Innovations in Systems and Software Engineering (2024) 20:121–135
https://doi.org/10.1007/s11334-021-00428-0

S.I. : MULTIFACETED INTELLIGENT COMPUTING SYSTEMS (MICS)

A Black Swan event-based hybrid model for Indian stock markets' trends prediction

Samit Bhanja[1,2] · Abhishek Das[2]

Received: 16 May 2021 / Accepted: 19 December 2021 / Published online: 7 January 2022

Abstract

Among all the application areas of the time-series prediction, stock market prediction is the most challenging task due to its dynamic nature, and dependency on many volatile factors. The unpredictable fatal events called Black Swan events also highly influence the stock market. If the successful stock trends prediction is achieved, then the investors can adopt a more appropriate trading strategy, and that can significantly reduce the risk of investment. In this work, a time-efficient hybrid stock trends prediction framework(HSTPF) is proposed to successfully predict the future trends of the stock market even during the periods of Black Swan events. Here, to improve the prediction accuracy of HSTPF, the Black Swan events analysis and features selection operations are performed, and also the performance of various machine learning classifiers are analyzed. A vast number of experiments are conducted on the two real-world stock market datasets S&P BSE SENSEX and Nifty 50, to analyze the performance of the proposed framework. The framework is applied for the single-step and multi-step ahead predictions. The experimental results show that the proposed framework produces over 86% of accuracy, and during the Black Swan events, its accuracy is almost 80% for single-step ahead predictions. For the multi-step ahead of predictions, the HSTPF is produced satisfactory results. The framework also outperforms other existing similar works even during the Black Swan events in terms of prediction accuracy, and its computational time is also very low.

Keywords Stock trends prediction · Deep learning · Machine learning · Technical indicators · Black Swan event

1 Introduction

Prediction is one of the biggest challenges for any research area. If a successful prediction is achieved, then that can smoothen the livelihood of the people. One of the most challenging time-series prediction research areas is the stock trend prediction because of its nonlinearity and highly dynamic behavior. The stock market is also suffered from unpredictable events that have a serious negative impact on it and is called the Black Swan events. If the successful stock trend is predicted, then the stock market regulators and the investors can develop better trading strategies, which helps them to reduce the risk of the investment and also increase profitability.

In recent times, numerous research work has been done to develop the most appropriate model for the stock market prediction. These stock market predictions have been done through two types of analysis, viz. fundamental analysis and technical analysis. The fundamental analysis [1,2] uses various types of economic and financial information, such as macroeconomic factors and foreign exchange rates. On the other hand, technical analysis [3,4] is done based on the stock price information. Nowadays, machine learning and deep learning are two popular artificial intelligence tools that are widely used for stock prediction. Gaussian Naïve Bayes (GNB), Autoregressive Integrated Moving Average (ARIMA), Support Vector Machine (SVM), Logistic Regression (LR), K-Nearest Neighbor (KNN), Classification and Regression Tree (CART), Linear Discriminant Analysis (LDA), Random Forest Classifier (RF) are the popular machine learning algorithms that are extensively used in various stock market prediction research works [5–7]. On the other hand, Recurrent Neural Networks (RNN),

✉ Abhishek Das
adas@aliah.ac.in

Samit Bhanja
samitbhanja@gmail.com

[1] Government General Degree College, Singur, Hooghly, W.B, India

[2] Aliah University, New Town, Kolkata, India

Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Networks (CNN), Deep Belief Network (DBN), Restricted Boltzmann Machine (RBM), Autoencoders (AE) are widely used deep learning algorithms, and they are successfully applied to predict the stock market [8–11]. Recently, the deep learning methods have become more popular compared to machine learning methods due to their automatic spatiotemporal feature extraction capabilities, whereas to deploy the machine learning method, domain expert knowledge is required for feature extraction. Recent research work [12,13] compares the performance of machine learning approaches with deep learning approaches for time-series forecasting and shows that the deep learning methods outperform machine learning methods for large volume input datasets, whereas the performance of machine learning methods is better when the dataset volume is small. Another major drawback of deep learning methods is their high computational time.

Some of the recent studies [14,15] show that the technical indicators of the stock market play a significate role in stock trends prediction. Although the deep learning-based models produce fair results, they are computationally very slow. Currently, the globalization effect increases the financial flow across the world, and as a result, the stock price of a country is not only influenced by local events but it also influenced by other international events. There are a limited number of efforts [16–18] that have been done in this regard. But they are not considered as the different international Black Swan Events that highly influence the stock price of a company.

In this study, we propose a hybrid stock trends prediction framework (HSTPF) to predict the stock trend of Indian stock market with a higher degree of accuracy and lower computational time. Here, a Black Swan event analysis algorithm is proposed to capture the uncertainty of the stock market. The different technical indicators of the stock market are also considered along with the raw stock market data for better prediction accuracy. The deep learning-based feature extraction phase is introduced to extract the features from the time-series data. The feature extraction phase maps the input sample to the features space for discriminating each pattern of the sample and that helps the classifier to more accurately predict the trends. And a machine learning-based classifier is used to classify the feature vector to predict the stock trends. Here, the performance of various machine learning-based classifiers is compared to select the best classifier to improve the prediction accuracy. The main contribution of this research work are as follows:

- To capture the uncertainty of the stock market, Black Swan Event analysis algorithm is proposed. A deep learning-based impact prediction model is constructed.
- Technical features are extracted from the stock market dataset to improve the prediction accuracy.
- To reduce the dimensionality of the dataset and also to extract features, a CNN-based autoencoder is deployed.
- The performance of different machine learning classifiers is compared to obtain the best classifier for the trend prediction.

The rest of the paper is organized as follows: In Sect. 2, we represent the concise description of the literature we reviewed. The detailed description of our proposed methodology is presented in Sect. 3. In Sect. 4, we describe the experimental process and compare our framework with other state-of-the-art models. Finally, in Sect. 5, the conclusion is drawn.

2 Literature review

In the last few decades, many efforts were made to predict stock market trends. These efforts are mainly classified into fundamental analysis and technical analysis. The fundamental analysis uses different macro-economic factors for the prediction; on the other hand, the technical analysis method [19] uses the historical stock price for the prediction. Later on, these technical analysis methods have been formulated as pattern recognition problems. These pattern recognition problems to predict the stock market are broadly divided into statistical methods, machine learning methods, deep learning methods, and hybrid methods [20–22] that combine two or more different types of methods.

Autoregressive Integrated Moving Average (ARIMA) model is a popular machine learning for forecasting linear time-series data. In [23,24], the ARIMA model was used for short-term stock trends prediction. In [25], the authors proposed the ARIMA-BP model by combining the ARIMA with the backpropagation (BP) neural network model to capture both the linearity and nonlinearity of the stock market time-series data for the stock trends prediction. Here, the authors showed that this ARIMA-BP model generates better prediction accuracy compared to the shallow ARIMA and BP models. In [5], the authors proposed a machine learning framework(MLF) to predict the stock market. They also tried to find out the impact of social media and financial news on the market. They show that the fusion of social media and financial news with the stock market data can improve the prediction performance.

In recent times, with the increased computational power and the availability of large volumes of historical data, deep learning architectures have become one of the popular tools for time-series prediction. A large number of deep learning-based novel research work has been done to predict the stock market trends. In [26], a deep learning model was proposed by combining the CNN and LSTM architectures. Here, the authors used the technical indicators and the textual informa-

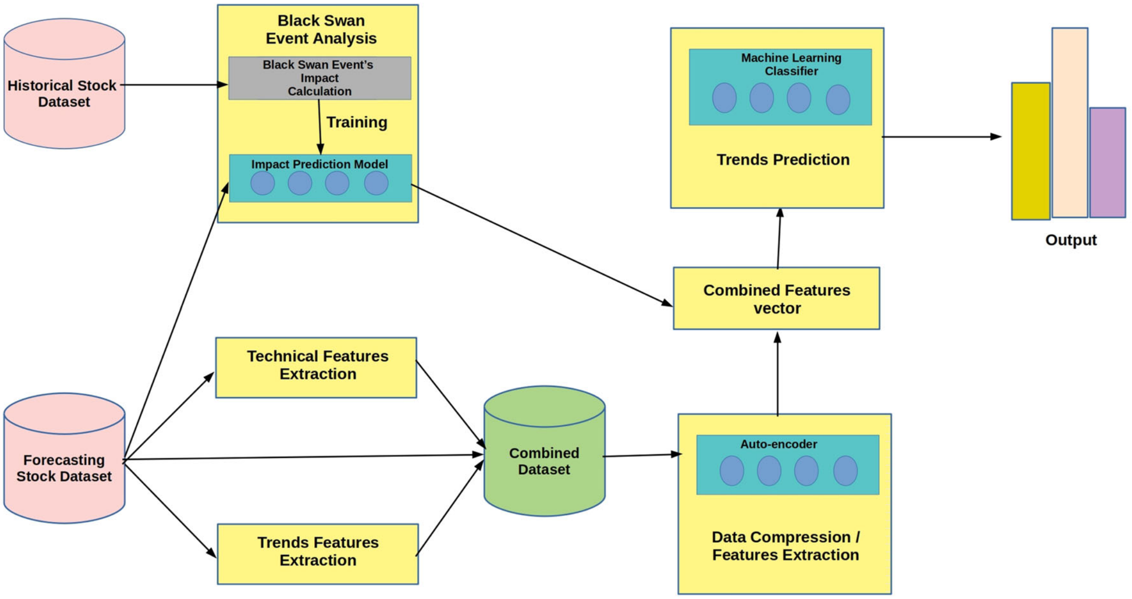
Historical Stock Dataset
Black Swan Event Analysis
Black Swan Event's Impact Calculation
Training
Impact Prediction Model
Machine Learning Classifier
Trends Prediction
Output
Combined Features vector
Technical Features Extraction
Forecasting Stock Dataset
Combined Dataset
Auto-encoder
Data Compression / Features Extraction
Trends Features Extraction

Fig. 1 Architecture of the proposed forecasting system

tion for the stock market prediction to improve the prediction performance. The LSTM model was applied to the large volume of the S&P500 dataset in [27], and it proved that the LSTM model is superior compared to the standard deep neural network architectures. In [10], authors integrated 1D-CNN with LSTM architecture to build a deep learning model and showed that the combined architecture is better than the shallow architectures. The LSTM, RNN, and GRU architectures were compared in [28] to forecast the Google stock price movement. Here, the authors showed that LSTM and GRU are exhibited better prediction accuracy compared to RNN. The authors in [29] proved that the combined model of wavelets and CNN is better than the traditional ANN approaches. The multiple pipeline model was developed in [30] by combining CNN and bi-directional LSTM. Here, it is been established that the multiple pipeline model's performances were increased by 9% over a single pipeline model. The attention-based stock market prediction model (ATN-GRU-M) is proposed in [31]. Here, the authors used the attention layer with the traditional GRU layers, and they also extract the technical features for increasing the prediction accuracy. They show that with the integration of the attention layer with the conventional GRU, the prediction performance is improved. The comparison between the ANN, SVM, Random Forest, and the Naive–Bias was done in [32] based on the two input approaches. The first approach contained ten technical parameters of the stock market, and in the second approach, these technical indicators were represented as the trends deterministic data. Here, the authors showed that the performance is improved with the second input approach. In [33], the authors proposed a motifs extraction algorithm to extract the higher-order structural features from the raw time-series data. These motifs were applied to the CNN model to extract the abstract features to predict stock trends. The authors also showed that the model improves the prediction accuracy compared to the other traditional methods.

However, most of the recent work considered the stock market prediction as a pattern recognition problem, and to improve the prediction accuracy they concentrated on the selection of the different types of machine learning, deep learning, and/or statistical models. Some of the research work was concentrated on the analysis of the various technical indicators and sentiments of the tweeter and news data that are related to the stock market. Differentiating from the existing work, here we propose a hybrid stock trends prediction framework (HSTPF) that combines both machine learning and deep learning methods. This proposed work analyzed the various Black Swan events and identify their impact on the stock market to deal with the uncertainty of the market. This work is also trying to reduce the computational time of the framework.

3 Methodology

The detail methodology of the proposed framework (HSTPF) is described in this section. The architecture of the proposed framework is shown in Fig. 1. This proposed work intends to build a robust forecasting system to predict the l days ahead of the Indian stock market's trends based on the current ten days stock market data. The entire operation of HSTPF is divided into three functional phases.

Table 1 Dataset description

Historical dataset name	Period	Attributes	Source
S&P BSE SENSEX	01.01.1991 to 31.12.2019	Open price, low price, high price, close price	www1.nseindia.com
Nifty 50	01.01.1996 to 31.12.2019	Open price, low price, high price, close price	www.bseindia.com
Forecasting dataset			
S&P BSE SENSEX	01.01.2020 to 31.03.2021	Open price, low price, high price, close price	www1.nseindia.com
Nifty 50	01.01.2020 to 31.03.2021	Open price, low price, high price, close price	www.bseindia.com

Table 2 Description of Black Swan events

Event	Year	Duration
Harsh Mehta Scam	Apr'1992	261 days
9/11 Attack	Sep'2001	38 days
Global Financial Meltdown	Jan'2008	418 days
China Slowdown	Aug'2015	185 days
Demonitization	Nov'2019	17 days
NBFC Crisis	Sep'2018	144 days
COVID19 Pandemic	Mar'2020	105 days

3.1 Data collection

In this work, two different types of datasets are constructed. The first dataset is named *Historical Dataset* which contains a large number of historical stock market data (open price, high price, low price, and close price) of S&P BSE SENSEX and Nifty 50. This *Historical Dataset* is used for the Black Swan event analysis. The second dataset called *Forecasting Dataset* contains the historical stock market data (open price, high price, low price, and close price) of a target stock market. The volume of the second dataset is relatively small, and it is used for stock trends prediction of that market. The detailed description of the dataset is presented in Table 1.

3.2 Black Swan events analysis

In the world economy, the unpredictable events that have a severe negative impact on the global economy and which are extremely rare are called the Black Swan event. In general, the standard forecasting tools are not capable to predict the destructibility of these events.

In this study, to capture the uncertainty of the Indian stock market, a Black Swan event analysis method is proposed. In this regard, the seven devastating global Black Swan events that have a severe impact Indian stock market are considered, and these events are presented in Table 2.

The entire operation of the Black Swan event analysis is segregated into two phases.

3.2.1 Impact calculation

To find the impact of Black Swan events on the stock market, here we propose an algorithm named "Black Swan Event's Impact(BSEI)." This algorithm generates multiple segments from the Historical Dataset and also determines the impact vector corresponding to each segment. In the following steps, we describe the BSEI algorithm.

Step 1. The entire Historical Dataset is divided into k number of segments ($Seg_i; i = 1, 2, \ldots, k$) with the segment length $sl = 10$ by striding the dataset with striding length $\lambda = 1$. So, the number of segments is calculated as follows:

$$k = n - sl - \lambda, \tag{1}$$

where n is the length(days) of the Historical Dataset.

Step 2. For each segment Seg_i, find the target closing price y_i. Here, y_i is the closing price of $(i+9+l)^{th}$ day, since the segment Seg_i contains the stock market data of the following days: $D_i, D_{i+1}, \ldots, D_{i+9}$. Here, l represents the number of days ahead prediction.

Step 3. Calculate the collapsing percentage ($Clsp$) and the recovery percentage ($Rcvy$) for each Black Swan event. Let us consider a Black Swan event $BSEvnt_r$, which is stared at the p^{th} day, maximum fall is reached on m^{th} day, and on q^{th} day, it is completely recovered. Let, $(x_p, x_{p+1}, \ldots, x_m, x_{m+1}, \ldots, x_q)$ are the closing price of a stock market during that event as shown in Fig. 2. Then, the formula to calculate the collapsing percentage($Clsp_r$) and the recovery percentage($Rcvy_r$) are as follows:

$$Clp_r = \frac{1}{x_p}(|x_p - x_m|) \times 100, \tag{2}$$

$$Rcvy_r = \frac{1}{x_q}(|x_m - x_q|) \times 100, \tag{3}$$

where x_p, x_m, and x_q are the closing price of p^{th}, m^{th}, and q^{th} day, respectively.

Step 4. To find the impact of each segment Seg_i whose closing prices are $x_i, x_{i+1}, \ldots, x_{i+9}$, an impact vector $Impct_i$ of length 2 is generated, where the first element of

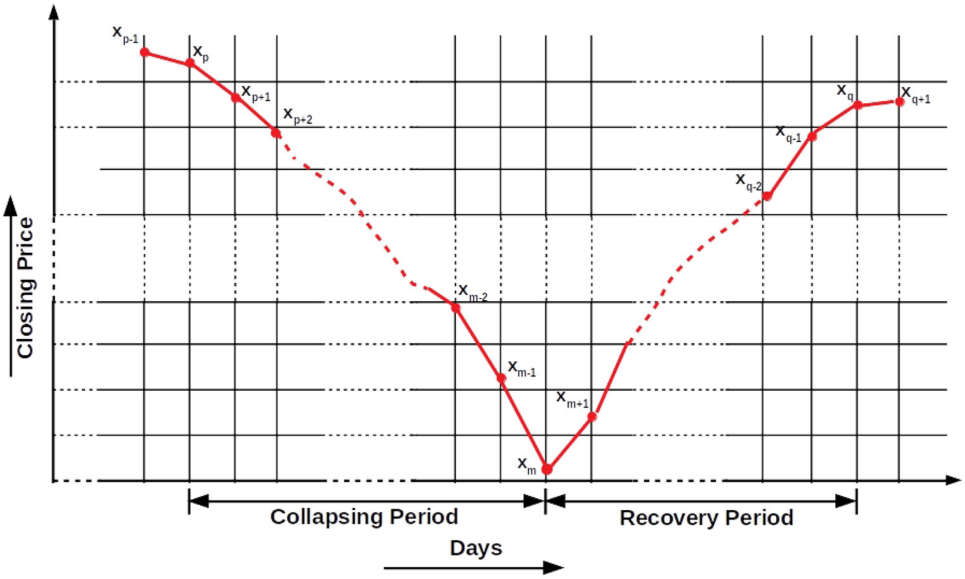

Fig. 2 Graphical representation of Black Swan event

that vector represents the impact type and the second element represents the impact score. To calculate the values of the impact vector, three strategies are used.

Strategy 1: If the y_i not belongs to the period of any Black Swan event, then set the impact type of the vector $Impct_i$ to $0(Neutral)$ and the impact score is determined by the Equation 5.

$$avg_i = \frac{1}{10}\sum_{j=i}^{i+9} x_j \tag{4}$$

$$ImpctScr_i = \frac{1}{avg_i}(|avg_i - y_i|). \tag{5}$$

Strategy 2: If the y_i belongs to the collapsing period of a Black Swan event $BSEvnt_r$, then the impact type of the vector $Impct_i$ is set to $-1(Negative)$ and the impact score is determined by the following formula:

$$BSImctScr_i = \frac{1}{100}Clp_r + ImpctScr_i. \tag{6}$$

Strategy 3: If the y_i belongs to the recovery period of a Black Swan event $BSEvnt_r$, then the impact type of the vector $Impct_i$ is set to $1(Positive)$, and the impact score is calculated by the following formula:

$$BSImctScr_i = \frac{1}{100}Rcvy_r + ImpctScr_i, \tag{7}$$

where the $ImpctScr_i$ is calculated as per Equation 5.

Step 5. Now all the segments are combined into a dataset named Historical Input Dataset (HID), and all the impact vectors are merged into another dataset called Historical Target Dataset (HTD). Here, the i^th element of the HTD represents the impact vector of the i^{th} segment of HID for all $i = 1, 2, \ldots, k$.

3.2.2 Impact prediction model construction

In this phase, we build a deep learning model (DLM) to predict the impact value from input stock market data. The architecture of DLM is shown in Fig. 3. This DLM consists of two blocks.

Block 1: This block is constructed by the series of 1D-CNN layers followed by a dense layer, where the output of one layer is taken as the input of the next layer. Here, the 1D-CNN layers are used to extract the higher-order spatial features matrix from the input dataset, whereas the dense layer converts this higher-order features matrix to a one-dimensional feature vector which is used as the input of the next block. Each 1D-CNN layer performs three operations, viz. convolution, activation, and pooling. The convolution operation operates between the input matrix and a kernel (filter) and generates a feature map. The convolution operation is performed as per the following formula:

$$F[m,n] = (I * K)[m,n] = \sum_p \sum_q I[p,q]I[m-p,n-q], \tag{8}$$

where I is the input matrix and K is the kernel. m and n denote the indexes of rows and columns of the resultant feature map F.

Then, the activation function is applied on the feature map to bound the feature map within a range. Here, we use the ReLU activation function.

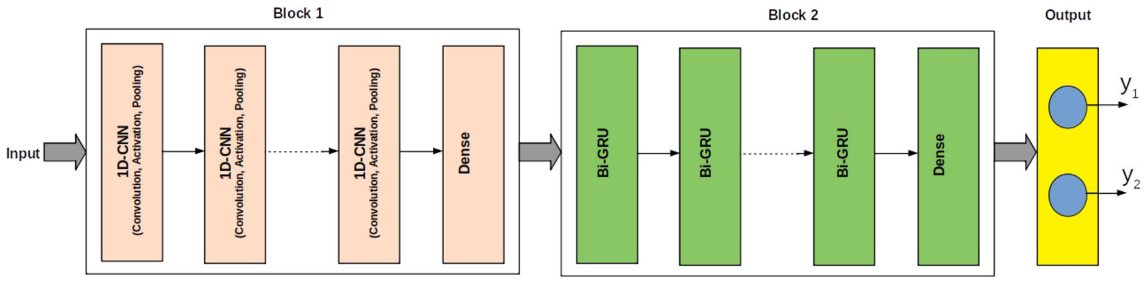
Block 1
Input
1D-CNN (Convolution, Activation, Pooling)
1D-CNN (Convolution, Activation, Pooling)
1D-CNN (Convolution, Activation, Pooling)
Dense
Block 2
Bi-GRU
Bi-GRU
Bi-GRU
Dense
Output
y_1
y_2

Fig. 3 Architecture of the impact prediction model(DLM)

$$ReLU(a) = \begin{cases} 0, & a < 0 \\ a, & a >= 0 \end{cases}. \tag{9}$$

After the activation operation, the pooling operation is performed that downsample the feature map to summarize the features. In this experiment, the MaxPooling operation is used. The next 1D-CNN layer takes this features map as the input and produces a refined feature map. In this way, this block extracts the higher-order spatial features.

Block 2: This block is build up by the series of bi-directional gated recurrent unit (Bi-GRU) layers and a dense layer where the output of each layer is fed to the next layer. And finally, the dense layer yields two outputs as shown in Fig. 3. The output y_1 represents the Black Swan impact type, whereas the output y_2 serves as the impact score. The Bi-GRU layers are used to extract the spatiotemporal long-term dependency features from both the forward as well as the backward direction of the input multivariate time-series data.

3.2.3 Model training

The datasets (HID & HTD) generated in Sect. 3.2.1 are used for the training and validation purpose of the DLM. We fed each segment of HID as the input, and the corresponding vector of HTD is considered as the target output of the model. In this study, to train the DLM, we use 70% of the HID and the remaining 30% is used for the validation purpose. Here, K-fold cross-validation is conducted to validate the performance of the model.

3.3 Stock trend forecasting system

3.3.1 Technical analysis

In this step, we generate new features called *Technical Indicators*. These features are generated by computing the technical indicators based on the closing price of the *Forecasting Dataset*. In this work, four different technical indicators [34, 35] are considered. The formula of the technical indicators is shown in Table 3.

3.3.2 Trend features extraction

Here to forecast the stock trends, two different types of trend features are extracted from the Forecasting Dataset - 1. *Daily Trend* and 2. *Future Trend*. These trend features capture three different types of trends, namely *Positive*, *Negetive*, and *Neutral* and 1, -1, and 0 are the corresponding values that are used to represent these trends. Here, we used the following formula to find the *Daily Trend*.

$$DailyTrend_d = \begin{cases} 1 & if\ x_{dc} - x_{do} > 0 \\ 0 & if\ x_{dc} - x_{do} = 0 \\ -1 & if\ x_{dc} - x_{do} < 0 \end{cases}, \tag{10}$$

where $DailyTrend_d$ is the daily trend, x_{do} and x_{dc} are, respectively, the opening price and the closing price of the d^{th} day.

The *Future Trend* feature is the target feature of our proposed forecasting framework HSTPF. To capture this feature, we use the following formula:

$$FutureTrend_l = \begin{cases} 1 & if\ x_{lc} - x_{dc} > 0 \\ 0 & if\ x_{lc} - x_{dc} = 0 \\ -1 & if\ x_{lc} - x_{dc} < 0 \end{cases}, \tag{11}$$

where x_{dc} is the closing price of d^{th} day and x_{lc} and $FutureTrend_l$ are, respectively, the closing price and the *Future Trend* after l days.

3.3.3 Dimensionality reduction

Here, to generate the input dataset, the *Technical Indicator*, and *Daily Trend* features are added with the *Forecasting Dataset*. Now, the dimension reduction (features extraction) process is performed to reduce the dimension of the *Input Dataset* to efficiently forecast the stock trends. To reduce the dimensionality of the input dataset, here we build an autoencoder as shown in Fig. 4. The reason behind the selection of autoencoder over other popular machine learning algorithms like PCA is that its performance is better especially for the multidimensional dataset [36,37].

Table 3 Formula of technical indicators

Technical indicator	Formula	Technical indicator	Formula
$MAI(n1, n2)$	$\frac{SMA(n1)}{SMA(n2)}$	$\%D$	$(D-1) \times \frac{2}{3} + \%K \times \frac{1}{3}$
$RSV(n)$	$\frac{C-Low_n}{High_n-Low_n} \times 100$	$RSI(n)$	$\frac{U}{U+D} \times 100$
$\%K$	$(K-1) \times \frac{2}{3} + RSV \times \frac{1}{3}$	$ROC(n)$	$\frac{C-C_n}{C_n} \times 100$

–$SMA(n1)$and $SMA(n2)$ are, respectively, the $n1$ and $n2$ days moving average. Here, $n1 < n2$.

–C is the current day closing price, Low_n and $High_n$ are, respectively, the n days low price and high price.

–$(K-1)$ and $(D-1)$ are, respectively, the K value and D value of the previous day.

–U and D, respectively, denotes the average of n days up and down of the closing price.

–C_n represents the closing price of n days before.

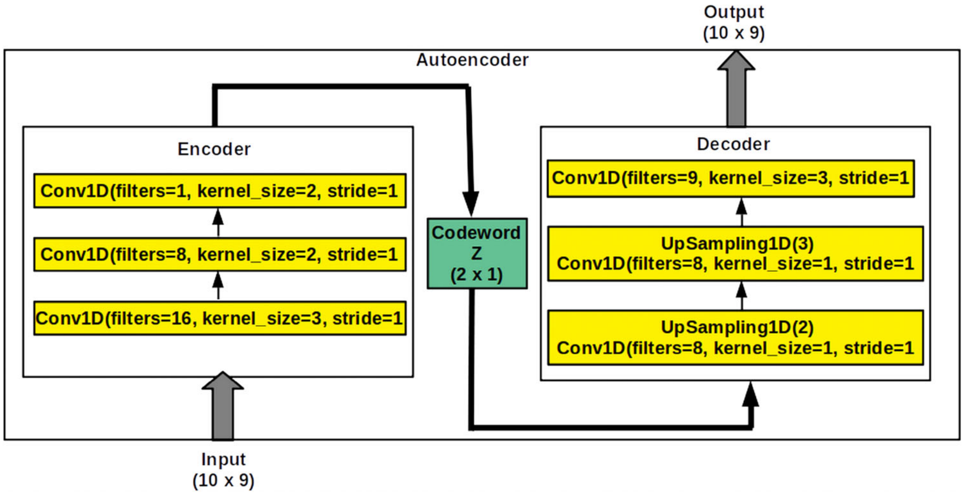
Output
(10 x 9)
Autoencoder
Encoder
Conv1D(filters=1, kernel_size=2, stride=1
Conv1D(filters=8, kernel_size=2, stride=1
Conv1D(filters=16, kernel_size=3, stride=1
Codeword
Z
(2 x 1)
Decoder
Conv1D(filters=9, kernel_size=3, stride=1
UpSampling1D(3)
Conv1D(filters=8, kernel_size=1, stride=1
UpSampling1D(2)
Conv1D(filters=8, kernel_size=1, stride=1
Input
(10 x 9)

Fig. 4 Architecture of the autoencoder

Autoencoder is a neural network-based unsupervised learning technique. It comprises two components—encoder and decoder. The encoder compresses the size of the input to a smaller dimension called a codeword, and the decoder reconstructs the original input data from the codeword. In this study, to build the autoencoder, a series of one-dimensional convolutional layers are used for the encoder part, and a series of one-dimensional deconvolutional layers are deployed for the decoder part shown in Fig. 4. Here, the autoencoder takes the input of the dimension 10×9 and produce the codeword z of size 2×1.

3.3.4 Impact features extraction

The *Impact Features* are extracted from the *Forecasting Dataset*. To extract these features, at first the impact calculation is done as per the BSEI algorithm described in Sect. 3.2.1. Then, the pre-trained DLM of section 3.2.2 is trained with the *Forecasting Dataset*.

3.3.5 Trend prediction

To predict the trend of the stock market, here we deployed the machine learning classifier. The *Impact Features* and the codeword z are combined to form the input dataset for the classifier. In this study, to get the best forecasting result, the performance of the five popular machine learning classifiers, viz. Multinomial Naïve Bayes (MNB), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), AdaBoost (AB), and Gradient Boosting Classifier (GBM), are analyzed.

4 Experiment

In this study, all the experiments are conducted on a laptop with an i3(2.6GHz, 4MB Cache) processor, 8 GB RAM, 256 GB SSD, and Ubuntu 18.01 operating system. For the simulation, the Python programming language is used. *Keras* library is used for the deep learning algorithms, and *Scikit-learn* library is used for the machine learning classifiers. Here to predict the l days ahead stock trend, we consider the cur-

Table 4 Optimal parameters of the deep learning and machine learning models

Sl. No.	Model	Optimal parameter values	
1	DLM	Block 1	layers = 4
			layer1(1D-CNN):filters=12, kernel_size=3, strides=1
			layer2(1D-CNN): filters=8, kernel_size=3,strides=1
			Layer3(1D-CNN): filters=4, kernel_size=3,strides=1
			Layer4(Dense): units=10, activation=None
		Block 2	layers=3
			Layer1(Bi-GRU): units=64, activation=tanh, return_sequences=False, dropout=0.0
			Layer2(Bi-GRU): units=32, activation=tanh, return_sequences=False, dropout=0.4
			Layer3(Dense): units=2, activation=None
2	Multinomial Naïve Bayes (MNB)	alpha:0.1	
3	Support Vector Machine (SVM)	kernel: rbf, C: 0.6	
4	K-Nearest Neighbor (KNN)	n_neighbors: 5	
5	AdaBoost (AB)	n_estimators: 120, learning_rate: 0.1	
6	Gradient Boosting Classifier (GBM)	n_ estimators: 200	

rent 10 days stock market data (open price, high price, low price, and close price). In this experiment, the HSTPF is tested for the single-day($l = 1$) ahead and also the multi-days($l = 2, 3, \ldots, 10$) ahead predictions.

4.1 Training and testing process

At first, the *Forecasting Dataset* is split into training and testing datasets, where 70% of the data is used for training purposes and the remaining 30% for the testing purpose.

In the training process, at first, we train the autoencoder. After successful completion of the training, the output from the encoder part of the autoencoder is taken. This output represents the compressed features. In the next step, the pre-trained DLM is trained with the training dataset. Now, the compressed features dataset and the impact vector dataset are combined. Henceforth, the machine learning classifiers are trained with the combined dataset.

To evaluate the performance of our proposed HSTPF, the testing dataset is used. At first, the consecutive 10 days stock market data and the technical indicators are taken as input and fed into the autoencoder. At the same time, also the 10 days stock market data are input to the DLM. Now, the outputs are taken from both the encoder and the DLM. Then, the combined output is put into the machine learning classifier, and the output of the classifier is the predicted trend.

To get the best result of our proposed HSTPF, parameter tuning is performed. After the parameter tuning, we set the parameters' optimal value as presented in Table 4.

4.2 Performance evaluation

In this study, to evaluate the performance of our proposed HSTPF, five different performance evaluation metrics are used. Here, our problem domain is the multiclass classification, so the accuracy, precision, recall, and F-measure are used as the performance evaluation metrics. The formula to calculate accuracy is expressed as:

$$Accuracy = \frac{Number\ of\ correct\ observations}{Total\ number\ of\ observations} \tag{12}$$

The accuracy is not a good measure to evaluate the performance when the input dataset is not uniformly distributed among all the classes. And to figure out the performance, here precision, recall, and F-measure are determined based on the confusion matrix. The formula for these metrics are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{15}$$

In this study, to evaluate the performance of HSTPF concerning the computational time, the training time is considered as a metric.

In this work, to analyze the effectiveness of the proposed framework, we compare them with other popular methods. Here, the performance of the proposed framework is compared with the shallow CNN and Bi-GRU models. The performance of HSTPF is also compared with the other recently published research works, viz. MLF [5], AT-GRU-M [31], and HOMM [33].

4.3 Results & discussion

4.3.1 Performance of machine learning classifiers

Here, to measure the performance of various machine learning classifiers, we generate the confusion matrix based on the single-step ahead prediction results on both the S&P BSE SENSEX and Nifty 50 datasets. Three performance metrics—precision, recall, and F-measure—are calculated from the confusion matrix for all three prediction classes (Positive, Negative, and Neutral). In Tables 5 and 6, results of these performance metrics are presented for both the datasets.

Table 5 Comparison of different machine learning classifier for HSTPF in terms of precision, recall, and F-measure for S&P BSE SENSEX dataset

Classes	Metrices (%)	Algorithms				
		MNB	SVM	KNN	AB	GBM
Positive	Precision	88.0	85.89	71.68	77.00	89.38
	Recall	83.33	80.17	91.01	88.78	93.52
	F-measure	85.84	81.92	80.20	82.47	91.40
Negative	Precision	77.01	83.91	70.80	87.36	91.95
	Recall	83.75	82.02	78.22	77.55	86.97
	F-measure	80.24	82.95	84.04	82.16	89.39
Neutral	Precision	0.00	0.00	0.00	0.00	0.00
	Recall	0.00	0.00	0.00	0.00	0.00
	F-measure	0.00	0.00	0.00	0.00	0.00

Table 6 Comparison of different machine learning classifier for HSTPF in terms of precision, recall, and F-measure for Nifty 50 dataset

Classes	Metrices (%)	Algorithms				
		MNB	SVM	KNN	AB	GBM
Positive	Precision	71.90	83.47	75.2	66.94	85.12
	Recall	84.47	87.82	85.05	90.00	87.82
	F-measure	77.70	85.60	81.23	76.78	85.60
Negative	Precision	82.97	85.10	64.46	90.43	88.30
	Recall	69.64	80.00	72.22	68.00	82.17
	F-measure	75.73	82.47	68.12	77.62	85.12
Neutral	Precision	0.00	0.00	0.00	0.00	0.00
	Recall	0.00	0.00	0.00	0.00	0.00
	F-measure	0.00	0.00	0.00	0.00	0.00

Table 5 shows the result on the S&P BSE SENSEX dataset. For the positive trend class, GBM exhibits the highest precision of 89.38%, the highest recall of 93.52%, and also the highest F-measure of 91.40%. For the negative trend class, GBM produces the highest values of 90.1%, 86.97%, and 89.39% for all three metrics. On the other hand, for the neutral trend class, the value of all three metrics of all the classifiers is 0.00%. The reason for getting the value of 0.00% is that the test sample size is small, and no two values of closing prices are the same. The result of the Nifty 50 dataset is shown in Table 6. The GBM classifier produces the highest values for all three metrics, and these values are 85.12%(precision), 90.35%(recall), and 87.66%(F-measure). For the negative trend class, the highest precision of 90.43% is achieved by the AB classifier, whereas the highest recall of 82.17% and the highest F-measure of 85.12% is attained by the GBM classifier.

The performance comparisons of the selected machine learning classifiers on both the S&P BSE SENSEX and Nifty 50 datasets are also presented in Figs. 5 and 6. These figures express the prediction accuracy for 1 to 10 days ahead of predictions. Undoubtedly, it is clear that the GBM classifier exhibits better accuracy for both datasets. And the GBM achieved his highest prediction accuracy for the singe-day ahead prediction that is 90.5% on the S&P BSE SENSEX and 86.5% on the Nifty 50 dataset, whereas the performance of the KNN classifier is poor on the S&P BSE SENSEX dataset and the MNB classifier's performance is worse on the Nifty 50 dataset. The GBM classifier exhibits stable performance for both the datasets, and it is the better classifier for our proposed HSTPF. Thus, the GBM classifier is selected for our proposed HSTPF, and the rest of the experiments are conducted on the HSTPF with GBM classifier.

4.3.2 Performance in terms of training time

In Figs. 7 and 8, the performance of HSTPF is compared with the other models in terms of training time in minutes. Figure 7 represents the training time of various models to train with the S&P BSE SENSEX dataset. It shows that the HOMM has the longest training time as 16.88 minutes, whereas the training time of MLF is the lowest, and it is 3.16 minutes. The training time of the modes to train with the Nifty 50 dataset is shown in Fig. 8. Here, also the HOMM takes the highest training time of 15.08 minutes, and the MLF takes the smallest training time of 3.93 minutes. For both datasets, S&P BSE SENSEX and Nifty 50, the training time of HSTPF exhibits the second-lowest training time of 5.19 and 4.98 minutes, respectively, which is quite small compared to the other models.

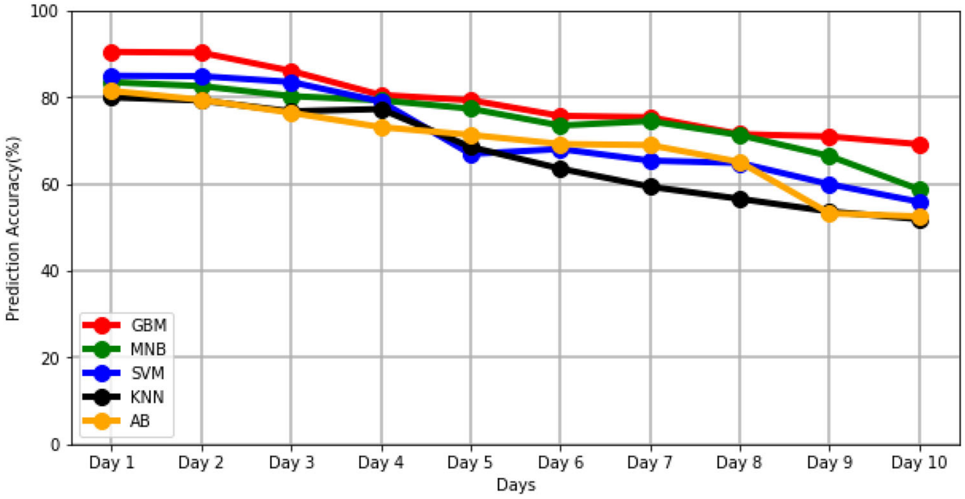

Fig. 5 Prediction accuracy of various machine learning classifiers on the S&P BSE SENSEX dataset for 1 to 10 days ahead prediction

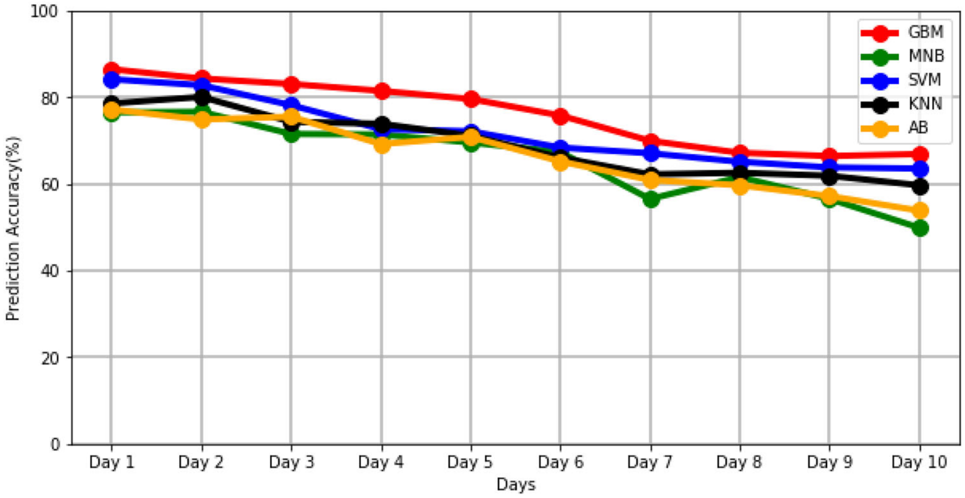

Fig. 6 Prediction accuracy of various machine learning classifiers on the Nifty 50 dataset for 1 to 10 days ahead prediction

4.3.3 Performance of Black Swan event analysis

Here, to analyze the effectiveness of the Black Swan event analysis on the proposed HSTPF, the performance of HSTPF is compared with and without Black Swan event analysis as shown in Figs. 9, 10. Both the figures represent the prediction accuracy(%) for the single-day as well as multi-day ahead of predictions. From these figures, it is absolutely clear that the prediction accuracy is significantly improved with Black Swan event analysis.

4.3.4 Performance comparison of HSTPF

The performance comparison results with the different well-established models for single-day ahead prediction is shown in Tables 7, 8. These tables represent the precision, recall, and F-measure values for all three forecasting trend classes.

Table 7 represents the comparison results on the S&P BSE SENSEX dataset. The HSTPF achieves the highest precision of 91.15%, the highest recall of 91.96%, and the highest F-measure of 91.55% for the positive trend class among all the models, whereas the lowest precision of 61.06% produces by AT-GRU-M, the lowest recall, and the lowest F-measure value produces by CNN. For the negative trend class, the highest precision value of 91.95% is obtained by HOMM, while HSTPF yields the highest recall of 88.64% and F-measure of 89.15%. On the other hand, CNN has the lowest values for all three metrics. Table 8 represents the comparison results on the Nifty 50 dataset. Here, HSTPF attains the highest precision, recall, and F-measure, and CNN produces the lowest values for all three metrics for the positive trend class. For the negative trend class, HSTPF also accomplished the better recall and F-measure, whereas the AT-GRU-M attains the best precision. Like the S&P BSE SENSEX dataset, here CNN also produces the worst result for all three metrics.

In Figs. 11 and 12, we compare the performance of different models in terms of prediction accuracy(%) for both the S&P BSE SENSEX and Nifty 50 datasets to predict 1 to 10 days ahead predictions. For both the datasets, although the performance of HSTPF is decreasing with the increase

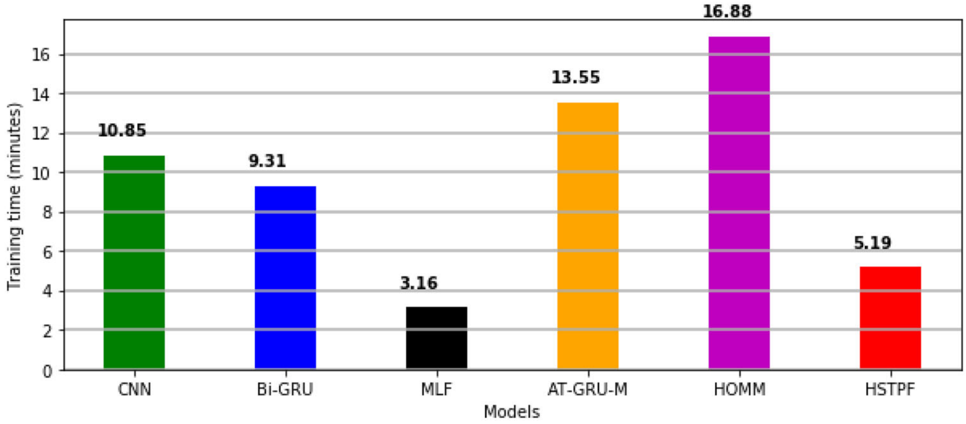

Fig. 7 Training time of various models to train S&P BSE SENSEX dataset

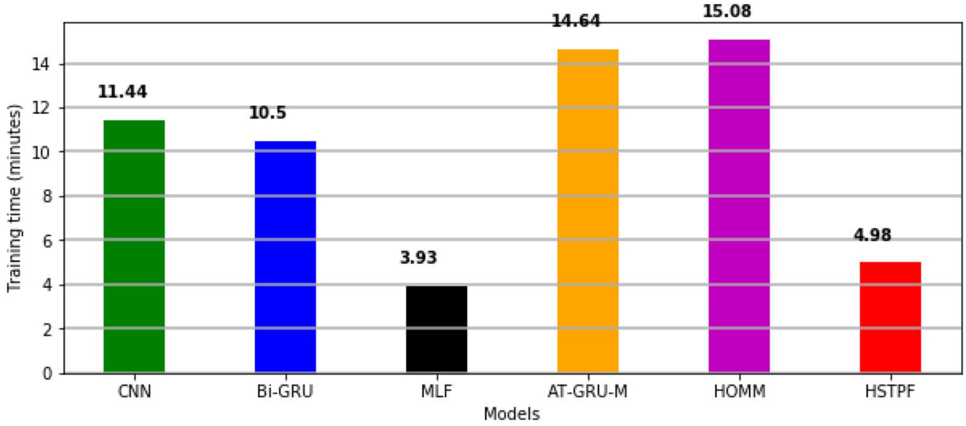

Fig. 8 Training time of various models to train Nifty 50 dataset

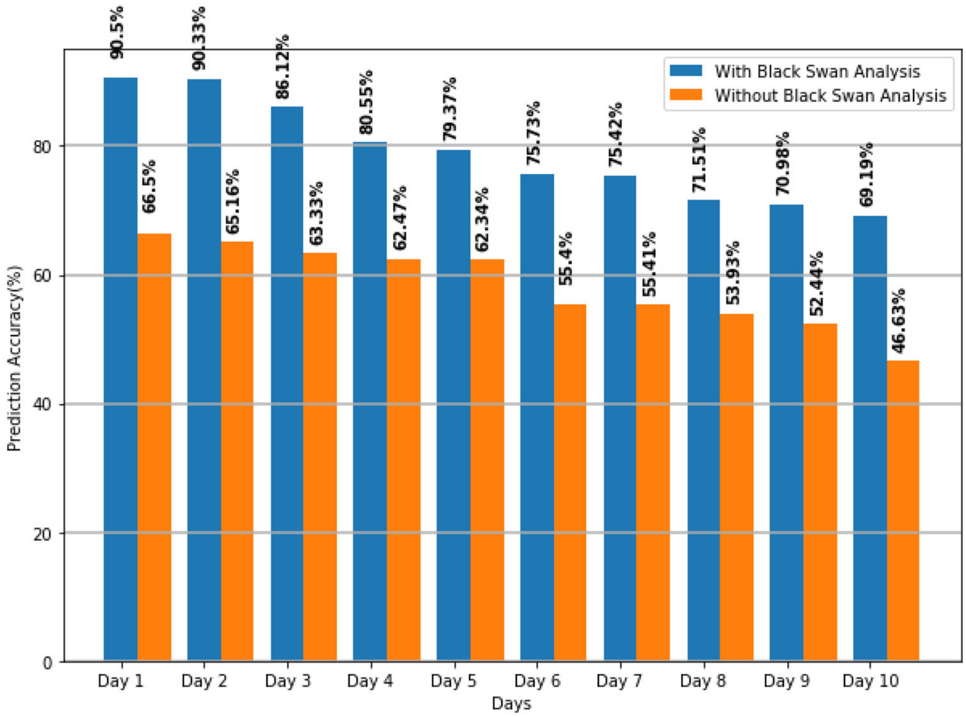

Fig. 9 Prediction accuracy of HSTPF with and without Black Swan event analysis on the S&P BSE SENSEX dataset for 1 to 10 days ahead prediction

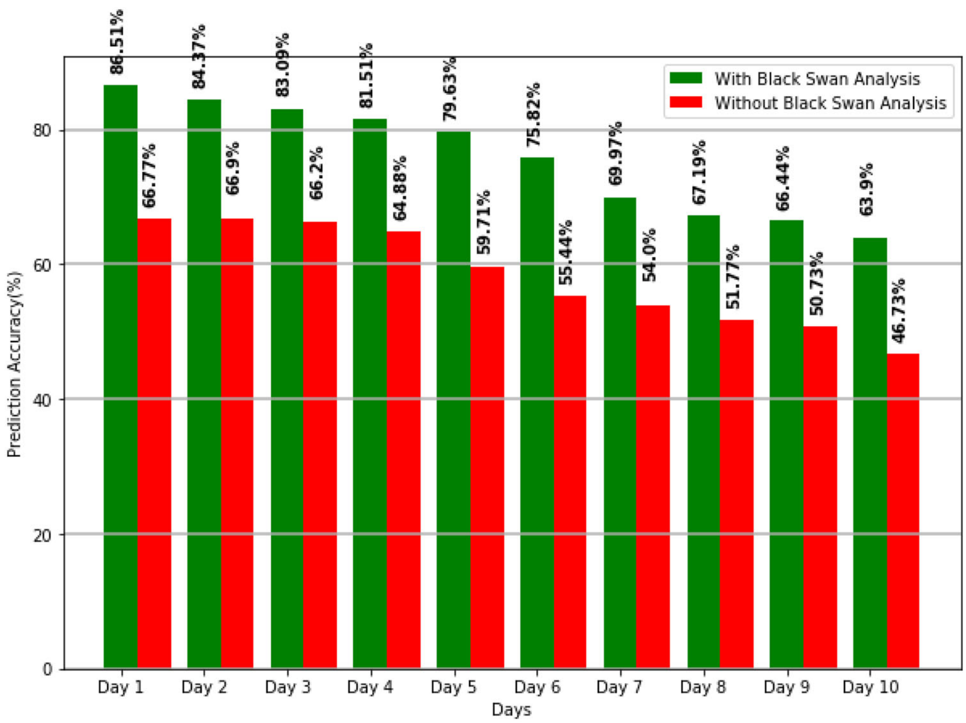

Fig. 10 Prediction accuracy of HSTPF with and without Black Swan event analysis on the Nifty 50 dataset for 1 to 10 days ahead prediction

Table 7 Comparison of different forecasting models in terms of precision, recall, and F-measure for S&P BSE SENSEX dataset for 1 day ahead forecasting

Classes	Metrices	Models					
		CNN	Bi-GRU	MLF	AT-GRU-M	HOMM	HSTPF
Positive	Precision (%)	70.80	63.72	79.65	81.42	61.06	91.15
	Recall (%)	65.31	73.47	90.00	85.19	90.79	91.96
	F-measure (%)	65.78	68.25	84.51	83.26	73.01	91.55
Negative	Precision (%)	50.57	70.11	88.51	79.22	91.95	89.66
	Recall (%)	57.06	59.8	77.78	74.39	64.52	88.64
	F-measure (%)	53.59	64.55	82.8	76.76	75.83	89.15
Neutral	Precision (%)	0.00	0.00	0.00	0.00	0.00	0.00
	Recall (%)	0.00	0.00	0.00	0.00	0.00	0.00
	F-measure (%)	0.00	0.00	0.00	0.00	0.00	0.00

Table 8 Comparison of different forecasting models in terms of precision, recall, and F-measure for Nifty 50 dataset for 1 day ahead forecasting

Classes	Metrices	Models					
		CNN	Bi-GRU	MLF	AT-GRU-M	HOMM	HSTPF
Positive	Precision (%)	48.85	48.76	90.08	79.34	83.47	92.56
	Recall (%)	64.00	69.41	78.99	83.48	82.11	84.85
	F-measure (%)	55.41	57.28	84.17	81.36	82.78	88.54
Negative	Precision (%)	61.67	72.34	69.15	79.79	76.6	78.72
	Recall (%)	46.4	52.31	84.42	75.00	78.26	89.16
	F-measure (%)	52.97	60.72	76.03	77.32	77.42	83.62
Neutral	Precision (%)	0.00	0.00	0.00	0.00	0.00	0.00
	Recall (%)	0.00	0.00	0.00	0.00	0.00	0.00
	F-measure (%)	0.00	0.00	0.00	0.00	0.00	0.00

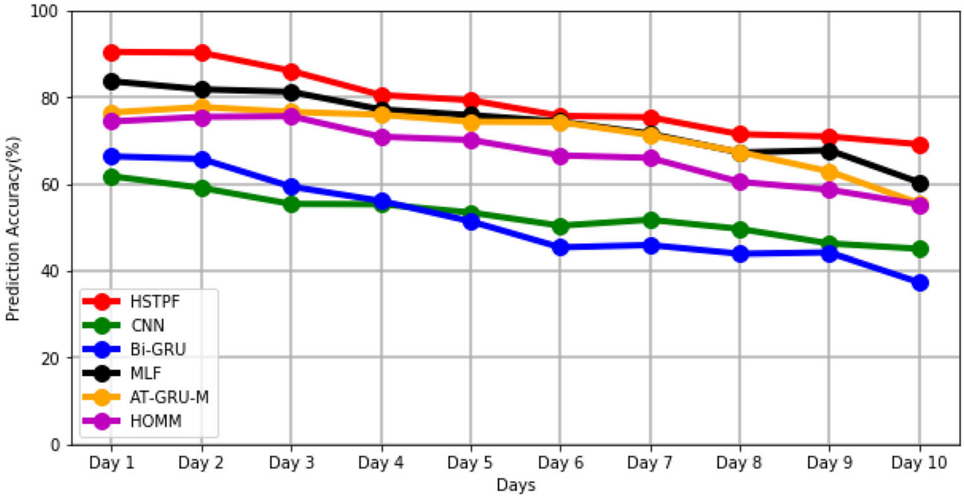

Fig. 11 Prediction accuracy of various models on the S&P BSE SENSEX dataset for 1 to 10 days ahead prediction

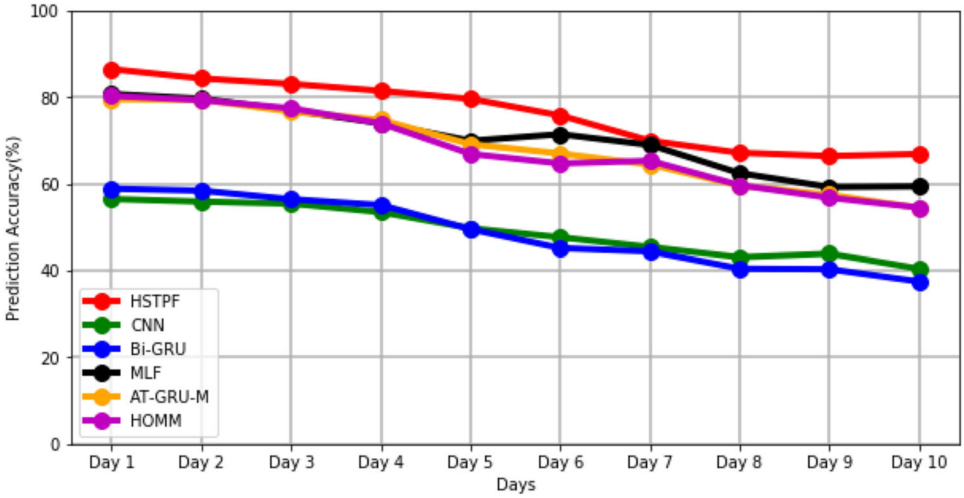

Fig. 12 Prediction accuracy of various models on the Nifty 50 dataset for 1 to 10 days ahead prediction

in days in multi-step ahead predictions, still it shows a stable performance for all the multi-step ahead predictions. The performance of CNN and Bi-GRU is very low compared to the other models. Even though the performance of MLF, AT-GRU-M, and HOMM models are very much satisfactory for all 10 days ahead predictions, then also the performance of HSTPF outruns them.

4.3.5 Performance in COVID19 pandemic

In this study, the performance of HSTPF is also analyzed during the Black Swan event. Here, the performance of various selected models is compared during the COVID 19 pandemic. In Table 9, we represent the prediction accuracy (%) for 1 to 10 days ahead of S&P BSE SENSEX trends predictions. Clearly, the prediction accuracy of HSTPF is much higher compared to the other models for all 10 days ahead predictions. And the performance of the CNN and the Bi-GRU is almost the same, and it is very poor. The performance on the Nifty 50 dataset is shown in Table 10. This table also shows that the prediction accuracy of HSTPF is considerably high compared to other models, and the performance of CNN and Bi-GRU is not up to the mark. Although the performance of HSTPF is decreased as the day ahead prediction is increased, it still produces satisfactory results up to 7 days ahead predictions.

5 Conclusion and future scopes

In this work, we intended to build a robust forecasting framework that could efficaciously predict the stock market trends, and also have been able to exhibit stable performance during unpredictable catastrophic events, and it will also be computationally faster. Here, we proposed a hybrid stock trend prediction framework named HSTPF. An innovative Black Swan event analysis method is proposed to capture the uncertainty of the market. The technical features are extracted from the raw stock market data, and these technical features are used along with the stock market data to forecast the trends.

Table 9 Accuracy (%) comparison of various models to forecast S&P BSE SENSEX during COVID 19 Pandemic

Day	CNN	Bi-GRU	MLF	AT-GRU–M	HOMM	HSTPF
Day 1	53.21	52.81	61.03	60.93	54.37	80.36
Day 2	53.32	53.11	60.14	60.81	51.82	80.34
Day 3	50.86	47.61	59.44	60.01	50.32	74.13
Day 4	47.58	45.83	55.39	55.87	50.07	73.31
Day 5	45.19	40.60	50.09	49.30	44.77	70.71
Day 6	45.41	40.89	50.09	49.30	44.77	68.13
Day 7	40.32	38.42	48.19	46.73	40.21	64.24
Day 8	35.66	34.96	4.35	44.80	40.09	60.08
Day 9	34.39	30.81	41.82	41.72	38.46	59.07
Day 10	32.06	31.03	40.09	39.11	36.98	60.01

Table 10 Accuracy (%) comparison of various models to forecast Nifty 50 during COVID 19 Pandemic

Day	CNN	Bi-GRU	MLF	AT-GRU–M	HOMM	HSTPF
Day 1	50.21	50.66	61.74	62.02	68.11	79.42
Day 2	49.38	50.372	59.10	60.38	63.21	74.26
Day 3	46.04	48.03	57.46	58.02	63.04	72.39
Day 4	46.11	45.79	55.82	54.37	59.83	70.17
Day 5	40.82	40.33	50.31	50.63	51.96	65.32
Day 6	40.01	38.41	48.36	45.94	43.03	61.26
Day 7	39.16	34.28	42.41	44.28	41.81	61.30
Day 8	30.70	32.61	41.57	42.07	42.04	59.44
Day 9	30.01	31.84	40.38	41.87	40.78	58.38
Day 10	29.17	30.21	39.16	40.08	41.53	57.26

In this study, the performance of various machine learning classifiers is also compared within the proposed framework. The HSTPF is applied to forecast the single-day and also the multi-day ahead predictions. The experimental results confirm that GBM is the superior machine learning classifier for HSTPF. The incorporation of the Black Swan event analysis method to HSTPF has significantly improved the prediction accuracy of HSTPF, and it confirms the fact that the Black Swan event analysis method plays a significant role in predicting stock trends. In comparison with other well-established models, the proposed framework has exhibited stable performance in terms of prediction accuracy and outperforms other models even during the period of Black Swan events. And the results also confirm that HSTPF is computationally faster. From these experimental results, we can conclude that HSTPF not only predicts the stock trends with a high degree of accuracy in normal circumstances, but also for unpredictable catastrophic situations with lower computational time, and it is also quite useful for multi-step ahead predictions.

The stock market can also be influenced by the other factors like news and social media. In the future, we like to capture the sentiments of news and social media and want to analyze these sentiments for more accurate stock trends prediction.

References

1. Dechow PM, Hutton AP, Meulbroek L, Sloan RG (2001) Short-sellers, fundamental analysis, and stock returns. J Financial Econom 61(1):77–106
2. Shen KY, Tzeng GH (2015) Combined soft computing model for value stock selection based on fundamental analysis. Appl Soft Comput 37:142–155
3. Mizuno H, Kosaka M, Yajima H, Komoda N (1998) Application of neural network to technical analysis of stock market prediction. Stud Inform Control 7(3):111–120
4. Chenoweth T, ObradoviĆZ, Lee SS (2017) Embedding technical analysis into neural network based trading systems. In: Artificial Intelligence Applications on Wall Street, pp. 523–541. Routledge
5. Khan W, Ghazanfar MA, Azam MA, Karami A, Alyoubi KH, Alfakeeh AS (2020) Stock market prediction using machine learning classifiers and social media, news. J Ambient Intell Human Comput pp. 1–24
6. Kumar S, Acharya S (2020) Application of machine learning algorithms in stock market prediction: a comparative analysis. In: Handbook of Research on Smart Technology Models for Business and Industry, pp. 153–180. IGI Global
7. Bandyopadhyay S, Thakur S, Mandal J (2021) Product recommendation for e-commerce business by applying principal component

analysis (pca) and k-means clustering: benefit for the society. Innov Syst Softw Eng 17(1):45–52

8. Nabipour M, Nayyeri P, Jabani H, Mosavi A, Salwana E et al (2020) Deep learning for stock market prediction. Entropy 22(8):840
9. Hiransha M, Gopalakrishnan EA, Menon VK, Soman K (2018) Nse stock market prediction using deep-learning models. Procedia Comput Sci 132:1351–1362
10. Bhanja S, Das A (2019) Deep learning-based integrated stacked model for the stock market prediction. Int J Eng Adv Technol 9(1):5167–5174
11. Chong E, Han C, Park FC (2017) Deep learning networks for stock market analysis and prediction: methodology, data representations, and case studies. Exp Syst Appl 83:187–205
12. Ardabili S, Mosavi A, Várkonyi-Kóczy, AR (2019) Systematic review of deep learning and machine learning models in biofuels research. In: International Conference on Global Research and Education, pp. 19–32. Springer
13. Ghorbanzadeh O, Blaschke T, Gholamnia K, Meena SR, Tiede D, Aryal J (2019) Evaluation of different machine learning methods and deep-learning convolutional neural networks for landslide detection. Remote Sens 11(2):196
14. Gao T, Chai Y (2018) Improving stock closing price prediction using recurrent neural network and technical indicators. Neural Comput 30(10):2833–2854
15. Alonso-Monsalve S, Suárez-Cetrulo AL, Cervantes A, Quintana D (2020) Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators. Exp Syst Appl 149:113250
16. Jiang X, Pan S, Jiang J, Long G (2018) Cross-domain deep learning approach for multiple financial market prediction. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8
17. Mukherjee D (2007) Comparative analysis of indian stock market with international markets. Great Lakes Herald 1(1):39–71
18. Phadnis C, Joshi S, Sharma D (2021) A study of the effect of black swan events on stock markets-and developing a model for predicting and responding to them. Aust Account, Bus Finance J 15(1):113–140
19. Murphy, J (1999) Technical analysis on the financial markets. new york institute of finance
20. AbdelKawy R, Abdelmoez WM, Shoukry A (2021) A synchronous deep reinforcement learning model for automated multi-stock trading. Progress in Artificial Intelligence pp. 1–15
21. Yu P, Yan X (2020) Stock price prediction based on deep neural networks. Neural Comput Appl 32(6):1609–1628
22. Yuan X, Yuan J, Jiang T, Ain QU (2020) Integrated long-term stock selection models based on feature selection and machine learning algorithms for china stock market. IEEE Access 8:22672–22685
23. Kamalakannan J, Sengupta I, Chaudhury S (2018) Stock market prediction using time series analysis. Computing Communications and Data Engineering Series 1(3)
24. Ariyo AA, Adewumi AO, Ayo CK (2014) Stock price prediction using the arima model. In: 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, pp. 106–112
25. Du Y (2018) Application and analysis of forecasting stock price index based on combination of arima model and bp neural network. In: 2018 Chinese Control And Decision Conference (CCDC), pp. 2854–2857
26. Oncharoen P, Vateekul P (2018) Deep learning for stock market prediction using event embedding and technical indicators. In: 2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA), pp. 19–24
27. Fischer T, Krauss C (2018) Deep learning with long short-term memory networks for financial market predictions. Eur J Operat Res 270(2):654–669
28. Di Persio L, Honchar O (2017) Recurrent neural networks approach to the financial forecast of google assets. Int J Math Comput Simulation 11
29. Di Persio, L., Honchar, O (2016) Artificial neural networks approach to the forecast of stock market price movements. Int J Econom Manag Syst p. 1
30. Eapen J, Bein D, Verma A (2019) Novel deep learning model with cnn and bi-directional lstm for improved stock market index prediction. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0264–0270
31. Zhao J, Zeng D, Liang S, Kang H, Liu Q (2020) Prediction model for stock price trend based on recurrent neural network. J Am Intell Human Comput pp. 1–9
32. Patel J, Shah S, Thakkar P, Kotecha K (2015) Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Exp Syst Appl 42(1):259–268
33. Wen M, Li P, Zhang L, Chen Y (2019) Stock market trend prediction using high-order information of time series. IEEE Access 7:28299–28308
34. Yeoh WL, Jhang YJ, Kuo SY, Chou YH (2018) Automatic stock trading system combined with short selling using moving average and gqts algorithm. In: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1570–1575
35. Vargas MR, De Lima BS, Evsukoff AG (2017) Deep learning for stock market prediction from financial news articles. In: 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), pp. 60–65
36. Siwek K, Osowski S (2017) Autoencoder versus pca in face recognition. In: 2017 18th International Conference on Computational Problems of Electrical Engineering (CPEE), pp. 1–4. IEEE
37. Almotiri J, Elleithy K, Elleithy A (2017) Comparison of autoencoder and principal component analysis followed by neural network for e-learning using handwritten recognition. In: 2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT), pp. 1–5. IEEE