Proposed system model

The prediction of stock markets is an important area of research for researchers, academicians, and investors. The investment strategies of institutional and individual investors are based upon these prediction models with the prime objective of earning profits and creating liquidity in the market. In this regard, this research work is an effort to analyze the sampled stock markets for proposing a better prediction model with enhanced accuracy. Hence, a recurrent DL prediction model the peephole LSTM with a TAL is proposed to efficiently predict the trend of four stock markets (U.S., U.K., India and China). [Fig. 2](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue#fg0030) presents the architecture of the proposed system model. A recurrent architecture is proposed as recurrence is more appropriate for processing the complex sequential time series data. This is a novel hybrid model which has not been used earlier for financial time series prediction, as to the best of our knowledge.

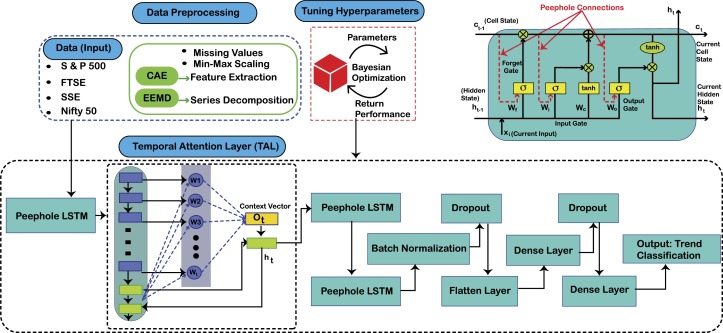
[](https://www.cell.com/cms/10.1016/j.heliyon.2024.e27747/asset/6a8035bc-2820-4b2f-86f1-692d8d484d88/main.assets/gr003_lrg.jpg)

Figure viewer

**Figure 2 Proposed System Model PLSTM-TAL.**

An LSTM is a special RNN with gate mechanism and is better capable of processing sequential data and to capture both short and long-term dependencies in the time series. The forget and input gates allow the retention of useful historical information in the hidden and cell states. This retained information is further used for predicting the future data trends. The introduction of peephole in the LSTM architecture develops a linkage between the cell state and gates of LSTM providing access of already stored information in the cell state to the gates [[38]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). This enhances the prediction efficiency of the LSTM for sequential data with long-term patterns [[39]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). The hyperparameters of LSTM are fine-tuned with Bayesian optimization which is computationally more efficient than the random and grid search optimization methods as it also considers previous choices while optimizing a given set of parameters [[20]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). The final output of LSTM is then passed to the TAL which is introduced so that the special instances in the time series are given due attention before predicting the final trend of the stock market. Results of the proposed PLSTM-TAL have been compared with those of SVM, RF, CNN and LSTM (without feature extraction and series decomposition). The performance of all classification models have been validated using the evaluation metrics of accuracy, precision, recall, F1-score, Area Under the Receiver Operating Characteristics Curve (AUC-ROC), Area under the Precision Recall Curve (PR-AUC) and Matthew's Correlation Coefficient (MCC). The proposed hybrid model achieved higher performance scores than those of the benchmark models.

4.1 Data labelling

In this article, the trend classification of stocks' time series is binary, that is an upward trend if the return at a given time t+1 is greater than the return at time t [[40]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). Hence, the target variable 𝑦𝑖⁡(𝑡) of the sample *i* at time *t* is given in equation [(1)](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue#fm0010) as follows:

𝑦𝑖⁡(𝑡)={1,𝑖⁢𝑓𝑟𝑖⁡(𝑡+1)>𝑟𝑖⁡(𝑡),0,𝑂⁢𝑡⁢ℎ⁢𝑒⁢𝑟⁡𝑤⁢𝑖⁢𝑠⁢𝑒.

**(1)**

Here, 𝑟𝑖⁢(𝑡+1) is the return at time t+1 whereas 𝑟𝑖⁡(𝑡) is the return at time *t*. The class label is one for an upward trend and is zero for a downward trend.

4.2 Data preprocessing

At the foremost step, the open, high, low, close prices and volume of stocks' indices have been considered, and 40 technical indicators have been calculated from these OHLCV values. These are the same indicators as considered in [[17]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). The data is preprocessed for missing values and is then standardized using min-max scaling method. The dimensionality of the data has been reduced with contractive autoencoder and the extracted data contains the most pertinent features helpful for the enhanced performance of the proposed prediction model. The closing price time series have been denoised using the series decomposition algorithm of EEMD which is a noise assisted method that adds Gaussian noise to the original price series and then decomposes it into Intrinsic Mode Functions (IMFs). The IMF with higher complexity and least energy is subtracted from the original closing price generating a filtered time series. This filtered series is less complex and chaotic than the original series however, at the cost of losing some information. This filtered series along with other covariates of TIs have been used as input to the system model following the approach used in [[17]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue).

4.3 Contractive autoencoder

The contractive autoencoder is used for features extraction in this article. It was initially proposed in [[18]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue) as a regularized autoencoder which is used for learning representations for the subsequent classification tasks. The loss function of a traditional autoencoder is mathematically presented in equation [(2)](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue#fm0020). However, the CAE loss function is given in equation [(3)](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue#fm0030). The second term of this equation is the CAE regularization/penalty term that is expressed as the sum of squares of all the dimensional differential of the feature space. This regularization term is only applied to the training examples forcing the model to learn the salient patterns in the training dataset. Moreover, this term relates to the Frobenius norm of the Jacobian matrix of the encoder activations for the input. The use of Jacobian term enhances the locally invariant and robust encoding representations that suppresses the impact of noise speckles. Thus, the CAE is efficient for discrimination and robust feature representation. The inclusion of an explicit regularization term in the objective function is the exclusive property of the CAE making it robust to slight fluctuations in the input data and therefore it works well with financial data. This autoencoder is termed as ‘contractive’ as it contracts the dimensions of inputs neighborhood by mapping them to a smaller outputs' neighborhood. The process of dimensionality reduction with the CAE is presented in lines 5-11 of the [Algorithm 2](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue#fg0040).

||⁢𝐽ℎ⁡(𝑋)⁢||2𝐹=∑𝑖𝑗⁢(∂ℎ𝑗⁡(𝑋)/∂𝑋𝑖)2

**(2)**

𝐿⁡𝑜⁢𝑠⁡𝑠𝐶⁢𝐴⁢𝐸⁡(𝜃)=∑(𝐿⁡(𝑥,𝑔⁡(ℎ⁡(𝑋))))+𝜆⁢||⁢𝐽ℎ⁡(𝑋)⁢||2𝐹

**(3)**

4.4 Ensemble empirical mode decomposition

A time series is a combination of different harmonics the details of which can be studied using the frequency domain of signal analysis. Various methods like Fourier transformation and spectral analysis have been used for this purpose. However, filtering the signal for noise becomes very difficult especially when the fundamental harmonics are being shared by the noise [[16]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). Moreover, methods like Fourier transformation do not perform well for non-linear and non-stationary signals like stock prices. Therefore, an adaptive analysis tool EEMD proposed by Wu and Huang in [[41]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue) is used in this study for the decomposition of stock price signals. It is basically an extension of the Empirical Mode Decomposition (EMD) and was proposed to address the problem of mode mixing in EMD. It is an iterative method to decompose the given series into a set of orthogonal components called IMFs which are the quasi-stationary components of the original signal. The process of EEMD for sifting the original signal has following steps [[10]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue):

Step 1: Add white Gaussian noise to the original series x(t) as: 𝑥𝑖⁡(𝑡)=𝑥⁡(𝑡)+𝑤⁢𝑛𝑖⁡(𝑡) for i = 1, 2, 3, ..., n.

Here, 𝑤⁢𝑛(⁢𝑖) are the different realizations of the white Gaussian noise.

Step 2: Identify the local maxima and minima of the resultant time series 𝑥𝑖⁡(𝑡).

Step 3: Create an upper envelope 𝑒𝑚⁢𝑎⁢𝑥⁡(𝑡) for all local maxima and a lower envelope 𝑒𝑚⁢𝑖⁢𝑛⁡(𝑡) for all local minima, using spline interpolation.

Step 4: Compute the average of the two envelopes as 𝑚𝑖⁡(𝑡)=(𝑒𝑚⁡𝑎⁢𝑥⁡(𝑡)+𝑒𝑚⁡𝑖⁢𝑛⁡(𝑡))/2

Step 5: Subtract this average from the data, ℎ𝑖⁡(𝑡)=𝑥⁡(𝑡)−𝑚𝑖⁡(𝑡). The ℎ𝑖⁡(𝑡) is an IMF (𝑐𝑖) if it satisfies the following two conditions:

**•**

The ℎ𝑖⁡(𝑡) is symmetric and may have only one extremum.

**•**

Its mean is approximately zero.

Step 6: If ℎ𝑖⁡(𝑡) does not meet the above two conditions, then it replaces the original series 𝑥(⁢𝑡) and steps 1-6 are repeated until it meets the criteria of being an IMF.

Step 7: When ℎ𝑖⁡(𝑡) is an IMF (𝑐𝑖⁡(𝑡)), then the residual signal in 𝑥⁡(𝑡)=ℎ𝑖⁡(𝑡)+𝑟𝑖⁡(𝑡) is then passed through the same sifting procedure until it becomes a monotonic function from which no more IMFs can be extracted.

Finally, the IMFs can be summed to get the final series with the residue as: 𝑋𝑡=∑𝑁𝑛=1𝑐𝑖⁡(𝑡)+𝑟𝑁⁡(𝑡)

The process of EEMD is presented in [Algorithm 1](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue#fg0020). The input, output and the variables are given in lines 1-4 and lines 5-21 present the decomposition process of the price series. It is pertinent to mention that the IMFs of a signal can be modeled and predicted independently and then the resultant predictions are reconstructed to get a final prediction. However, the accumulative effect of the prediction errors of sub-signals aggravates the final prediction error. Therefore, a filtered series is generated in this study by subtracting the most complex and chaotic IMF (𝐶1) from the original series. The Sample Entropy (SaEn) value is used as a proxy for the complexity of IMFs presented by their noise level. The noise level of each IMF is calculated as per the lines 22-32 of the [Algorithm 1](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue#fg0020). Afterwards, only the filtered series is predicted by the proposed model. This is the same approach as followed in [[16]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue).

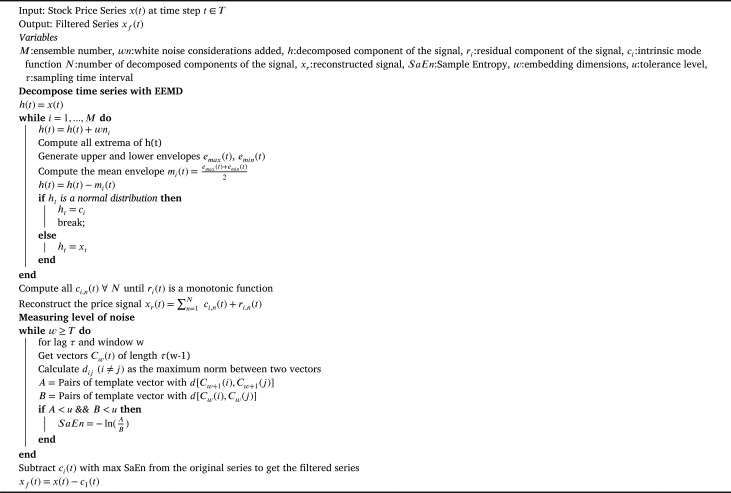
[](https://www.cell.com/cms/10.1016/j.heliyon.2024.e27747/asset/1638521f-01b7-4a9b-bf93-48b1c49fe1ea/main.assets/gr002_lrg.jpg)

Figure viewer

**Algorithm 1 Filtering Stock Price Signals for Noise using EEMD and SaEn**

4.5 Long short term memory

RNNs were developed to overcome the limitations of ANNs in dealing with sequential data like text and images etc. They have been successfully used in speech recognition and natural language processing tasks. Their main purpose was to predict the next probable outcome by using the order in the data. However, they could only remember few recent orders or patterns because of their short-term memory structures and therefore they cannot relate older data patterns with current ones. Moreover, they are prone to the vanishing gradient problem [[42]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). In order to overcome these limitations of RNNs, Hochreiter and Schmidhuber in [[43]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue) developed a special version of RNNs as LSTM network by introducing a memory line the Constant Error Carrousel (CEC). The LSTM derives its name from its structure that helps in developing long-term memory (the cell state) with the help of short-term memory (the hidden state). In simpler words, it is an artificial neural network with feedback connections capable of processing not only single data points but entire sequences of the data. The feedback mechanism of an LSTM is comprised of the CEC and the three gates (forget, input and output). The cell retains information over arbitrary long-term intervals while the gates control the flow of information by keeping or forgetting it. The information is filtered on the basis of its relevance and importance to predict the future values. An LSTM is a viable deep network solution for time series data which may have time lags of indefinite duration between important events.

In a recent study, a novel optimization approach based on a multi-layered sequential LSTM with adam optimizer has been used for stock prediction. The proposed model illustrated the training and testing accuracies of 96% and 98% respectively thus outperforming other ML and DL algorithms [[28]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). In another work [[6]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue), the stock markets' time series have been forecasted with dual-LSTM using the sequentially moving window approach. The dual-LSTM outperformed all benchmark models and showed that RNN based models can forecast better than the classical models like ARIMA.

4.6 Peephole LSTM

One of the limitations of the traditional LSTM is the inability of its gates to access the output of the memory unit when the output gate is closed. Moreover, the learning of precise timings of fluctuations in a time series is not possible unless the memory cell is allowed to control the gates and that is only possible with an open output gate. To address this problem, Gers and Schmidhuber in [[38]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue) proposed an LSTM architecture with a peephole providing access of the cell state to the gates of LSTM even when the output gate is closed. With the peephole, the gates of the LSTM can access the cell state even when the output gate is closed [[39]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). Therefore, the forget, input and output gates of the LSTM consider the cell state as input and are updated accordingly as presented in the equations [(4)](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue#fm0040) - [(10)](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue#fm0100). However, the only difference of peephole LSTM from that of the original LSTM is the calculation of additional connections, while the rest of the working principle is same. The LSTM gates can access the CEC with the peephole connections.

𝑓𝑡=𝜎𝑧⁡(𝑊𝑓.[𝐶𝑡−1,ℎ𝑡−1,𝑥𝑡]+𝑏𝑓)

**(4)**

𝑖𝑡=𝜎𝑧⁡(𝑊𝑖.[𝐶𝑡−1,ℎ𝑡−1,𝑥𝑡]+𝑏𝑖)

**(5)**

ˆ𝑐𝑡=𝜎𝑧⁢(𝑊𝑐⁢𝑥𝑡+𝑈𝑐⁢ℎ𝑡−1+𝑏𝑐),

**(6)**

𝑐𝑡=𝑓𝑡⁎𝑐𝑡−1+𝑖𝑡⁎ˆ𝑐𝑡,

**(7)**

𝑜𝑡=𝜎𝑧⁡(𝑊𝑜.[𝐶𝑡,ℎ𝑡−1,𝑥𝑡]+𝑏𝑜)

**(8)**

ℎ𝑡=𝑜𝑡⁎𝜎𝑧⁡(𝑐𝑡).

**(9)**

𝑦𝑡=ℎ𝑡

**(10)**

In the above PLSTM equations, 𝑓𝑡,𝑖𝑡,𝑜𝑡,𝑐𝑡,ℎ𝑡 and ˆ𝑐𝑡, represent output of the forget gate, input gate, output gate, cell state, hidden state and predicted value of the cell state respectively. Moreover, 𝑥𝑡 represents the input data at time t; 𝐶𝑡−1 and ℎ𝑡−1 are the cell and hidden states at time t-1; 𝑊𝑓,𝑊𝑖,𝑊𝑜 and 𝑊𝑐 represent the weight matrices of the forget gate, input gate, output gate and the cell state and 𝑏𝑓,𝑏𝑖,𝑏𝑜,𝑏𝑐 are the offsets (bias terms) respectively. 𝜎𝑧 is the sigmoid activation function. Other activation functions like tanh can also be used. Lastly, 𝑦𝑡 is the final output (label).

The connection of LSTM gates with 𝐶𝑡−1 significantly enhances the accuracy of sequence-to-sequence prediction tasks. The peephole convolutional LSTM has been used to extract abstractive summaries [[44]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue), predict wind speed [[39]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue), for electricity load prediction [[45]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue) and for theft detection in smart grid [[46]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). However, the usefulness of the peephole LSTM has not been proved in some studies [[45]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). This study considers peephole LSTM in lieu of simple LSTM as the former is more appropriate for long-term sequence forecasting problems. The detailed algorithm for the proposed PLSTM-TAL is given in [Algorithm 2](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue#fg0040). Lines 1-4 describe the input space, output and the variables. The working of peephole LSTM with TAL is presented in lines 12-30 of the same algorithm.

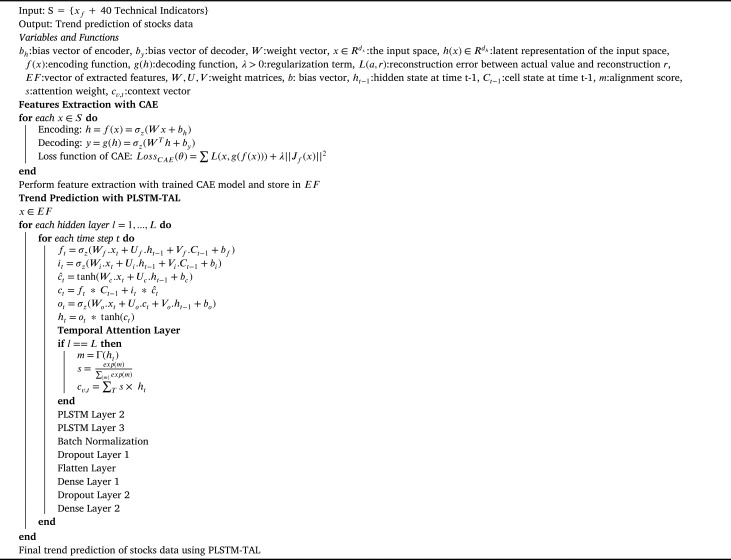
[](https://www.cell.com/cms/10.1016/j.heliyon.2024.e27747/asset/985f156e-808f-40b3-86ee-b115ce18f606/main.assets/gr004_lrg.jpg)

Figure viewer

**Algorithm 2 The Proposed Model PLSTM-TAL**

4.7 Bayesian optimization

The utility of LSTM for a particular problem depends upon the values of its hyperparameters. For instance, size of the time window (input) contains contextual information which is important for forecasting price trends. A small input window is usually deficient of significant signals while a large window is often noisy with extra information. Moreover, number of hidden layers and number of neurons in those layers decide upon the computational power of the model. While, many studies applied LSTM with manual selection of hyperparameters, others applied some heuristics algorithms for optimization [[30]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). In this study, the bayesian optimization method for optimization of LSTM hyperparameters is applied. Bayesian method was proposed by Jonas Mockus in [[47]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue) as a sequential method for global optimization of black-box models. The bayesian method converges to the optimal solution in fewer iterations and performs better than other optimization techniques like random search and grid search [[20]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). Unlike random and grid search methods, bayesian algorithm speeds up the optimization process while considering the previous performances. Whereas, grid and random search methods perform independent of their previous evaluations. Thus, bayesian provides more efficient optimization as it converges to the optimized solution in fewer iterations [[48]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). The optimized values of the hyperparameters of our proposed model are given in [Table 2](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue#tbl0020).

| **Hyperparameters** | **Range of values** | **Optimal value** |
| --- | --- | --- |
| Units | 16, 32, 64, 128 | 64 |
| Activation Function | relu, tanh, linear, sigmoid | tanh |
| Optimizer | SGD, Adam, Adamax, RMSprop, Adagard | Adamax |
| Loss | binary crossentropy, hinge, square hinge | binary crossentropy |
| Dropout | 0.1, 0.2, 0.3, 0.4 | 0.1 |

**Table 2**

Tuning of PLSTM-TAL Hyperparameters using Bayesian Optimization.

* [Open table in a new tab](https://www.cell.com/action/showFullTableHTML?isHtml=true&tableId=tbl0020&pii=S2405-8440%2824%2903778-2)

4.8 Attention layer

A feed-forward NN considers all input features as unique and independent implying that nothing can be inferred from the current feature about the next consecutive feature. This approach is suitable for a data having no association and dependencies among variables. However, it is not viable in the presence of an underlying local structure to the data. Besides, the DL models like RNNs consider an input as a long data string and generate an output of shorter length thus losing some information. RNNs have to develop connections between lengthy input and output sequences with dozens of words. RNNs complemented with attention mechanism are capable of predicting a particular output sequence by focusing on certain parts of the input sequence. LSTMs like other advanced models have their limitations specially when dealing with long data sequences. LSTM based encoder-decoder networks have been used for time-series forecasting purposes, particularly when sequence-to-sequency mapping is required. These models have performed exceptionally well with small sequences however their performance declines with longer sequences [[19]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue) and [[49]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue). Moreover, LSTMs can circumvent the vanishing gradient problem [[50]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue), yet they are sensitive to the exploding gradient issue. In textual data analysis, LSTMs like other RNNs, give higher weights to words in closer proximities and the upstream context is emphasized higher than the downstream context. The attention mechanism is capable of addressing the aforementioned limitations. The attention mechanism was proposed by Bahdanau et al. in [[49]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue) that is a three steps process to compute alignment scores, weights and context vector. It was introduced to increase the performance of ML tasks using encoder-decoder models. Our brain does not consider and process the whole set of overwhelming background information, rather it picks and processes only the important information for the task at hand and discards the rest of it rendering it as irrelevant. This feature of brain is termed as its ‘attention mechanism’. The same can be incorporated in NNs. The NNs with attention principle are adept to comprehend the sequential data like textual, video, voice or time series. Main purpose of attention is to filter the data for important sequences be them spatial or temporal. It compares current inputs with the previously stored ones. The attention-based NNs are computationally faster than the RNNs and LSTMs in capturing the time dependent context-based patterns from the data. This study uses attention layer in the proposed model because the data of stocks indices is time dependent and contextual. Besides making accurate predictions, the attention LSTM also helps the researchers in understanding the reasons for these predictions by providing intermediate outputs. In [[19]](https://www.cell.com/heliyon/fulltext/S2405-8440(24)03778-2?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS2405844024037782%3Fshowall%3Dtrue), the authors applied attention layer to the inputs focusing more on relevant features with respect to time and then they integrated a TAL to emphasize on the relevant temporal hidden states of the LSTM units. The attention layer is integrated at the output of the hidden states of the proposed LSTM model.