges and Applications of Data Analytics in Social Perspectives, IGI Global, 2021, pp. 80–90.
[19] D. Kumar, Stock forecasting using natural language and recurrent network, in: 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), IEEE, 2020, pp. 1–5.
[20] M. Castán-Lascorz, P. Jiménez-Herrera, A. Troncoso, G. Asencio-Cortés, A new hybrid method for predicting univariate and multivariate time series based on pattern forecasting, Inf. Sci.
[21] W. Long, Z. Lu, L. Cui, Deep learning-based feature engineering for stock price movement prediction, Knowl.-Based Syst. 164 (2019) 163–173.
[22] A.F. Kamara, E. Chen, Q. Liu, Z. Pan, Combining contextual neural networks for time series classification, Neurocomputing 384 (2020) 57–66.
[23] Q. Chen, W. Zhang, Y. Lou, Forecasting stock prices using a hybrid deep learning model integrating attention mechanism, multi-layer perceptron, and bidirectional long-short term memory neural network, IEEE Access 8 (2020) 117365–117376.
[24] W. Jiang, Applications of deep learning in stock market prediction: recent progress, Expert Syst. Appl. 184 (2021) 115537.
[25] Y. Hao, Q. Gao, Predicting the trend of stock market index using the hybrid neural network based on multiple time scale feature learning, Appl. Sci. 10 (11) (2020) 3961.
[26] Y. Cao, T.A. Geddes, J.Y.H. Yang, P. Yang, Ensemble deep learning in bioinformatics, Nat. Mach. Intell. 2 (9) (2020) 500–508.
[27] S. Zhang, Y. Chen, W. Zhang, R. Feng, A novel ensemble deep learning model with dynamic error correction and multi-objective ensemble pruning for time series forecasting, Inf. Sci. 544 (2021) 427–445.
[28] Y. Zhang, C. Jia, C.K. Kwoh, Predicting the interaction biomolecule types for lncrna: an ensemble deep learning approach, Briefings Bioinf.
[29] M. Abdar, M. Zomorodi-Moghadam, X. Zhou, R. Gururajan, X. Tao, P.D. Barua, R. Gururajan, A new nested ensemble technique for automated diagnosis of breast cancer, Pattern Recogn. Lett. 132 (2020) 123–131.
[30] S. Carta, A. Ferreira, A.S. Podda, D.R. Recupero, A. Sanna, Multi-dqn: An ensemble of deep q-learning agents for stock market forecasting, Expert Syst. Appl. 164 (2021) 113820.
[31] Y. Xia, K. Chen, Y. Yang, Multi-label classification with weighted classifier selection and stacked ensemble, Inf. Sci.
[32] M. Abdar, U.R. Acharya, N. Sarrafzadegan, V. Makarenkov, Ne-nu-svc: A new nested ensemble clinical decision support system for effective diagnosis of coronary artery disease, IEEE Access 7 (2019) 167605–167620.
[33] S. Tuli, N. Basumatary, S.S. Gill, M. Kahani, R.C. Arya, G.S. Wander, R. Buyya, Healthfog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated iot and fog computing environments, Future Gener. Comput. Syst. 104 (2020) 187–200.
[34] A. Koohestani, M. Abdar, A. Khosravi, S. Nahavandi, M. Koohestani, Integration of ensemble and evolutionary machine learning algorithms for monitoring diver behavior using physiological signals, IEEE Access 7 (2019) 98971–98992.
[35] D. Liang, B. Yi, Two-stage three-way enhanced technique for ensemble learning in inclusive policy text classification, Inf. Sci. 547 (2021) 271–288.
[36] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, A. Jatowt, Yake! keyword extraction from single documents using multiple local features, Inf. Sci. 509 (2020) 257–289.
[37] W. Chen, M. Jiang, W.-G. Zhang, Z. Chen, A novel graph convolutional feature based convolutional neural network for stock trend prediction, Inf. Sci. 556 (2021) 67–94.
[38] J. Chai, H. Zeng, A. Li, E.W. Ngai, Deep learning in computer vision: A critical review of emerging techniques and application scenarios, Mach. Learn. Appl. 6 (2021) 100134.
[39] B. Soundarya, R. Krishnaraj, S. Mythili, Visual speech recognition using convolutional neural network, in: IOP Conference Series: Materials Science and Engineering, vol. 1084, IOP Publishing, 2021, p. 012020.
[40] Y. Liu, C. Gong, L. Yang, Y. Chen, Dstp-rnn: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction, Expert Syst. Appl. 143 (2020) 113082.
[41] Q. Huang, C. Huang, X. Wang, F. Jiang, Facial expression recognition with grid-wise attention and visual transformer, Inf. Sci. 580 (2021) 35–54.
[42] H. Jahangir, H. Tayarani, S.S. Gougheri, M.A. Golkar, A. Ahmadian, A. Elkamel, Deep learning-based forecasting approach in smart grids with microclustering and bidirectional lstm network, IEEE Trans. Industr. Electron. 68 (9) (2020) 8298–8309.
[43] A.F. Kamara, E. Chen, Q. Liu, Z. Pan, A hybrid neural network for predicting days on market a measure of liquidity in real estate industry, Knowl.-Based Syst. 208 (2020) 106417.
[44] A. Jierula, S. Wang, T.-M. Oh, P. Wang, Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data, Appl. Sci. 11 (5) (2021) 2314.
[45] Q. Ye, F. Ahammed, Quantification of relationship between annual daily maximum temperature and annual daily maximum rainfall in south australia, Atmospheric Ocean. Sci. Lett. 13 (4) (2020) 286–293.
[46] P. Pławiak, M. Abdar, J. Pławiak, V. Makarenkov, U.R. Acharya, Dghnl: A new deep genetic hierarchical network of learners for prediction of credit scoring, Inf. Sci. 516 (2020) 401–418.
[47] M. Hammad, P. Pławiak, K. Wang, U.R. Acharya, Resnet-attention model for human authentication using ecg signals, Expert Syst. (2020) e12547.
[48] P. Pławiak, U.R. Acharya, Novel deep genetic ensemble of classifiers for arrhythmia detection using ecg signals, Neural Comput. Appl. 32 (15) (2020) 11137–11161.
[49] M. Kurtz, J. Kopinsky, R. Gelashvili, A. Matveev, J. Carr, M. Goin, W. Leiserson, S. Moore, N. Shavit, D. Alistarh, Inducing and exploiting activation sparsity for fast inference on deep neural networks, in: International Conference on Machine Learning, PMLR, 2020, pp. 5533–5543.
[50] G. Alfonso Pérez, Forecasting and optimization of stock trades.