Received: 11 October 2022 | Revised: 18 November 2022 | Accepted: 29 December 2022

IET Software

DOI: 10.1049/sfw2.12092

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Prediction of stock market movement via technical analysis of stock data stored on blockchain using novel History Bits based machine learning algorithm

Nitin Nandkumar Sakhare[1] | Imambi S. Shaik[1] | Suman Saha[2]

[1]Computer Science and Engineering Department, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India

[2]Department of ICT, Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh, Gazipur, Bangladesh

Correspondence

Suman Saha, Department of ICT, Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh, Gazipur, Bangladesh.
Email: suman@ict.bdu.ac.bd

Nitin Nandkumar Sakhare, Computer Science and Engineering Department, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, 522302, India.
Email: nitinsakhare4@gmail.com

Abstract

Analysts and investors use data on market activity, such as historical returns, stock prices-open, high, low, close, and volume of trades to chart patterns in securities movement. With technical analysis investors can get mixed signals. For stock market predictions using technical analysis, various Machine Learning algorithms are available. A novel algorithm formulated on History Bits is hatched for deriving beneficial facts from the massive established dataset which is stored on a private blockchain for the quick retrieval and avoid data manipulation. The proposed algorithm predicts the trading call out of five different calls-strong buy, strong sell, buy, sell, and hold. For the implementation and testing of the History Bits based algorithm, 75 technical parameters are computed using stock trading data (open, high, low, close prices, and volume), prioritised using ensemble-based rank search strategy acting as an input for the proposed algorithm. For the experimentation, transformed NIFTY 50 dataset was used over the time frame of 20 years. The performance of proposed History Bits model is compared with Decision Tree, Naïve Bayes, Random Forest, Support Vector Machine and Multilayer Perceptron Artificial Neural Network, algorithms. History Bits algorithm outperforms machine learning algorithms in terms of prediction accuracy.

KEYWORDS

data mining, decision making, feature extraction, learning (artificial intelligence)

1 | INTRODUCTION

Each and every investor invests in different kinds of commodities intending to create assets or multiply their funds. An example of this is the stock market where there are multiple companies listed and several stocks are available for the investor to buy or sell [1]. In the stock market, investors offer for stocks by providing a specific price, and sellers request a special price. At the point when these two prices coordinate, a deal happens [2]. Frequently, numerous financial specialists are offering on a similar stock. At the point when this happens, the first investor to put the offer is the first to get the share. Prediction of equity prices is the analysis of historical prices to decide the future estimation of an organisation stock exchanged on a trade. Persistent unsettlement in the securities exchange is real motivation behind why financial specialists sell out at the off-base time and frequently neglect to pick up the advantage [3]. This could potentially cause uncertainty in trading decisions. The fruitful forecast of a stock's future prices could return noteworthy benefit. Technical analysis is associated with stocks, records, products, fates, or any tradable instrument where the price is impacted by the powers of free-market activity-demand and supply. The period can be founded on intraday (1-min, 5-min, 10-min, 15-min, 30-min or hourly), every day, week after week or month to month value information and last a couple of hours or numerous years. The aim of technical analysis is to enable investors to make preferable & safe investment decisions and identify trading opportunities. The profit earned is the way to measure the efficiency of technical analysis [4]. The free market activity of stocks all relies on technical analysis. Different indicators have different significance for technical analysis [5]. Based on various indicators and types, investors can define suitable strategies. Strategies are predefined conditions that determine entry, exit, and/or trade settlement. This research paper does not focus on any developing specific trading strategies; however, it gives direction explanation of how indicators can be used for technical analysis to predict stock market direction using machine learning techniques and thereby helping investors whether to buy or sell stocks [6–8]. It is commonly advised most technical analysts not to use technical indicators alone for technical analysis [9]. Technical indicators should be used in a group for analysis. Such trading strategies can be formulated using a machine learning-based technical analysis approach. This approach helps analysts/traders to analyse a massive amount of stock market data and give prediction results with more accuracy and within less time so that traders can identify profitable opportunities, eliminate risks and adjust or update investment strategy if required [10]. To enable quick retrieval of data and to ensure immutability of the data, the dataset is stored on a private blockchain. Section 3 describes the machine learning algorithms and working of the novel History Bits algorithm; Section 4 gives comparative analysis of History Bits based machine learning model with various machine learning techniques used by analysts for the stock movement prediction problem; Section 5 concludes the research work.

2 | LITERATURE REVIEW

Since a long time, many ML algorithms have been implemented for prediction of stock price movements. Tsai C, and Quan proposed a stock prediction system by identifying similarities using candlestick charts [11]. These charts were prepared using technical indicators. They did not explore the significance of indicators. Prediction results can be significantly improved if indicators of different types are part of the prediction process. Feng et al. used deep learning-neural modelling with temporal graph convolution approach [12]. They developed a model based on deep learning for identifying stocks that would yield the highest expected profit. For better results, other machine learning and deep learning algorithms can be studied, and their performance can be evaluated to suggest the best strategy or decision making. Xiaodong Li used extreme learning machine based on Support Vector Machine–Back Propagation Neural Networks for prediction. They used two types of datasets for prediction: Historical stock prices and news related to the market. This work can be extended with fundamental analysis of the stock market for long term investments and also the interpretation of various machine learning algorithms can be studied to get more accurate results. Calzon et al. developed a group-based decision making for selecting optimised indicators. They used an expert selection-based approach to preventing market decision [13]. This work can be extended to a combination of decision from a group of market experts with the trading signal from technical indicators to get more accurate prediction results. Sheikh and Agarwal did time series analysis of Indian Stock Market. They developed different models- Auto-Regressive Model, Moving Average Model, and Auto-Regressive with Moving Average Model for prediction of stock market volatility. This work would have created more significance with the use of the combination of technical indicators based on moving averages, volatility, oscillations etc. Pehlivanli and Guzhan had done significant work on indicator selection. They used features selection technique with information gain for prediction of 1 day ahead stock price. An optimal subset of indicators is selected by removing irrelevant and redundant indicators [14]. Other attribute evaluation techniques like principal component analysis [15], Pearson correlation coefficient, information gain, gain ratio can be used to validate an optimal subset of indicator. Jean-Marc Le Cailee had focussed on performance comparison of machine algorithms, and for analysis, they used different combinations of indicators [16]. This work can create more sense when combinations of indicators are based on one combination containing indicators belonging to different types. The same kind of indicators could create more ambiguity in predictions. For example, a combination of all moving average-based indicators could lead to false signals. Ross Barmish developed a stock trading model based on classical feedback controllers. This work can be extended with statistics based technical indicators for analysis based on backtracking. Wen developed a novel methodology to analyse noisy financial time series via sequence reconstruction using frequent patterns [17]. Different machine learning methods can be experimented to generate frequent patterns. Eunsuk Chong performed a detailed study of deep learning method to understand the consequence of three unsupervised or self-governed feature extractors-principal component analysis, autoencoder and restricted Boltzmann machine for prediction of stock prices. They improved the performance of the autoregressive model with feature extractors [18]. Feature extractors applied in a broader dimension of technical indicators can generate enough number of subsets with optimal indicators. Rudra Kalyan Nayak developed a hybrid approach based on Support Vector Machine (SVM) and k nearest neighbours for technical analysis of the Indian stock market. They used different kernel functions to understand its impact on prediction of profit or loss. Other machine learning algorithms can be used, and the performance of different models can be analysed [17].

Forecast of security exchange development is amazingly troublesome because of its high impermanent nature. Machine leaning techniques help in revealing meaningful patterns and gaining insights which help in the process of decision making. For example, machine learning techniques enable us to predict the coming trend of the stock market considering many factors affecting the stock price. This knowledge can be a key to gain profits and gain more insights within the market. Machine learning is a well-known and proven technique in a far-reaching range of applications and has been broadly studied for its capacities in the projection of financial markets. Technical analysis is often performed for short term trading which includes statistical evaluation of technical indicators like trend indicators, momentum indicators, volatility indicators and volume indicators [19]. Despite the fact that it is beyond the realm of imagination foresee securities exchange development with full exactness, misfortunes from selling stocks at wrong time and its effects can be lessen to more noteworthy degree utilising forecast of financial exchange development dependent on investigation of authentic information. Financial specialists dependably need exact forecasts and they should utilise stock data carefully. An extraordinary amount of sequential information is accessible with regards to securities exchange conduct. In this work, we represent a novel algorithm-History Bits for prediction of stock market. We have also studied comparison of the performance of proposed model in terms of prediction precision with multilayer perceptron artificial neural network (MLP-ANN) Decision Tree, SVM, Random Forest and, Naïve Bayes algorithms. An input to these algorithms is passed as a transformed trend deterministic NIFTY dataset for 20 years and performance is compared accordingly. Rest of the study is organised into following sections. Section states research data, technical indicators and ensemble based ranking of technical indicators which acts as an input to the proposed History Bits algorithm.

3 | RESEARCH DATA

We have utilised NIFTY 50 list dataset of National Stock Exchange: India for experimentation work utilising 75 technical indicators belonging to various categories like moving averages, trend deterministic, volume based, Bollinger Bands, momentum etc. to avoid redundancy and have a fair result. These indicators are shown in Table 1. We can utilise the dataset of any organisation recorded National or Bombay Stock Exchange that is, BSE or NSE. for technical analysis given that in contains esteems: date, open, high, low, close, and volume traded numbers. This informational index is considered as an essential wellspring of information which directly accessible on www.nseindia.com. This directly available dataset is considered as a level 1 dataset. The important step in the preparation of transformed dataset is to compute technical indicators. For this computation, we have used Financial and Technical Analysis library which is available on https://github.com/leomrocha/finta. The dataset containing the statistical values of 75 technical indicators is called as layer 2 dataset. Layer 2 dataset is converted into layer 3 dataset using indication given by the corresponding technical indicator. Most of the research work on stock market prediction is based on two class prediction: buy and sell. The novelty in our work is we have considered five classes for the prediction: strong_buy, strong_sell, buy, sell, hold to give more accurate trading decision. We have used NIFTY datasets over a time period of

TABLE 1 Technical indicators

Simple moving average	Stochastic oscillator %D	Triple exponential moving average	Double exponential moving average
Simple moving median	Stochastic RSI	Elastic volume moving average	Elastic-volume weighted MACD
Smoothed simple moving average	Williams %R	Bollinger bands width	Market momentum 'MOM'
Exponential moving average	Ultimate oscillator	Pivot points	Rate-of-change
Weighted moving average	Awesome oscillator	Fibonacci pivot points	Relative strength index RSI
Hull moving average	Mass index	Volume flow indicator	Inverse Fisher transform
Triangular moving average	Vortex indicator	Average directional index	True range
Triple exponential moving average oscillator	Know sure thing	Volume weighted average price	Average true range
Volume adjusted moving average	True strength index	Smoothed moving average	Stop-and-reverse
Kaufman efficiency indicator	Typical price	Convergence divergence	Money flow index
Kaufman's adaptive moving average	Accumulation	Percentage price	On balance volume (OBV)
Zero lag exponential moving average	Chaikin oscillator	Volume-weighted	Weighted OBV
Wave trend oscillator	Volume price trend	Bull power and bear power	Buy and sell pressure (BASP)
Moving standard deviation	Volume zone oscillator	Ease of movement	Normalised BASP
Ichimoku cloud	Price zone oscillator	Commodity channel index 'CCI'	Chande momentum oscillator
Vector size indicator	Elder's force index	Coppock curve	Chandelier exit
Squeeze momentum indicator	Cumulative force index	Twiggs money index	Qstick
Directional movement indicator	Stochastic oscillator %K	Adaptive price zone	Finite volume element
Fisher transform	Bollinger bands		

20 years (June 1999–November 2019) which are stored annually on a private blockchain to avoid data manipulation, for validating the prediction results of the History Bits algorithm. Private blockchain offers quick retrieval of data which is stored annually and ensures secure transmission of the same. Storage of data on blockchain ensures immutability of the data, that is, a piece of information in a database that cannot be deleted or modified hence further reassuring no data manipulation. Considering dataset over different time frames will also prove the fitness and avoid biasness of the proposed algorithm towards dataset of particular a time frame.

4 | PROPOSED WORK—HISTORY BITS ALGORITHM

In this study, we put forward an innovative algorithm based on history bits with the aim of formulating a maximum profitable trading model. The proposed algorithm takes an input from the stock market's financial historical data, which is converted to trend deterministic buy or sell calls for each of the 75 indicators, depending upon respective statistical values and its interpretation as given in Table 4. Most of the researchers have solved the stock market prediction problem using two classes of prediction—buy and sell. We have considered 5 classes namely strong buy, buy, strong sell, sell and hold for prediction using wisdom of crowd mechanism. This mechanism is quite simple but strong for deciding the outcome using majority based voting scheme. It works on the principle—'A large population of uncorrelated features as a group will outperform the opinion given by any individual feature'. Uncorrelated features come together to give an opinion which is far accurate than the cumulative of its parts. These uncorrelated features collectively make ensemble based predictions that are far better than any of the predictions made by the individual feature.

Algorithm Preparation of Transformed Dataset

```
Input: dataset containing buy/sell call for each of the 75 indicators
Output: A transformed dataset with the class label-strong_buy, strong_sell, buy, sell, and hold
for $i$ = 1 to $n$ do ($n$ number of trading days)
if ≥75% indicators generate buy/sell then signal strong_buy/strong_sell
if >55% && <75% indicators generate buy/sell then signal buy/sell
if ≥45% indicators && ≤55% indicators generate buy/sell signal then hold.
end for
```

In this paper, we have considered 75 technical indicators from which buy or sell calls are generated based on their statistical values. Some of the indicators may produce accurate calls whereas some may produce false calls. The main reason for using wisdom of crowd approach is that these indicators help each other in minimising their individual errors. Based on this approach we have generated five classes of prediction using following approach.

If 60 indicators out of 75 generate buy or sell call for the next trading day then we predict the trading decision for the next day as strong_buy or strong_sell respectively. If 37 indicators give trading call as buy and 38 indicators give trading call as sell or vice versa then we predict trading call for the next day as hold. If 39 to 59 indicators generate buy or sell call then we predict the trading decision as buy or sell respectively. Here some indicators may generate false buy or sell calls and others may generate accurate buy or sell calls. So together as a team these indicators will be able to move in the right direction giving the confidence to the investors for taking the trading decision.

The main idea behind the working of the history bits algorithm is to prioritise the 75 technical indicators into number of groups using attribute ranking strategy. There are number of attribute ranking strategies like information gain, correlation coefficient, gain ratio, OneR, relief evaluator, symmetric uncertainty evaluator etc. are available. Also in our experimentation we have considered the NIFTY 50 dataset over different periods of time. Using any particular attribute ranking strategy would create a biasing effect on deciding the ranks of the indicators [20]. To minimise this biasing effect we have developed an ensemble based ranking strategy which will take into account the combined outcomes in terms of rankings given by different attribute ranking algorithms. This ensemble based ranking strategy would minimise the error produced by the particular individual ranking strategy.

Once technical indicators are prioritised using ranks, history bits algorithm categorises these indicators into number of groups based on their priorities. We have considered five groups so that every group will contain 15 indicators in each group. After grouping of the indicators according to their priorities, weights are assigned to the indicators based on their groups.

Step 1 —Higher the rank of the indicator more is the weight assigned to it.

Initially weights are assigned to the indicators using Equation (1) which is the same equation that is used to assign weights to the features in perceptron learning.

$$y = w * x + b \tag{1}$$

$$y = \varphi\left(\sum\nolimits_{i=1}^{n} w_i x_i + b\right) \tag{2}$$

$$y = \varphi\left(w^T + b\right) \tag{3}$$

However, we do not update the weights assigned in any case and then proceed to step 2.

Step 2 —For every group if number of 'buy' signals are more than 'sell' signals then the history bit corresponding to that group is set as 1 else it is set as 0.

Step 2 may produce ambiguity and thus resulting into wrong prediction of the trading decision caused by a small fraction of deviation. For example, the relief evaluators extract the indicator ranking which in turn is used to generate bit pattern. These indicators are then categorised into five groups representing an individual bit of the 5-bit pattern. The indicators in the first group are all rated above the highest rated indicator in second group. This creates a case of ambiguity where the boundary-case prevails as the law breaker. It can be explained by very rare worst-case scenario, when indicators having 'buy' calls are just greater than those having 'sell' calls. In such case, the majority calls of the subsequent group prevail and are taken into consideration while developing the model. Though these boundary-cases are very rare, if one exists, it will have minimum impact on algorithm due to the 'Average approach' considered while experimenting with the algorithm's accuracy.

From the pattern generated while grouping the indicators, it is observed that, for the worst case, <20% of the indicators in each of the first three groups have call opposite to that of calls of majority of the indicators.

Step 3 —Collect the bit pattern generated from all the groups

Step 4 Bit pattern will have five bits as we have five groups in which indicators are categorised. The interpretation of the bit pattern after resolving the ambiguity is given in Table 2.

The five bits represent 75 technical indicators divided into group of 15 each. The more powerful of higher ranked one are kept on most significant position while the least one are kept on least significant position. The grouping is done on based upon attribute ranking strategies. Moreover, looking at the output of bit pattern the majority of 1's on most significant side states the strong buy or buy while more number of 0's on most significant side states sell or strong sell. The bit value (0 or 1) is solely decided by group of technical indicators. If majority of technical indicators in group says sell then the bit is set to 0 else if the majority is with buy, then it is set to 1.

5 | EXPERIMENTAL WORK

Inside this research work, we pitch a novel History Bit based algorithm for stock market prediction. For experimental work, we will be using NIFTY 50 dataset of the Indian stock market index for the 20 years having 4977 instances. Table 3 shows the example of NIFTY 50 dataset available on Internet. This dataset is then used to calculate technical indicators as shown in Table 4.

Technical indicators indicate the trend of the stock using numerical values. Trend determined by these indicators helps us to transform the numerical values into buy and sell indication as mentioned in Table 5.

We then apply wisdom of crowd approach to make this prediction problem as five class prediction where we use

TABLE 3 Layer 1 dataset (sample)

Date	Open	High	Low	Close
15-Nov-18	10,580.6	10,646.5	10,557.5	10,616.7
16-Nov-18	10,644	10,695.15	10,631.15	10,682.2
19-Nov-18	10,731.25	10,774.7	10,688.8	10,763.4
20-Nov-18	10,740.1	10,740.85	10,640.85	10,656.2

TABLE 4 Layer 2 dataset (sample)

Date	EMA	VWAP	TP	Pivot
15-Nov-18	10,616.7	10,606.9	10,606.9	–
16-Nov-18	10,653.09	10,640.45	10,669.5	10,606.9
19-Nov-18	10,698.3	10,670.85	10,742.3	10,669.5
20-Nov-18	10,684.04	10,672.91	10,679.3	10,742.3

Abbreviations: EMA, Exponential Moving Avg.; TP, True price; VWAP, Volume Weighted Avg. Price.

TABLE 2 Interpretation of the bit patterns

Strong buy	Buy	Hold	Sell	Strong sell
['1', '1', '1', '1', '1']	['0', '1', '1', '1', '1']	['0', '1', '1', '1', '0']	['0', '1', '0', '0', '1']	['0', '0', '0', '1', '1']
['1', '1', '1', '1', '0']	['1', '1', '0', '1', '0']	['1', '0', '1', '0', '1']	['0', '0', '1', '1', '0']	['0', '1', '0', '0', '0']
['1', '1', '1', '0', '1']	['1', '0', '1', '1', '0']	['0', '1', '1', '0', '1']	['0', '0', '1', '0', '1']	['0', '0', '1', '0', '0']
['1', '1', '0', '1', '1']	['1', '1', '0', '0', '1']	['1', '0', '0', '1', '1']	['1', '0', '0', '0', '0']	['0', '0', '0', '1', '0']
['1', '0', '1', '1', '1']		['0', '1', '0', '1', '1']		['0', '0', '0', '0', '1']
['1', '1', '1', '1', '1']		['0', '0', '1', '1', '1']		['0', '0', '0', '0', '0']
['1', '1', '1', '1', '0']		['1', '1', '0', '0', '0']		
['1', '1', '1', '0', '0']		['1', '0', '1', '0', '0']		
		['1', '0', '0', '1', '0']		
		['1', '0', '0', '0', '1']		
		['0', '1', '1', '0', '0']		
		['0', '1', '0', '1', '0']		

TABLE 5 Layer 3 dataset (sample)

Date	EMA	VWAP	TP	Pivot
15-Nov-18	Sell	Buy	Sell	–
16-Nov-18	Buy	Buy	Sell	Buy
19-Nov-18	Buy	Buy	Sell	Buy
20-Nov-18	Sell	Sell	Buy	Sell

classes—strong_buy, buy, strong_sell, sell, hold for prediction. We have also compared the prediction performance of proposed algorithm with, SVM Random Forest, Multilayer Perceptron (MLP), Naïve Bayes, Decision Tree and Artificial Neural Network which are well known and widely used machine learning algorithms.

5.1 | Decision Tree

It uses the tree structure where internal nodes of the tree represent attributes and each leaf node represents the prediction class labels. Following are some assumptions made when using Decision Tree.

- Initially the whole training dataset is considered as root.
- Decision Tree prefers the categorical feature values. Continuous feature values are discretised before the construction of the model.
- Ordering of the attributes as internal node or root node is performed using attribute selection techniques like information gain and gini index.

Partitioning of the training instances in the decision tree is performed at the node level. These partitioning changes the entropy and this change in entropy represents the information gain. The entropy is used to calculate the Information gain of a class label. Depending on the value of Information gain the splitting node in a decision tree is decided, the labels with higher information gain are used as the splitting node.

Information gain is calculated using Equation (1).

$$\mathrm{Gain}(I,A) = \mathrm{Entropy}(I) - \sum\nolimits_{x=0}^{n} V(A)\cdot E(I_x I_x) \tag{4}$$

where I is a set of instances, A is a set of attributes, x indicates values that attributes hold and $I_x I_x$ is a subset of I.

Entropy measures the uncertainty of a particular attribute. More the entropy more is the information gain.

5.2 | Naïve Bayes

It assumes that every feature is independent of each other. Each feature equally and independently contributes to the outcome. Given the probability of an event B that has already occurred Bayes theorem calculates the probability of an event A.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{5}$$

where $P(B)$ is constant.

Here, we are trying to find probability of A given the probability of B. Hence B is known as evidence. Probability of (A) is known as prior probability. It is the probability of the event A before conducting any event. Probability of $(A\,|\,B)$ is known as posterior probability. It is the probability of the event A after conducting the event B.

Consider Y as the outcome variable and X is a feature set where,

$$x = \{\, x_1, x_2, x_3, \ldots, x_n \,\}$$

According to Bayes theorem,

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{6}$$

$$p(y|x_1, x_2, \ldots, x_n) = \frac{P(x_1(y)p(x_2|y)\ldots PCy)}{p(x_1, P(x_2)\cdots P(x_n)} \tag{7}$$

As denominator remains constant, Equation (4) can be rewritten as,

$$p(y|x_1, x_2, \ldots, x_n)\alpha P(Y)\Pi_{i=1}^{n} P(X_i|Y)(Y) \tag{8}$$

To construction a Naïve Bayes prediction model for all the possible outcomes, we need to calculate the probability for input set of features and take the maximum probability

$$Y = \mathrm{argmax}_y P(y)\Pi_{i=1}^{n} P(X_i, Y) \tag{9}$$

Probability (Y) is probability of class and $(X_i\,|\,Y)$ is conditional probability [10].

5.3 | Multilayer Perceptron-ANN

Perceptron is a linear classification algorithm that classifies the instances into two classes by a straight line [19].

$$y = \mathrm{w} * \mathrm{x} + \mathrm{b} \tag{10}$$

x is a feature vector which is multiplied by weights w and bias b is added. It performs a linear combination using its feature vectors and weight along with non-linear activation function. These activation functions present the non-linear relationship of input and output [10].

$$y = \varphi\left(\sum\nolimits_{i=1}^{n} w_i x_i + b\right) \tag{11}$$

$$y = \varphi\left(w^T + b\right) \tag{12}$$

Here, **w** represents the vector of weights, **x** is the feature vector, **b** is the bias and phi (φ) is the non-linear activation function.

Whenever there is any error in classification, weights are updates using Equation (10).

$$W = W + 1 * (\text{actual} - \text{predicted}) * x \tag{13}$$

where l is the learning rate of the classifier.

MLP is a neural network consisting of n layers of perceptron. It has a nodal structure with at least trio of layers: mainly input, hidden and output layer. Number of hidden layers can be increased as per the task and model can be made more complex. MLP uses backpropagation supervised learning technique for learning. Every node in MLP is a neuron with non-linear activation function which can distinguish non-linearly separable data. Given a classifier function: $y = f(x)$ MLP performs a mapping $y = f(x; \Theta)$ to best approximate the classifier and learn best parameters $\boldsymbol{\Theta}$. All the layers in MLP are fully connected. Parameters of each layer are independent and hence they have a unique weight. MLP classifier's performance is measured using loss function. Lower is the loss when the predicted class is same as that of actual class. In MLP initially weights are assigned with some random values and then they are modified to get a lower loss. Learning rate is the rate at which algorithm performs its iterations to minimise the loss. It controls the amount of change in the model in responding to the estimated error when weights are modified. MLP classifier model is trained using three steps. The first step is known as forward pass, shown in Figure 1, where we pass the input to the classifier, multiply the input with weights and add bias to calculate the output as given in Equations (11) and (12).

The second step in training is known as loss calculation. The output calculated in forward pass is known as forecasted output which is then compared with expected output. The difference between the expected output and predicted output gives the loss incurred. The third step is backward pass in which weights are updated using gradient to minimise the loss.

5.4 | Random Forest

Random forest is supervised or regulated learning technique based on creating the no of decision trees on given data samples which are of same size that of original training dataset. The prediction result of each decision tree is collected and finally the best result is selected by using majority based call or by voting. This process is known as bootstrap aggregation or bagging. This is ensemble based technique which proves to be much better than single decision tree as it average outs the prediction results by minimising the over-fitting problem.

In bagging, the training dataset is not replaced by the smaller subsets. If we consider the size of the training dataset as N then we have random samples of size N with replacement passed as an input to the individual decision tree. Therefore each tree which is part of random forest picks a random sample of size N and predicts the outcome. This creates more diversification among multiple trees in the model resulting into lower correlation between trees.

5.5 | Support Vector Machine

SVM is built on the idea of creating the decision line which is known as the hyperplane in N-dimensional place where N is number of features, so that data points can be correctly separated. Hyperplane created by SVM should best divide the data points into two classes. To create this hyperplane SVM uses the extreme data points known as support vectors. The distance between the hyperplane and these support vectors is known as the margin. There could be multiple hyperplanes possible that can segregate the datapoints. SVM chooses the hyperplane with maximum distance between the hyperplane and support vectors. This hyperplane is known as optimal hyperplane. Margin can be maximised by using support vectors. If the given datapoints can be classified into two classes linearly then these data points are known to be linearly separable and SVM classifier is known as linear SVM. Non-linear

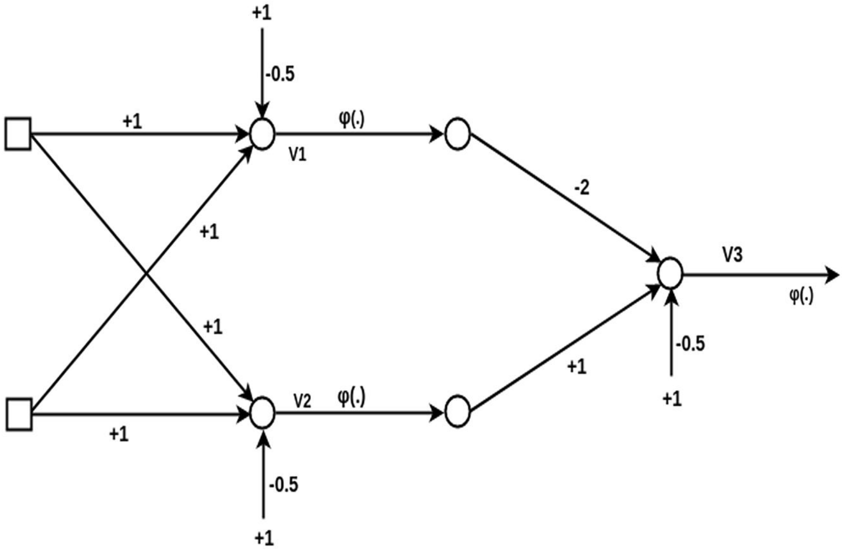

FIGURE 1 Forward pass in multilayer perceptron artificial.

SVM separates the data points that cannot be separated in a linear way using kernel functions. Classification of data points in SVM is based on linear function. If the output of the linear function is >1, we identify that data point belongs to one class and if output is less than −1 then data point belongs to another class. Therefore, reinforcement range for SVM becomes [−1, +1] which acts as margin.

SVM always looks to maximise the margin in middle of the supporting vectors and the hyperplane. This is achieved using the loss function given in Equations (14) and (15).

if $y \times f(x) \geq 1$

$$c\,(x,\ y,\ f(x)) = 0 \tag{14}$$

else,

$$c\,(x,\ y,\ f(x)) = 1 - y * f(x) \tag{15}$$

If the predicted value and expected value matches then cost is 0. If they do not match then we have to calculate the loss and the cost function and regularisation parameter to balance the loss value and the margin maximisation.

$$\min \lambda|w|^2 + \sum\nolimits_{i=1}^{n} (1 - y_i\langle x_i \mu\rangle) \tag{16}$$

Gradients are then calculated by taking partial derivatives over the weights and then the weights are updated using these gradients

$$\frac{\delta}{\delta w_k}\lambda|w|^2 = 2\lambda w_k \tag{17}$$

$$\frac{\delta}{\delta w_k}(1 - y_i\langle x_i, w\rangle) = 0 \ if\ y_i\langle x_i, w\rangle \geq 1 = -y_i x_{ik} \tag{18}$$

When SVM correctly classifies the gradient will be updated using regularisation parameter.

$$w = w - \alpha, (2\lambda w) \tag{19}$$

In case of misclassification, the gradient is updated by using the loss and the regularisation parameter.

$$w = +\alpha, \left(y_i x_i - 2\lambda w\right) \tag{20}$$

6 | RESULT AND DISCUSSION

Here in this indicated analysis we have put forward a novel algorithm-based history bits for the prediction of stock market trends. We have compared the performance and prediction precision of the planned algorithm with some of the well-known machine learning algorithms—decision tree, naïve Bayes, MLP-ANN, random forest and SVM. Confusion matrix contains the entries of the instances that are correctly predicted and incorrectly predicted. As we have 5 classes of prediction, the size of the confusion matrix becomes 5 × 5. The dataset used for the experimentation is NIFTY 50 index of the Indian stock market over different time frames. This dataset is divided into training and testing dataset with percentage split of 70:30. 70% of the instances are used as training set and 30% of the instances are used as testing set.

True positives: If the output from a forecast/prediction and the real value is a 'hit', then it's known as true positive.

$$\text{True Positive Rate (TPR)} = \frac{\text{No of correct positive predictions}}{\text{Total no. of positives}}$$

False positives: If the output from a forecast/prediction and real value is a 'miss', then it's known as false positive.

$$\text{False Positive Rate (FPR)} = \frac{\text{No of correct negative predictions}}{\text{Total no. of negatives}}$$

Precision is the number of acquired specimen which are relevant.

$$Precision = \frac{\text{No. of true positives for the class}}{\text{Total no. of true positives + total no of false positives}}$$

*F*_measure used for calculating the efficiency of a classification algorithm in means of precision and recall (recall is same as that of true positive rate).

$$F - \text{Measure} = \frac{2 * P * \text{TPR}}{P + \text{TPR}}$$

Matthew's correlation coefficient (MCC) measures the strength of the classification problem. MCC value is bounded from −1 to +. −1 indicates a comprehensive divergence between actual and predicted value, 0 indicates arbitrary prediction, and +1 shows the ideal prediction. The mathematical formula to calculate MCC is:

$$\text{MCC} = \frac{\text{TP} * \text{TN} - \text{FP} * \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

6.1 | Performance evaluation of Decision Tree

Performance summary of the Decision Tree algorithm is given in Table 6.

Confusion matrix for the Decision Tree algorithm is given in Table 7.

Detailed accuracy for the Decision Tree algorithm is given in Table 8.

6.2 | Performance evaluation of Naïve Bayes

Performance summary of the Naïve Bayes algorithm is given in Table 9.

Confusion matrix for the Naïve Bayes algorithm is given in Table 10.

Detailed accuracy for the Naïve Bayes algorithm is given in Table 11.

TABLE 6 Summary of the performance

Total number of instances in test split	1493
Correctly predicted instances	1121
Incorrectly predicted instances	372
Prediction accuracy	75.08%
Kappa statistics	0.6754
MAE	0.1105
RMSE	0.2996

Abbreviations: MAE, Mean Absoluter Error; RMSE, Root Mean Squared Error.

TABLE 7 Decision Tree—confusion matrix

Actual versus predicted	a	b	c	d	e
a = hold	69	30	0	26	0
b = buy	34	313	64	1	0
c = strong_buy	0	98	315	0	0
d = sell	17	4	0	284	61
e = strong_sell	0	0	0	37	140

TABLE 8 Detailed accuracy

	TPR	FPR	Precision	*F*_Score	MCC
Hold	0.552	0.037	0.575	0.563	0.679
Buy	0.760	0.122	0.703	0.730	
Strong_buy	0.763	0.059	0.831	0.795	
Sell	0.776	0.057	0.816	0.796	
Strong_sell	0.791	0.046	0.697	0.741	

TABLE 9 Summary of the performance

Total number of instances in test split	1493
Correctly predicted instances	1064
Incorrectly predicted instances	429
Prediction accuracy	71.26%
Kappa statistics	0.632
MAE	0.1162
RMSE	0.3245

6.3 | Performance evaluation of Multilayer Perceptron

Performance summary of the MLP algorithm is given in Table 12.

Confusion matrix for the MLP algorithm is given in Table 13.

Detailed accuracy for the MLP algorithm is given in Table 14.

TABLE 10 Naïve Bayes—confusion matrix

Actual versus predicted	a	b	c	d	e
a = hold	104	10	0	11	0
b = buy	87	251	74	0	0
c = strong_buy	0	87	325	1	0
d = sell	56	0	0	260	50
e = strong_sell	0	0	0	53	124

TABLE 11 Detailed accuracy

	TPR	FPR	Precision	*F*_Score	MCC
Hold	0.832	0.105	0.421	0.559	0.644
Buy	0.609	0.090	0.721	0.661	
Strong_buy	0.787	0.069	0.815	0.800	
Sell	0.710	0.058	0.800	0.753	
Strong_sell	0.701	0.038	0.713	0.707	

TABLE 12 Summary of the performance

Total number of instances in test split	1493
Correctly predicted instances	1135
Incorrectly predicted instances	358
Prediction accuracy	76.02%
Kappa statistics	0.687
MAE	0.1002
RMSE	0.285

TABLE 13 Multilayer Perceptron—confusion matrix

Actual versus predicted	a	b	c	d	e
a = hold	78	28	0	19	0
b = buy	29	306	77	0	0
c = strong_buy	0	75	338	0	0
d = sell	23	2	0	288	53
e = strong_sell	0	0	0	62	125

6.4 | Performance evaluation of Random Forest

Performance summary of the Random Forest algorithm is given in Table 15.

Confusion matrix for the Random Forest algorithm is given in Table 16.

Detailed accuracy for the Random Forest algorithm is given in Table 17.

TABLE 14 Detailed accuracy

	TPR	FPR	Precision	*F*_Score	MCC
Hold	0.624	0.038	0.600	0.612	0.690
Buy	0.743	0.097	0.745	0.744	
Strong_buy	0.818	0.071	0.815	0.816	
Sell	0.787	0.063	0.802	0.794	
Strong_sell	0.706	0.040	0.702	0.704	

TABLE 15 Summary of the performance

Total number of instances in test split	1493
Correctly predicted instances	1201
Incorrectly predicted instances	292
Prediction accuracy	80.44%
Kappa statistics	0.743
MAE	0.133
RMSE	0.2428

TABLE 16 Random Forest—confusion matrix

Actual versus predicted	a	b	C	d	e
a = hold	63	32	0	30	0
b = buy	13	337	61	1	0
c = strong_buy	0	54	359	0	0
d = sell	14	0	0	305	47
e = strong_sell	0	0	0	40	137

TABLE 17 Detailed accuracy

	TPR	FPR	Precision	*F*_Score	MCC
Hold	0.504	0.020	0.700	0.586	0.746
Buy	0.818	0.080	0.797	0.807	
Strong_buy	0.869	0.056	0.855	0.862	
Sell	0.833	0.063	0.811	0.822	
Strong_sell	0.774	0.059	0.745	0.759	

6.5 | Performance evaluation of SVM

Performance summary of the SVM algorithm is given in Table 18.

Confusion matrix for the SVM algorithm is given in Table 19.

Detailed accuracy for the SVM algorithm is given in Table 20.

6.6 | Performance evaluation of History Bits algorithm

For the proposed algorithm, there is no training-testing split of the dataset and we consider all the instances as unknown instances.

Performance summary of the History Bits algorithm is given in Table 21.

Confusion matrix for the History Bits algorithm is given in Table 22.

TABLE 18 Summary of the performance

Total number of instances in test split	1493
Correctly predicted instances	1174
Incorrectly predicted instances	319
Prediction accuracy	78.63%
Kappa statistics	0.720
MAE	0.2488
RMSE	0.3298

TABLE 19 Support Vector Machine—confusion matrix

Actual versus predicted	a	b	c	d	e
a = hold	73	24	0	28	0
b = buy	22	318	71	1	0
c = strong_buy	0	63	350	0	0
d = sell	12	1	0	299	54
e = strong_sell	0	0	0	43	134

TABLE 20 Detailed accuracy

	TPR	FPR	Precision	*F*_Score	MCC
Hold	0.584	0.025	0.682	0.629	0.723
Buy	0.772	0.081	0.783	0.778	
Strong_buy	0.847	0.066	0.831	0.839	
Sell	0.817	0.064	0.806	0.811	
Strong_sell	0.757	0.041	0.713	0.734	

TABLE 21 Summary of the performance

Total number of instances	4977
Correctly predicted instances	4348
Incorrectly predicted instances	629
Prediction accuracy	87.36%
Kappa statistics	0.788
MAE	0.2835
RMSE	0.848

TABLE 22 History Bits—confusion matrix

Actual versus predicted	a	b	c	d	e
a = buy	13	21	0	106	0
b = hold	32	328	20	127	15
c = sell	1	78	14	0	20
d = strong_buy	6	9	0	2593	0
e = strong_sell	0	177	17	0	1400

TABLE 23 Detailed accuracy

	TPR	FPR	Precision	*F*_Score	MCC
Buy	0.092	0.008	0.251	0.140	0.790
Hold	0.628	0.063	0.545	0.588	
Sell	0.123	0.007	0.273	0.173	
Strong_buy	0.994	0.098	0.922	0.950	
Strong_sell	0.878	0.010	0.988	0.980	

Detailed accuracy for the History Bits algorithm is given in Table 23.

From the Figure 2, it is clear that the proposed history bits algorithm outperforms other machine learning algorithms in the prediction task.

7 | FUTURE SCOPE & CONCLUSION

Stock forecasting has been the dream of investors since its existence. This paper addresses the hassle of determining the trend of the stocks and thus helping investors to take sharp decision on whether to buy or sell particular stocks. Numerous machine learning strategies are used to perform the prediction of the stock trends and help traders to make accurate decision of buying or selling the stocks. In this paper, we have proposed a novel algorithm based on history bits for the prediction of trading decision. For the implementation and testing of the History Bits based algorithm, 75 technical parameters are computed using stock trading data (open, high, low & close prices), prioritised using ensemble-based attribute ranking strategy where we combined the ranking results from attribute ranking strategies to decide the final ranking which acts as input for the proposed algorithm. From the results it is clear that our proposed algorithm based on history bits shows better prediction accuracy than other well-known machine learning algorithms such as Decision Tree, Naïve Bayes, MLP, SVM, Random Forest We have used five classes of prediction strong_buy, strong_sell, buy, sell and hold for helping traders to take more accurate decisions. However, it is interesting to see how the proposed algorithm works for the fundamental analysis of the stocks. Also, we have considered ensemble

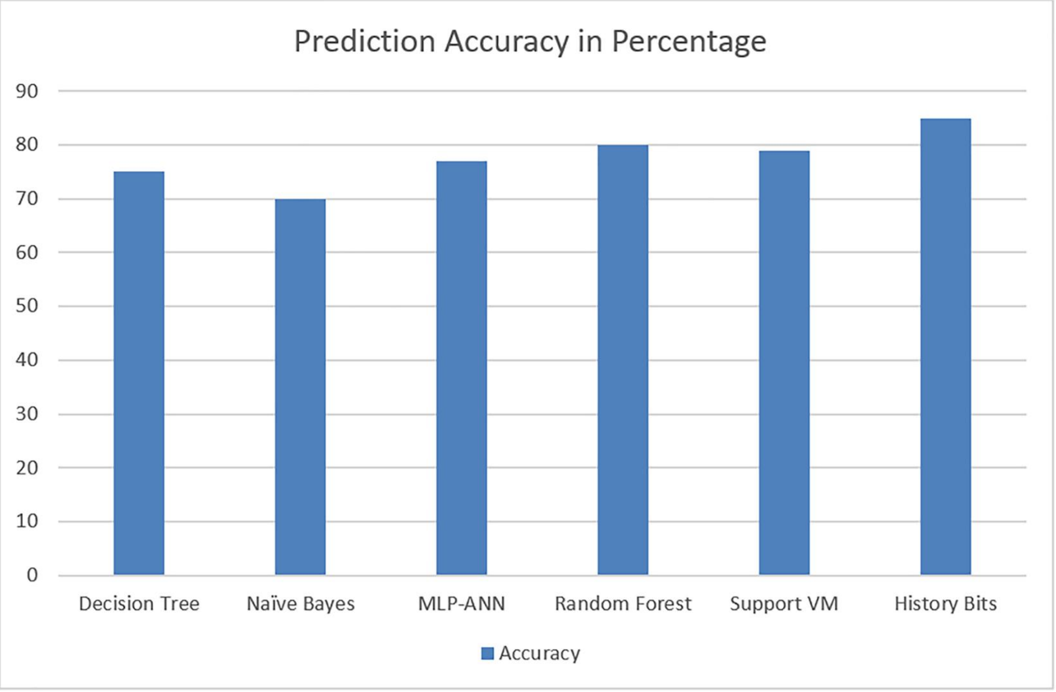

FIGURE 2 Prediction accuracy comparison of History Bits with other algorithms.

based approach for deciding the final rankings of the technical indicators. Also we have implemented data governance using secure decentralised private blockchain. It is important to analyse the prediction result of the proposed history bits algorithm with different attribute ranking strategies like information gain, correlation coefficient, OneR, relief evaluator, gain ratio, symmetric uncertainty evaluator. History bits algorithm unlike other algorithms we considered in this paper is unsupervised algorithm and can also act as an input to the other machine learning algorithms.

AUTHOR CONTRIBUTIONS

Nitin Nandkumar Sakhare: Conceptualisation; Methodology; Project administration; Software; Writing – original draft; Writing – review & editing. **Imambi S. Shaik**: Formal analysis; Supervision; Validation; Writing – review & editing. **Suman Saha**: Data curation; Investigation; Resources.

ACKNOWLEDGEMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

CONFLICT OF 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 manuscript.

DATA AVAILABILITY STATEMENT

Data/datasets related to the research experiment can be made available upon request to the corresponding author.

ORCID

Nitin Nandkumar Sakhare https://orcid.org/0000-0002-1748-799X
Imambi S. Shaik https://orcid.org/0000-0003-0600-6959
Suman Saha https://orcid.org/0000-0003-4226-5966

REFERENCES

1. Tsai, C., Quan, Z.: Stock prediction by searching for similarities in candlestick charts. ACM Transaction on Management Information System 5(2), 1–21 (2014). https://doi.org/10.1145/2591672
2. https://www.investopedia.com/terms
3. Minh, D.L., et al.: Deep learning approach for short term stock trends prediction based on two-stream gated recurrent unit network. IEEE Access 6, 55392–55404 (2018). INSPEC Accession Number: 18159342. https://doi.org/10.1109/ACCESS.2018.1868970
4. Barmish, B.R., Primbs, J.A.: On a new paradigm for stock trading via a model- free feedback controller. IEEE Trans. Automat. Control 6(3), 662–676 (2016). INSPEC Accession Number: 15806153. https://doi.org/10.1109/TAC.2015.2444078
5. Huynh, H.D., Dang, L.M., Dhong, D.: A new model for stock price movement prediction using deep neural network. In: ACM, Proceedings of the 8th ISICT, pp. 57–62 (2017). https://doi.org/10.1145/3155133.3155202
6. Li, X., et al.: Empirical analysis: stock market prediction via extreme learning machine. Neural Comput. Appl. 27(1), 67–78 (2016). https://doi.org/10.1007/s00521-014-1550-z
7. Sakhare, N., Imambi, S.: Performance analysis of regression-based machine learning techniques for prediction of stock market movement. IJRTE 7(6S) (2019)
8. Patel, J., et al.: Predicting stock market index using fusion of machine learning techniques. Expert Syst. Appl. 42(4), 1–11 (2014). https://doi.org/10.1016/j.eswa.2014.10.031
9. Sakhare, N., Joshi, S.: Classification of criminal data using J48 algorithm. Int. J. Data Warehous. Min. 4, 167–171 (2014)
10. Ocharoem, P., Vateekul, P.: Deep learning using risk-reward function for stock market prediction. In: ACM, Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, pp. 556–561 (2018). https://doi.org/10.1145/3297156.3297173
11. Yetis, Y., Kaplan, H., Jamshedi, M.: Stock market prediction by using artificial neural network. In: IEEE, World Automation Congress (2014). INSPEC Accession Number: 14700939. https://doi.org/10.1109/WAC.2014.6936118
12. Sharma, N., Juneja, A.: Combining of random forest estimates using LSboost for stock market index prediction. In: 2017 2nd International Conference for Convergence in Technology (2017). INSPEC Accession Number: 17449287. https://doi.org/10.1109/I2CT.2017.8226316
13. Joshi, S., Sakhare, N.: History Bits based novel algorithm for classification of structured data. In: 2015 IEEE International Advance Computing Conference (IACC), pp. 609–612. Banglore (2015). INSPEC Accession Number: 15292807. https://doi.org/10.1109/IADCC.2015.7154779
14. Feng, F., et al.: Temporal relational ranking for stock exploration. ACM Trans. Inf. Syst. 37(2), 1–30 (2019). https://doi.org/10.1145/3309547
15. Bousono-Calzon, C., et al.: Expert selection in prediction market with homological invariants. IEEE Access 6, 32226–32239 (2018). INSPEC Accession Number: 17905708. https://doi.org/10.1109/ACCESS.2018.2846878
16. Waqar, M., et al.: Prediction of stock market by principle component analysis. In: 2017 13th International Conference on Computational Intelligence and Security (CIS). INSPEC Accession Number: 17578286. https://doi.org/10.1109/CIS.2017.00139
17. Wen, M., et al.: Stock market trend using high order information of time series. IEEE Access 7, 28299–28308 (2019). INSPEC Accession Number: 18521046. https://doi.org/10.1109/ACCESS.2019.2901842
18. Cailee, J.L., et al.: Stock picking by probability-possibility approaches. IEEE Trans. Fuzzy Syst. 25, 333–349 (2017). INSPEC Accession Number: 16776016. https://doi.org/10.1109/TFUZZ.2016.2574921 2
19. Chang, E., Han, C., Frank, C.: Deep learning networks for stock market analysis and prediction: methodology, data representation, and case studies. Expert Syst. Appl. 83, 187–205 (2017). https://doi.org/10.1016/j.eswa.2017.04.030
20. Pehlivanli, A.C., Guzhan, A., Gulay, G.: Indicator selection with committee decision of filter methods for stock market price trend in ISE. Appl. Soft Comput. 49, 792–800 (2016). https://doi.org/10.1016/j.asoc.2016.09.004

How to cite this article: Sakhare, N.N., Shaik, I.S., Saha, S.: Prediction of stock market movement via technical analysis of stock data stored on blockchain using novel History Bits based machine learning algorithm. IET Soft. 1–12 (2023). https://doi.org/10.1049/sfw2.12092