A Best Algorithm for Short-Term and Long-Term Stock Price Prediction

Surat Banerjee[1] Sabyasachi Bandyopadhyay[2]
[1,2]Department of Information Technology
[1,2]Maulana Abul Kalam Azad University of Technology, India

Abstract— Stock price forecasting may be a popular and important topic in financial and academic studies. Share A market is an untidy place for predicting since there are not any significant rules to estimate or predict the price of shares within the share market. Many methods like technical analysis, fundamental analysis, statistic analysis, and statistical analysis, etc. are all wont to plan to predict the worth within the share market but none of those methods are proved as a consistently acceptable prediction tool. In this project, we attempt to implement different algorithms (Moving Average, Linear Regression, KNN, ARIMA, LSTM) to predict stock market prices and will check which algorithm gives us the best prediction. Artificial Neural networks are very effectively implemented in forecasting stock prices, returns, and stock modeling, and therefore the most frequent methodology. We select a certain group of parameters with a relatively significant impact on the share price of a company. With the assistance of statistical analysis, the relation between the chosen factors and share price is formulated which may help in forecasting accurate results. Although the share market can never be predicted, due to its vague dons, this project aims at applying different Time Series Analysis Algorithm and Non-Time series Analysis Algorithm in forecasting the stock prices.

Keywords: Machine Learning, Data Preprocessing, Data Mining, Moving Average, Linear Regression, KNN, ARIMA, LSTM, RNN, Backpropagation

I. INTRODUCTION

The stock market is basically an aggregation of various buyers and sellers of the stock. A stock (also referred to as shares more commonly) generally represents ownership claims on business by a specific individual or a gaggle of individuals. The attempt to determine the future value of the stock price is known as a stock price prediction. The prediction is expected to be accurate, and efficient. The system must work consistently with real-life scenarios and will be suited to real-world settings. The system is additionally expected to require under consideration all the variables which may affect the stock's value and performance. There are many methods and ways of implementing the prediction system like Fundamental Analysis, Technical Analysis, Machine Learning, Market Mimicry, and Time-series aspect structuring. With the advancement of the digital era, the prediction has moved up into the technological realm. The most prominent and promising technique involves the use of Artificial Neural Networks, Recurrent Neural Networks, which is basically the implementation of machine learning. Machine learning in- volves AI which empowers the system to find out and improve from past experiences without being programmed time and again. Traditional methods of prediction in machine learning use algorithms like Backward Propagation, also mentioned as Backpropagation errors. Many researchers are using more ensemble learning techniques. It would use low price and time lags to predict future highs while another network would use lagged highs to predict future highs.

Stock price prediction for short time windows appears to be a random process. The stock price movement over an extended period of your time usually develops a linear curve. People tend to shop for those stocks whose prices are expected to rise within the near future. The uncertainty within the stock exchange refrain people from investing in stocks. Thus, there's a requirement to accurately predict the stock exchange which may be utilized in a real-life scenario. The methods used to predict the stock market includes time series forecasting along with technical analysis, machine learning modeling, and predicting the variable stock market. The datasets of the stock exchange prediction model include details just like the price opening price, the data, and various other variables that are needed to predict the thing variable which is the price in a given day. The aim is to style a model that gains from the market information utilizing machine learning strategies and gauge the longer-term patterns available value development.

The stock exchange is a crucial part of the economy of the country and plays an important role within the growth of the industry and commerce of the country that eventually affects the economy of the country. Both investors and industry are involved within the stock exchange and need to understand whether some stock will rise or go over a particular period of your time. The stock exchange is that the primary source for any company to boost funds for business expansions. It is based on the concept of demand and supply. If the demand for a company's stock is higher, then the corporate share price increases and if the demand for the corporate's stock is low then the company share price decrease.

II. PROBLEM DEFINITION

Stock value prediction is largely outlined as attempting to see the stock price and supply a strong plan for the individuals to understand and predict the stock costs. The matter with estimating the stock value can stay a drag if a more robust stock value prediction rule isn't planned. Predicting however the securities market can perform is kind of tough. The movement within the stock value is typically determined by the feelings of thousands of investors. Stock value prediction involves the capability to predict the impact of recent events on the investors. These events may be political events sort of a statement by a politician, a bit of report on scams, etc. It may also be a global event like sharp movements in currencies and commodities etc. of these events have an effect on the company earnings, which successively affects the sentiment of investors. It's on the far side the scope of almost all investors to properly and systematically predict

these hyper parameters. These factors create stock value prediction terribly tough. Once the right knowledge is collected, it than area unit typically used to train a machine and to urge a prognostic result.

III. OBJECTIVE

The aims of this project are as follows:

1) To identify factors poignant the share worth.
2) Extract the pattern from the large set that's suffering from stock worth changes.
3) To predict the associate approximate value of share worth.
4) Implement utterly totally different algorithms on a continuing dataset and predict the stock worth for the constant quantity (Short-Term and Long-Term).
5) Get recommendations from Actual stock worth with the expected stock value.
6) Conclude that rule is that the simplest.

The project goes to be useful for investors to require an edge inside the exchange supported varied factors. The project target is to use a special rule that analyses previous stock worth data for an organization and implement these values in processing and machine learning rule to figure out the value that specific stock will have in about to future with acceptable accuracy.

The main feature of this project is to induce associate correct foretelling output victimization utterly totally different algorithms and build a general arrange of future values supported by the previous data by generating a pattern. The scope of this project does not exceed quite a generalized content tool and in addition recommends the best prediction rule.

IV. FINDING

1) Weekday returns remained but different days and Friday returns remained larger than different days.
2) Positive come back is detected for the months of January, February, June, July, August, and December above different months. Most come back within the month of Gregorian calendar month.
3) Negative returns area unit detected for the months of March, April, May, September, October, and Gregorian calendar month.
4) Semi-Month impact is detected in animal disease, wherever a better come back is detected throughout the primary 0.5 months than the last half month.
5) The higher than literature survey confirms that the January impact stock returns area unit detected in the Asian nation.
6) Indian Tax year ends in March the higher than literature survey confirmed tax-loss marketing within the Indian market to tax year impact.

V. SYSTEM ARCHITECTURE

Kaggle is a web community for data analysis and predictive modeling. It also contains a dataset of different fields, which is contributed by data miners. Various data scientist competes to create the best models for predicting and depicting the information. It allows the users to use their datasets so that they can build models and work with various data science engineers to solve various real-life data science challenges. The dataset used in the proposed project has been downloaded from Kaggle. However, this data set is present in what we call raw format. The data set may be a collection of stock exchange information for a few companies.

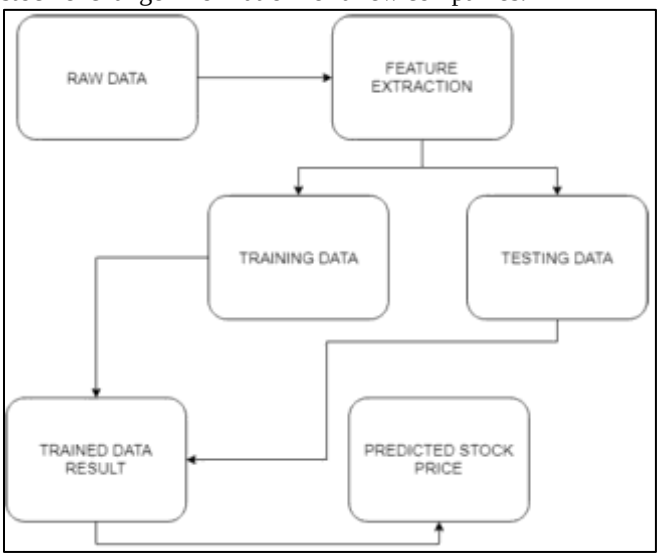


Fig. 1: System Flow Diagram

The first step is that the conversion of this data into processed data. This is done using feature extraction, since in the raw data collected there are multiple attributes but only a few of those attributes are useful for the purpose of prediction. So, the primary step is feature extraction, where the key attributes are extracted from the entire list of attributes available within the raw dataset. Feature extraction starts from an initial state of measured data and builds derived valuesor features. These features are intended to be informative and non-redundant, facilitating subsequent learning and generalization steps. Feature extraction may be a dimensionality reduction process, where the initial set of raw variables is diminished to progressively reasonable features for simple management, while still precisely and totally depicting the first informational collection.

The feature extraction process is followed by a classification process wherein the info that was obtained after feature extraction is split into two different and distinct segments. Classification is that the issue of recognizing to which set of categories a replacement observation belongs. The training data set is used to train the model whereas the test data is used to predict the accuracy of the model. The splitting is done in a way that training data maintain a higher proportion than the test data.

VI. MODULES AND FUNCTIONALITY

A. Data Assortment

Data assortment may be a terribly basic module and also the initial step towards the project. It usually deals with the gathering of the right dataset. The dataset that's to be employed in the market prediction should be wont to be filtered supported numerous aspects.

Data assortment additionally enhances to boost the knowledge set by adding a lot of data that square measure external. Our knowledge chiefly consists of the previous

year's stock costs. Initially, we'll be analyzing the NSE – TATA world dataset from Kaggle and according to the accuracy, we'll be exploiting the model with the information to research the predictions accurately.

B. Pre-Processing

Data pre-processing could also be a locality of the knowledge process that involves remodeling knowledge into a lot of coherent formats. Data is typical, inconsistent, or incomplete and typically contains several errors. The information pre- processing involves sorting out for missing values, attempting to search out categorical values, cacophonous the data-set into coaching and take a look at the set, and eventually do a feature scaling to limit the variety of variables in order that they'll be compared on common geographic region.

C. Training the Machine

Training the machine is analogous to feeding the info—the knowledge—the information to the algorithmic program to touch up the take a look at data. The coaching sets square measure accustomed tune and works the models. The take a look at sets square measure untouched, as a model shouldn't be judged supported unseen knowledge. The coaching of the model includes cross-validation wherever we tend to get a reasoned approximate performance of the model exploitation the coaching knowledge. Calibration models square measure meant to specifically tune the hyperparameters just like the range of trees during a random forest. We tend to perform the full cross-validation loop on every set of hyperparameter values.

Finally, we'll calculate a score, for individual sets of hyperparameters. Then, we tend to choose the most effective hyperparameters. The concept behind the coaching of the model is that we tend to some initial values with the dataset then optimize the parameters that we might prefer to among the model. This is often unbroken on repetition till we tend to get the best values. Thus, we tend to take the predictions from the trained model on the inputs from the take a look at the dataset. Hence, it's divided within the quantitative relation of eighty:20 wherever 80 p.c is for the coaching set and also the rest twenty p.c for a testing set of the information.

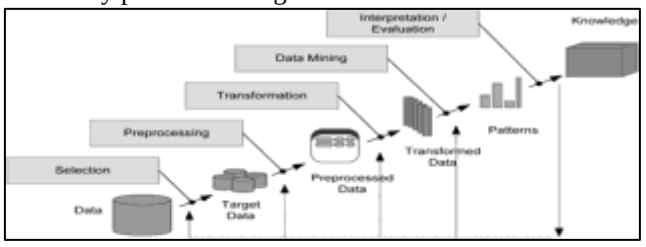
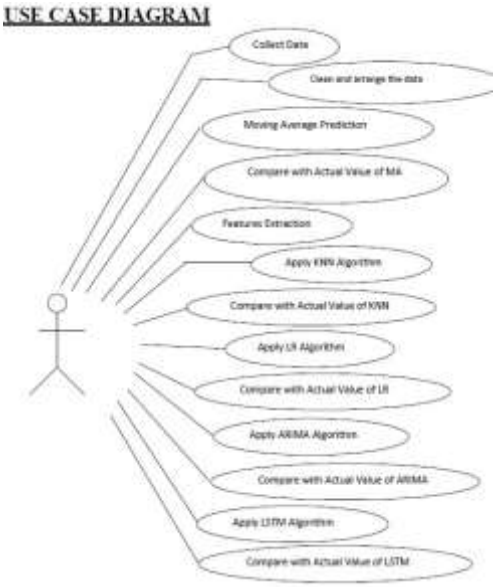


Fig. 2: Module Diagram

D. Data Scoring

The process of applying a prognostic model to a bunch of information is mentioned as grading the data. Supported the coaching models, we tend to succeed in attention-grabbing results. The last module so describes however the results of the model will facilitate to predict the chance of a stock to rise and sink supported by bound parameters. It additionally shows the vulnerabilities of a selected stock or entity. The user authentication system management is enforced to create positive that solely the licensed entities square measure accessing the results.

VII. METHODOLOGIES



WE WILL IMPLEMENT THE FOLLOWING ALGORITHMS FOR 15-DAYS, 45-DAYS AND 90-DAYS PREDICTION AND COMPARE RESULT TO FIND THE BEST ALGORITHM TO PREDICT THE SHARE PRICE.

1) Moving Average
2) Linear Regression
3) K-Nearest Neighbors
4) ARIMA
5) Long Short-Term Memory (LSTM)

VIII. ARIMA ARCHITECTURE

ARIMA stands for Auto-Regressive Integrated Moving Average. It's a way of modeling time series data for forecasting (i.e., for predicting future points in the series).

ARIMA has three components – AR (Autoregressive Term), I (Differencing Term), and MA (Moving Average Term).

A. Steps for ARIMA implementation

The steps to implement an ARIMA model are –

1) Load the data: The first step for model building is of course to load the dataset.
2) Pre-processing: looking at the dataset, the steps of preprocessing are going to be outlined. This can embody making timestamps, changing the dtype of the date/time column, creating the series univariate, etc.
3) Make series stationary: In order to satisfy the assumption, it is necessary to make the series stationary. This would include checking the stationarity of the series and performing required transformations.
4) Determine d value: For making the series stationary, the number of times the difference operation was performed will be taken as the d value.
5) Create ACF and PACF plots: This is the most important step in ARIMA implementation. ACF, PACF plots are accustomed verify the input parameters for our ARIMA model

6) Determine the p and q values: Read the values of p and q from the plots in the previous step.
7) Fit ARIMA model: Using the processed data and parameter values we calculated from the previous steps, fit the ARIMA model.
8) Predict values on validation set: Predict the future values.
9) Calculate RMSE: To check the performance of the model, check the RMSE value using the predictions and actual values on the validation set.

IX. RNN AND LSTM ARCHITECTURE

A. An overview of Recurrent Neural Network (RNN)

In a classical neural network, final outputs seldom act as an output for subsequent steps but if we concentrate on a real-world phenomenon, we observe that in many situations our final output depends not only on the external inputs but also on earlier output. For example, when humans read a book, understanding of every sentence depends not only on the present list of words but also on the understanding of the previous sentence or on the context that's created using past sentences. Humans do not start their thinking from scratch every second. As you read this essay, you understand each word supported your understanding of previous words. This concept of 'context' or 'persistence' is not available with classical neural networks. The inability to use context-based reasoning becomes a major limitation of traditional neural net- works. Recurrent neural networks (RNN) are conceptualized to alleviate this limitation. RNN are networked with feedback loops within to allow the persistence of information.

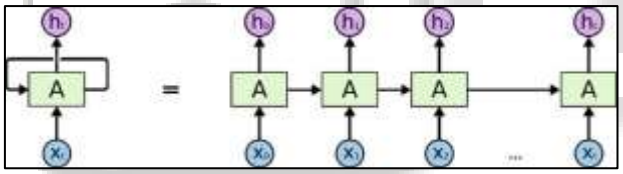


Fig. 3: An unrolled recurrent neural network

At first, (at time step t) for some input Xt the RNN generates an output of ht. In the next time step (t+1) the RNN takes two inputs Xt+1 and ht to generate the output ht+1. A loop allows the information to be passed from one step of the network to the next. RNNs are not free from limitations though. When the 'context' is from the near past it works great towards the correct output. But when an RNN has to depend on a distant 'context' (i.e. something learned long past) to produce correct output, it fails miserably.

A. *LSTM Networks*

Hochreiter and Schmidhuber introduced a special type of RNN which is capable of learning long term dependencies. Later on, many other researchers improved upon this pioneering work. LSTMs are perfected overtime to mitigate the long- term dependency issue. The evolution and development of LSTM from RNNs are explained in. Recurrent neural networks are in the form of a chain of repeating modules of the neural network. In standard RNNs, this repeating module has a simple structure like a single tanh layer as shown in Figure 4.

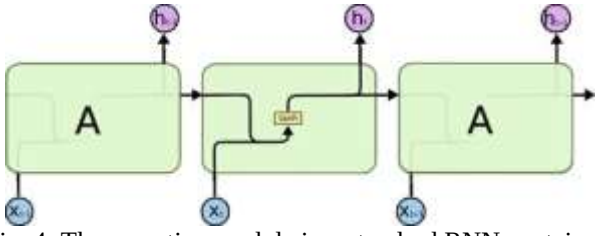


Fig. 4: The repeating module in a standard RNN contains a single layer

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.

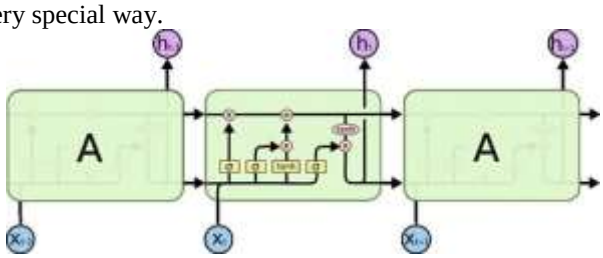
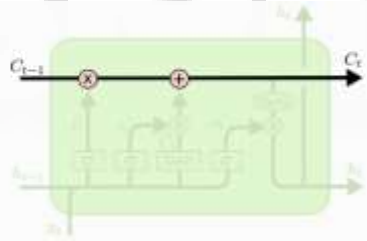


Fig. 5. The repeating module in an LSTM contains four interacting layers

B. *The Working of LSTM*

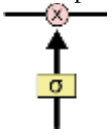
The key to LSTMs is the cell state, the horizontal line running through the top of the diagram.

The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged



The LSTM does have the ability to remove or add information to the cell state, carefully regulated by structures called gates.

Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.

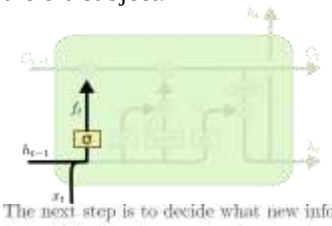


The sigmoid layer outputs numbers between zero and one, describing how much of each component should be let through. A value of zero means "let nothing through," while a value of one means "let everything through!"
An LSTM has three of these gates, to protect and control the cell state.

C. *Briefly LSTM Working*

The first step in our LSTM is to decide what information we're going to throw away from the cell state. This decision is made by a sigmoid layer called the "forget gate layer." It looks at h_{t-1} and x_t, and outputs a number between 0 and 1 for each number in the cell state C_{t-1}. A 1 represents "completely keep this" while a 0 represents "completely get rid of this."

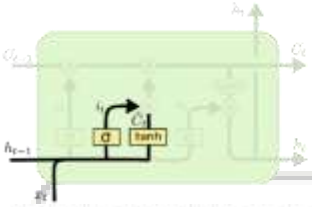
Let's go back to our example of a language model trying to predict the next word based on all the previous ones. In such a problem, the cell state might include the gender of the present subject, so that the correct pronouns can be used. When we see a new subject, we want to forget the gender of the old subject.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

The next step is to decide what new information we're going to store in the cell state. This has two parts. First, a sigmoid layer called the "input gate layer" decides which values we'll update. Next, a tanh layer creates a vector of new candidate values, $\tilde{C}_t$, that could be added to the state. In the next step, we'll combine these two to create an update to the state.

In the example of our language model, we'd want to add the gender of the new subject to the cell state, to replace the old one we're forgetting.

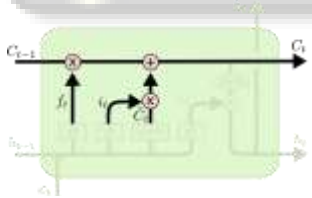


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

It's now time to update the old cell state, C_{t-1}, into the new cell state C_t. The previous steps already decided what to do, we just need to actually do it.

We multiply the old state by f_t, forgetting the things we decided to forget earlier. Then we add $i_t * \tilde{C}_t$. This is the new candidate values, scaled by how much we decided to update each state value.

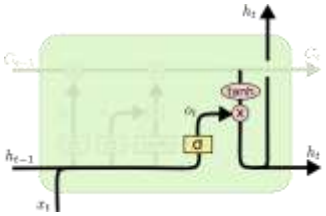
In the case of the language model, this is where we'd actually drop the information about the old subject's gender and add the new information, as we decided in the previous steps.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Finally, we'd like to make a decision on what we're getting to output. This output will be based on our cell state but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state we're getting to output. After that, we put the cell state through tanh and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

For the language model example, since it just saw a topic, it would want to output information relevant to a verb, just in case that's what's coming next. i.e, it might output where the subject is singular or plural so that we know what form a verb should be conjugated into if that's what follows next.



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

X. ALGORITHM COMPARISON WITH RESULT

ALGORITHM	TIME IN DAYS	RMSE	OBSERVATION
MOVING AVERAGE (Window Size 7)	**15**	**4.0762**	**SUITABLE FOR ALL TIME PERIOD PREDICTION**
	45	4.2988	
	90	5.7359	
MOVING AVERAGE (Window Size 90)	15	9.4363	NOT Suitable because RMSE Value is Large
	45	15.2000	
	90	19.4395	
LINEAR REGRESSION	**15**	**5.3079**	Suitable for Short- and Medium-Term Prediction
	45	6.4737	
	90	16.7328	
KNN	15	89.8367	NOT SUITABLE
	45	98.4237	
	90	105.8836	
ARIMA	**15**	**4.4175**	Suitable for Short Term Prediction
	45	14.7645	
	90	24.1911	
LSTM	15	6.2628	**SUITABLE FOR ALL TIME PERIOD PREDICTION**
	45	4.6568	
	90	6.5316	

XI. CONCLUSION

1) Moving Average is suitable for Time Series Analysis (Stock Prediction) for all time period, if we take small window size (i.e. 6 previous value). Here RMSE value is nearly same for all time period prediction.If we take long window size(90), it we will not suitable for prediction for any time period.
2) Linear Regression is suitable for Short Term and Mid Term Prediction. Not Suitable for Long term Prediction.
3) KNN is not suitable for Stock Price Prediction because RMSE value is very high.
4) ARIMA is used for Time series Analysis but on this data set it is not performing well for Mid-Term and Long-Term Prediction but for Short Term it performs good.
5) LSTM is suitable for Time Series Analysis (Stock Prediction) for all time period to predict the stock price. Here RMSE value is nearly same for long-term and short-term prediction, so this algorithm can be used for any time period prediction.

REFERENCES

[1] R. Wilson and R. Sharda, ”Bankruptcy prediction using neural net- works”, Decision Support Systems, vol. 11, no. 5, pp. 545-557, 1994. [2]Elliott wave theory and neuro-fuzzy systems, in stock market prediction:
[2] The WASP system. Expert Systems with Applications, 38, 9196–9206 [3]K. Tsai and J. Wang, ”External technology sourcing and innovation performance in LMT sectors”, Research Policy, vol. 38, no. 3, pp. 518-526, 2009.
[3] K. Han and J. Kim, ”Genetic quantum algorithm and its application to combinatorial optimization problem”, Evolutionary Computation, 2000, vol. 2, pp 1354-1360, 2000.
[4] ”Market Capitalisation Definition from Financial Times Lexicon”, Lex- icon.ft.com, 2016. [Online].
[5] ”MarketTrak’s Forecast Model Overview”, Markettrak.com, 2016. [On- line].
[6] M. Roondiwala, H. Patel and S. Varma, ”Predicting stock prices using LSTM,” International Journal of Science and Research (IJSR), vol. 6, no. 4, pp. 1754-1756, 2017.
[7] T. Kim and H. Y. Kim, ”Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data,” PloS one, vol. 14, no. 2, p. e0212320, April 2019.
[8] Y. Bengio, P. Simard, P. Frasconi and others, ”Learning long-term dependencies with gradient descent is difficult,” IEEE transactions on neural networks, vol. 5, no. 2, pp. 157-166, 1994.
[9] S. Hochreiter and J. Schmidhuber, ”LSTM can solve hard long time lag problems,” in Advances in neural information processing systems, NIPS, 1997, pp. 473–479.
[10] S. Hochreiter, ”The vanishing gradient problem during learning recurrent neural nets and problem solutions,” International Journal of
[11] Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 6, no. 2, pp. 107-116, 1998. J. Schmidhuber, D. Wierstra, M. Gagliolo and F. Gomez, ”Training recurrent networks by evolino,” Neural computation, vol. 19, no. 3, pp. 757-779, 2007.

