mathematics

MDPI

Article

A Fusion Framework for Forecasting Financial Market Direction Using Enhanced Ensemble Models and Technical Indicators

Dushmanta Kumar Padhi [1], Neelamadhab Padhy [1], Akash Kumar Bhoi [2], Jana Shafi [3] and Muhammad Fazal Ijaz [4,*]

[1] Department of Computer Science and Engineering, School of Engineering and Technology, GIET University, Gunupur 765022, Odisha, India; dushmantapadhi@hotmail.com (D.K.P.); dr.neelamadhab@gmail.com (N.P.)
[2] Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology, Sikkim Manipal University, Majitar 737136, Sikkim, India; akashkrbhoi@gmail.com
[3] Department of Computer Science, College of Arts and Science, Prince Sattam bin Abdul Aziz University, Wadi Ad-Dwasir 11991, Saudi Arabia; j.jana@psau.edu.sa
[4] Department of Intelligent Mechatronics Engineering, Sejong University, Seoul 05006, Korea
* Correspondence: fazal@sejong.ac.kr

Abstract: People continuously hunt for a precise and productive strategy to control the stock exchange because the monetary trade is recognised for its unbelievably different character and unpredictability. Even a minor gain in predicting performance will be extremely profitable and significant. Our novel study implemented six boosting techniques, i.e., XGBoost, AdaBoost, Gradient Boosting, LightGBM, CatBoost, and Histogram-based Gradient Boosting, and these boosting techniques were hybridised using a stacking framework to find out the direction of the stock market. Five different stock datasets were selected from four different countries and were used for our experiment. We used two-way overfitting protection during our model building process, i.e., dynamic reduction technique and cross-validation technique. For model evaluation purposes, we used the performance metrics, i.e., accuracy, ROC curve (AUC), F-score, precision, and recall. The aim of our study was to propose and select a predictive model whose training and testing accuracy difference was minimal in all stocks. The findings revealed that the meta-classifier Meta-LightGBM had training and testing accuracy differences that were very low among all stocks. As a result, a proper model selection might allow investors the freedom to invest in a certain stock in order to successfully control risk and create short-term, sustainable profits.

Keywords: stock exchange; stock market; ensemble; cross-validation; LDA; hist gradient boosting; securities exchange; CatBoost

JEL Classification: G10; G15; G17; E17; E44; D53

check for updates

Citation: Padhi, D.K.; Padhy, N.; Bhoi, A.K.; Shafi, J.; Ijaz, M.F. A Fusion Framework for Forecasting Financial Market Direction Using Enhanced Ensemble Models and Technical Indicators. *Mathematics* **2021**, *9*, 2646. https://doi.org/10.3390/math9212646

Academic Editors: Cristina Raluca Gh. Popescu, Amir Mosavi and Bahram Adrangi

Received: 30 July 2021
Accepted: 15 October 2021
Published: 20 October 2021

1. Introduction

Forecasting future stock values has long been a contentious academic issue. For a significant stretch of time, it was assumed that fluctuations in stock values could not be predicted. The share value index is an important part of the financial system since it represents global economic success. Real-world businesses must be watchful of their security as well as their growth. At almost the same moment, investors and analysts were interested in learning about the overall capital market patterns and trends. As a result, correctness in forecasting is critical for stakeholders' well-being. In the midst of the messy and volatile character of stock markets, forecasting future price movements is a difficult topic on which academicians are seeking to improve forecasting models.

Stock value trend forecasting is a masterpiece and fascinating subject drawn by various trained professionals and researchers from fields such as financial engineering, economics, operations research, statistics, and artificial intelligence. Although a large amount of effort has been put in over the last few years, the exact figure of the stock cost and its directions are still challenging to achieve at this point, even though some high-level AI strategies are used. Globalisation of the economy constantly requires developments in the field of computational science and data innovation. In recent years, monetary exercises have been progressively developing in number with the fast financial turn of events, and their varied pattern has additionally become gradually more intricate. The securities exchange assumes a fundamental part of the monetary space of any country [1].

As of now, with the fast improvement of AI and manufactured reasoning in the previous 10 years, an ever-increasing number of market analysts have begun to execute the index value estimating of gaugeable models, have exclusive requirements, and have attempted different strategies [2]. The best standard for deciding on the presentation of the model is to look at the anticipated effects of the model with genuine information. By investigating the current exploration, we can find that even though it is hard to anticipate the securities exchange law precisely, it can foresee the future pattern of the financial exchange somewhat and decrease the dangers looked at by financial backers [3].

Currently, although there are numerous approaches to anticipate the cost of the financial exchange, to all the more likely dissect and manage the information available, as we may found delightful many issues and curtailments, it has furthermore become the point of convergence of examination to come into more critical data. Since traditional insightful techniques have apparent imperfections in taking care of non-linear issues, some machine learning algorithms are brought into securities exchange investigation [4]. A predictive model that can gauge the direction of a stock value development assists financial backers with settling on suitable choices, improve productivity, and consequently decline potential misfortunes. As a result, precise forecast and investigation of the stock market become more challenging and advantageous. For stock value prediction, we must constantly increase determining methods. Previously numerous researchers at local and overseas committed themselves to develop gaugeable monetary frameworks to anticipate index value development. Before the beginning of proficient AI calculations, analysts regularly utilised diverse statistical techniques to fabricate expectation models. There are linear models and non-linear models used for stock price prediction. Most of the linear models come under statistical methods, whereas non-linear models are based on ML algorithms. Implementing a fiscal framework to accomplish precise index value gauging has become a hypothetical and pragmatic work [5]. In principle, the conventional fiscal frameworks and the arising computerised reasoning model can accomplish the expectation of stock costs, yet the forecast impact is very extraordinary [6].

Discovering frameworks with better prescient impacts through model blend and examination is beneficial for some researchers, and it likewise has significant hypothetical importance [7]. In actuality, realised information can be infused into the monetary frameworks to anticipate future information. For example, if the stock value gauge is higher than the end cost of the day, the model predicts that the future stock cost may rise, and financial backers can decide to keep holding the stock to acquire higher venture pay [8].

On the off chance that the stock value gauge is lower than the day's end value, it demonstrates that the stock cost may fall later on. Subsequently, it is incredibly viable to develop a monetary model to acknowledge stock value gauging [9]. Moreover, if you can figure out how to precisely anticipate stock value developments and unpredictability patterns, at that point, it has a significant incentive for nations, recorded organisations, and individual financial backers [10]. As of late, there have been a developing number of studies taking a direction at the course or pattern of developments of financial markets. Now the study gradually increases by looking at the demand and trend of stock markets. Technical investigation and fundamental investigation are two different strategies by which we can foresee the securities exchange. The fundamental investigation relies on precise information on the other variables that affect the securities exchange such as miniature financial aspects; large-scale financial matters; and political and, surprisingly, mental components.

In any case, the information is typically not promptly accessible. The technical examination endeavours to make expectations dependent on past designs. In any case, these models are not, for the most part, evident because of the upheaval [11]. For customary measurable strategies, it is tough to catch the abnormality. In these conventional frameworks, we need to accept a practical connection between information and yield and attempt to fit the data according to that relationship. This has empowered scholastic scientists and business professionals to grow more unsurprising estimation frameworks. Numerous different technologies and strategies have been proposed to embrace and anticipate stock costs through multiple approaches. Yet, the appropriate blend of feature selection and the dynamic behavior of the stock market consistently are an open challenge for researchers to discover a solution. With the expanding accessibility of high-recurrence trading data and the irregularity given by prior models, it is consistently available for local and international researchers to foster a model, which will provide a reliable outcome [12].

1.1. The Inspiration Is as per the Following

Forecasting of the stock market always is an interesting and open challenge problem for researchers [8]. As day to day more information is opening up, we face new difficulties in securing and handling the information to extricate the knowledge and examine the impact on stock costs. Finding the best possible approach for predicting the daily return direction of the financial market is always a challenging and debatable topic [13]. However, the desired goal of this study is to forecast the future market. The most common and fascinating part of this research area in the forecasting of the stock market is its self-sabotaging behaviour. The rapid development of machine learning models tools and technologies always provides opportunities for the researcher to find the hidden truths of the market and analyse the market in their own ways [14]. Identification of proper feature selection increase the performance of prediction of machine learning models [15]. Only a few studies have attempted to identify significant input features [16]. More research is needed on technical indicators for finding an optimal combination of input features for predicting stock prices [17]. However, the performance of forecasting models depends on quality features, and inappropriate feature selection leads to degrading the performance of the model and returns as a biased result. As proper feature selection takes an important role in the model building process in that building a reliable forecasting model which can identify risk factors and provide the positive and negative direction of the market is equally important. Thus, proper selection of algorithms during the model building process is a large challenge for researchers. Past researchers have attempted to adopt hybridisation techniques using either base-level machine learning models or deep learning models, but there is still a question mark as to whether we can hybridise ensemble models; ensemble models are one type of hybridisation of weak learners [18]. Hybridisation of ensemble models can provide better accuracy using voting/averaging techniques. Most researchers' basic selection criteria for finding the best model is to look at the testing accuracy, which is a common and straightforward process that leads to overfitting.

Along these lines, our objective was to develop an ensemble-based hybrid model that learns from the past stock market data and gauges the directional movement of the stock.

1.2. Our Research Contributions in a Nutshell

- We propose a novel framework where six ensemble models are hybridised, minimising the model risk and increasing accuracy.
- A new set of input features were designed, providing a real test for future researchers to think of that combination.
- In this approach, we adopted two-phase overfitting protection. The first is LDA, and the second is the K-fold cross-validation. These techniques were merged into a single framework, making our model a unique one.
- Our model selection process is somewhat different and uncommon. Instead of selecting the model which provides the highest accuracy, we selected the model whose

training and testing accuracy difference is minimal, which is very much uncommon and innovative, and this selection process produces a model neither overfitted nor underfitted.

- Specifically, a long time period of data was collected for our experimental setup, which explores the performance level of volatility–stress periods and smooth trending periods and it also examines the persistence of financial crisis and clustering.

The rest of the article is figured out as given below. Section 2 portrays the related work, while Section 3 depicts the Materials and Methods, and in Section 4, we explore our proposed framework. In Section 5, we focus on the exploratory outcomes and discuss critical discoveries in our examination. Finally, in Section 6, we discuss the conclusion part of our paper and the future scope of our study.

2. Related Work

For a long period of time, financial backers and researchers were of the belief that stock cost cannot be anticipated. This conviction appeared due to the efficient market hypothesis (EMH) term coined by Fama [19]. According to Fama, due to the dynamic behaviour and non-stationary nature of financial market data, the financial market cannot be predictable [19]. The EMH says that once a piece of new information is entered into the investment securities, the market reacts instantaneously. Thus, it is impossible to crack the market.

On the other hand, the hypothesis again revised by the hypothesiser and their revised version classified the study into three forms as strong form, semi-strong form, and weak form [20]. The weak form of the hypothesis surmises that using historical prices future stock price cannot be forecasted. The semi-strong form surmises that the stock market behaves instantly as any new information (publicly available) is entered, practically showing there is no opportunity to forecast the market. The third form is the strong form, which deals with both public and private information, which implies that it does not provide financial backers an edge on the lookout. However, some researchers accept the EMH theory, and some researchers have disputed the efficient market hypothesis, both empirically and theoretically [21–24]. According to Nti et al. [25], the amenable involved in the EMH is open for discussion to choose which one is correct. According to Shiller [26], a new era opened for the financial market in the 1990s when behavioural finance was focused on academics. The Nobel Laureate Robert Shiller's [26] investigation revealed that during the period of 1989 to 2000, the up and down of the stock market was influenced by sentiment. At the turn of the century, Thaler [27] implemented behavioural finance to forecast the cave-in of the internet stock boom and accused the broadly held EMH that acknowledged all financial backers as being normal and making fair-minded figures about what is to come. According to Shiller [26], the behavioural finance remains on the opposite side of EMH and puts an idea that the market made changes inside stock always reflects genuine information. Shiller [26] showed that stock costs are very unstable over a short period of time; however, to some degree, the stock market can be predictable over long periods.

Thus, we consider the above authors' outcomes that in the current scenario, there is a chance of prediction of the stock market.

The financial market forecasting is commonly based on two factors: fundamental and technical factors [12,25,28–30]. The fundamental analysis utilises the monetary remaining of the firm, workers, the directorate, management decision policies, monetary status, company's yearly report, asset report, pay reports, earthbound and climatic conditions such as unnatural or catastrophic events, and political information to anticipate the future of the stock market [31–34]. The fundamental factor normally deals with the companies' GDP, CPI, and P/E ratios [35]. For the prediction of stock market, using a fundamental approach is more suitable for long run instead of short run forecasting [36]. The specialised investigators attempt to foresee the securities exchange through the learning of graphs that depict the historical market costs and technical indicators [37–39]. Technical indicators are statistical techniques that are calculated with the help of mathematical formulas using historical prices [40]. The development of artificial intelligence techniques and the increased number of datasets that are easily publicly available brings about new opportunities for researchers to explore something new from the market. According to Tshilidzi [41], the rapid development of AI techniques influences the EMH theory and provides an efficient way to learn from the market. A growing amount of research has been conducted [42–48], finding that post attestation demonstrates that the financial market may be anticipated to some extent [37,49]. Thus, there is a scope for investors to minimise the loss and maximise the profit when dealing with the stock market [50]. In recent studies, the financial market analysis and forecasting basically falls into two categories, i.e., statistical and machine learning [51].

2.1. Statistical Technique

Before the implementation of machine learning techniques, statistical techniques are used to learn the patterns of the stock are and given an approach to dissect and anticipate stocks. A group of statistical approaches are used, i.e., ARIMA, ARMA, GARCH, STAR, EMA, LDA, QDA, and regression techniques for the analysis of the financial market [52], with the ARIMA, EMA, and regression approaches having a predictable capability to some extent [53,54]. As the stock market is dynamic and non-linear in nature, the traditional statistical techniques have suffered a large amount to learn non-linear behaviour, and therefore the emerging machine learning techniques can avoid the limitations of traditional statistical techniques [55].

2.2. Machine Learning Technique

For the forecasting of the stock market, a large number of machine learning algorithms have been implemented [13,16,49,52,56–62]. As is known from previous studies, to predict stock market directional movement using machine learning techniques normally produces better outcomes than any other techniques [63]. Leung et al. [64] found that the exact estimates of the stock worth list development are critical for building effective trading methods such as financial backers that can fence against the expected dangers from the securities exchange. And even though a small amount of improvement on accuracy, its anticipating execution will be profoundly beneficial. Machine learning technique commonly uses two approaches to predict the stock market (a) using a single model to predict the stock market, and (b) using an ensemble of machine learning models [13,60,63,65,66]. The use of ensemble models reported by some researchers found that ensemble models provide better performance than a single predictive model [40,67,68]. According to Fatih et al. [69], there still is little research that has been done to predict the stock market using ensemble models.

As compared to traditional models, machine learning models behave more flexible. There are so many machine learning algorithms that have been applied in previous studies [70]. Examples are logistic regression, support vector machine, k-nearest neighbours, random forest, decision tree [40], and neural networks [49,52,60,71]. As seen in the literature, the most commonly used algorithms for stock market forecasting are support vector machines and artificial neural networks [72]. Milosevic et al. [73] proposed a classification framework to predict the financial market in a long window. They suggest that if the stock value rises 10% in a financial year, we can consider the stock as a good stock; otherwise, it is lousy stock. During their model building process, they extracted 11 fundamental ratios applied to different algorithms as input features. Their study revealed that in differentiation with naïve Bayes and SVM, random forest shows a good F-fcore, i.e., 0.751. Ballings et al. [37] discussed how various ML models have been created for discovering the direction of the stock market. Their study adopted different ensemble machine learning algorithms such as random forest, AdaBoost, neural network, logistic regression, SVR, and KNN, as well as the datasets chosen from European Companies. Their model attempted to predict the price movement of the long-term stock market, and their study revealed that the random forest algorithm performs well in their dataset. Choudhury et al. [74] proposed an

ANN model wherein their model used a backpropagation algorithm for the training phase and a multilayer feed-forward network for the testing phase for forecasting the value of a share. Their proposed model provided 0.996 as the regression value. Boonpeng et al. [75] proposed a multi-class classification problem in which their model can classify whether to buy, hold, or sell the stock. The author developed two models, one-against-all and one-against-one neural networks, and compared their performances with the traditional neural network. They concluded that one-against-all neural networks performed better than one-against-one and traditional neural network models, with an accuracy of 72.50%. According to Yang et al. [76], for a successful forecasting model, it is necessary to learn the non-linear factors of a stock. The authors proposed a radial basis function based on SVM with a genetic algorithm that is used for the forecasting of the stock market for the short run.

Dey et al. [77] proposed a model to forecast the stock whose input features are technical indicators. Their model was developed by using XGBoost algorithm, obtaining an accuracy level of 87.99% on the dataset Apple and Yahoo indexes. They compare their proposed model with SVM and ANN and finally revealed that their XGBoost models were the best among them. Basak et al. [19] proposed a framework for classification problems that forecast the price of the stock will increase or decrease. They used random forest and XGBoost classifier, and their study revealed ensemble models perform better if the proper combination of technical indicators is used as input features for a model. According to Ernest et al. [69], the ensemble machine learning models provide superior results in comparison with any individual machine learning model. In their study, they focused on the tree-based ensemble models, and their models were trained with three different stock exchange datasets. Their findings show that the extra trees ensemble classifier performed better than other tree-based classifiers.

Yang et al. [78] proposed an ensemble-based multi-layer feedforward network for the forecasting of the Chinese stock market. Their model was trained with backpropagation and Adam algorithms, whereas an ensemble was created with the help of the bagging approach. The performance of the model may increase if the used dataset is normalised further. Fatih et al. [69] proposed two models using multilayer perceptron with genetic algorithm and particle swarm optimisation. To train their model, they used nine technical indicators, which were recorded as the RMSE of 0.732583 and 0.733063 for MLP–GA and MLP–PSO, respectively. Finally, they concluded that hybrid machine learning methods can improve the accuracy level during forecasting. Wang et al. [79] developed a hybrid model with a combined BPNN, ARIMA, and ESM effort to predict the stock market weekly. The datasets used here were Shenzhen Integrated Index and DJIA. After successful modelling, they attempted to analyse every single framework with the combined ensemble framework. They found hybrid models performed better than traditional individual models and found 70.16% accuracy when forecasting the stock market's direction. Chenglin et al. [80] proposed a model that can accurately predict stock prices' direction. They used a combined model, a combination of SVM and ARIMA, and concluded that the performance of combined models performed better than a single predictive model. Tiwari et al. [81] proposed an ensemble model combined with the Markov framework with a decision tree to forecast the Bombay stock exchange. The proposed model provides an accuracy level of 92.1%, and it was concluded that combined models provide better accuracy than any individual models. A comparative study conducted by Prasad et al. [51] used three different algorithms, namely, XGBoost, Kalman filters, and ARIMA, and two different datasets taken, namely, NSE and NYSE. Their study was based on individual algorithm forecasting capability as well as a hybrid model also developed by them using Kalman filters and XGBoost. Finally, they compared four models and found the ARIMA and XGBoost to show promising results on both datasets, whereas the accuracy of the Kalman filter was not consistent in both datasets. A total of 87.64% accuracy level was maintained by the ARIMA model on the NSE dataset, whereas 79.44% was maintained on NYSE. On the other hand, an 88.66% accuracy level was maintained by the XGBoost model on the

NSE dataset, whereas 79.44% was maintained on NYSE. The Kalman filter model showed a promising accuracy level of 89.09% on the NSE dataset, whereas 64.96% was shown on the NYSE dataset. The hybrid model provided 76.79% on the NSE dataset and 70.91% on NYSE dataset. Instead of suggesting the best model among them all, they left the decision for the users in terms of finding the best one among the four. Qiu et al. [66] proposed a combined model LSTM with attention mechanism, i.e., WLSTM+Attention, on three different indexes, finding that the proposed model MSE was less than 0.05. Moreover, they suggested that proper selection of features can improve the predictive capability of the model.

Several authors used technical indicators as input features to train their model. Weng et al. [18] concluded that only by using macroeconomic indicators with a machine learning approach can one predict the stock index efficiently. Markovic et al. [82] implemented different technical indicators in the LS-SVM model to find the trend movement of the stock market in the Southeast European Market, finding technical indicators to have a certain level of prediction power. Valavanis et al. [83] found that approximately 20% of the stock market predictive models used technical indicators as their input features. According to the authors, technical indicators are generally used to learn the flow of complex patterns from specific stock data and to forecast the upcoming behaviours. Fernández et al. [84] found an optimal combination of technical indicators; the researchers developed several indicators suitable for their use and then found an appropriate mix for their predictive model. Andrade et al. [85], discussed how for finding the optimal combination of indicators for modelling, a large amount of effort has been made. However, still, there is no sophisticated, easy technique available for developers to select appropriate technical indicators.

We refer to a summary of various recent research on short-term finance market prediction for more comprehensive and extensive assessments in Table 1.

Table 1. Some of the resent studies based on short-term stock prediction.

Sl.No.	Authors (Year)/Publisher	Dataset Used	Target Output	Forecasting Period	Method Adopted/Model Proposed	Outcome (According to Authors)
1	[86] Nguyen et al. (2019), MDPI	KOSPI 200, S&P 500	Market direction	Short term	Transfer learning + LSTM	According to the author, their proposed model provides satisfactory prediction performance in comparison to SVM, RF, and KNN.
2	[18] Weng et al. (2018), Elsevier	13 U.S. sector indices.	Stock price	One month ahead price	QRF, QRNN, BAGReg, BOOSTReg, ARIMA, GARCH, Deep LSTM	Combination of technical indicators and Ensemble approach can significantly improve the forecasting performance.
3	[87] Bisoi et al. (2018), Elsevier	BSE S&P 500, HIS, FTSE 100	Stock price and stock direction	Short-term	VMD-RKELM	Their proposed model was superior to the SVM, ANN, naïve Bayes, and ARMA.
4	[88] Naik et al. (2019), Springer	NSE data	Market direction (up or down)	Short term (intraday)	Five-layer DNN	Five-layer DNN outperformed 8 to 11% than three-layer ANN.
5	[89] Lahmiri et al. (2019), Elsevier	Six American stocks	Stock price	Short term (intraday)	VMD, PSO, BPNN	According to the authors, their proposed VMD–PSO–BPNN model is superior to the PSO-BPNN model.
6	[12] Dhanya and Yadav (2019), Springer	INDIA, Nifty Index	Stock price	Short-term	ANN, SVR, EMD, EEMD, CEEMDAN	The proposed CEEMDAN-SVR model performed well as compared to other models. The authors want to improve their model using boosting and bagging algorithms and deep-learning algorithms.
7	[66] Jiayu et al. (2020), PLoS ONE	S&P 500, DJIA, HSI	Stock price	Short-term	Attention-based LSTM	The author addressed that their proposed model shows promising results in comparison to widely used LSTM and GRU models.
8	[13] Xiao and David (2019), Springer	US SPDR S&P 500 ETF (SPY)	Stock direction (up or down)	Daily return direction	DNN + PCA	The author revealed the addition of a proper quantity of hidden layers can achieve the highest accuracy, as is also the case in future.
9	[15] Shen et al. (2020), Springer	Chinese stock market	Stock trend	Short-term	PCA + LSTM with feature engineering	The authors stated that instead of focusing only on the best model, to find the best predictive model, it is essential to add an innovative feature engineering approach to improve model performance.
10	[90] Cervelló-Royo et al. (2015), Elsevier	Athens Stock Exchange	Market trend (bull/bear-flag)	Short-term	Fuzzy model + technical indicators	According to the authors, their model is a conservative model wherein during the bull market, small losses and small gain will be found.

The summary given in Table 1 indicates that there is some scientific study in place to predict the stock market in the short run, and the findings are remarkable [28,63,69,91] to some extent, inducing us to take our direction of study in the short run to forecast the stock market. Zulkernine et al. [92] suggested that it is easier to predict the stock market on a long-term basis than via daily stock forecasting. This is because the daily stock forecasting data constantly fluctuate and are full of noise. Thus, in our study based on fusion ensemble models and technical indicators, a brief review was conducted to some extent in our literature survey. According to Weng et al. [18], the prediction results may increase using the voting or averaging technique of different ensemble models. Singh et al. [93] provide their opinion that in recent times the machine learning models have shown promising results. Consequently, we aimed to build a hybrid model which learns from the historical prices in form of indicators and gauges the direction of the stock on the next day.

Some exploration leads to using a combination of unique models to produce their ensemble or hybrid models with technical indicators for the directional movement of individual stocks. Some researchers use pre-existing ensemble models such as XGBoost, CatBoost, etc., for finding the trend of the stock, and some of them use deep learning techniques to forecast the stock market. Thus far, we have noticed that more researchers are trying to develop hybrid models, i.e., a combination of individual machine learning models or deep learning models or their variety. Another challenge researchers face in finding a proper mix of input features is a very fuzzy and tedious job. We believe that there is still a gap in the hybridisation of individual ensemble models into a single frame with a unique combination of input features.

Therefore, after a successful review, we proposed a novel approach wherein ensemble models used a stacking framework, and the stacking framework takes the trained ensemble models as the base-level classifiers. Again, the ensemble models are to be used as the ensemble model meta-classifiers. In addition, we developed a unique combination of input features for the prediction of the stock market.

3. Materials and Methods

When we attempted to develop an ensemble framework, six types of ensemble algorithms were used: XGB classifier, AdaBoost Classifier, Gradient boosting, LightGBM, CatBoost, and Hist gradient boosting as base learners. In this section, we briefly describe the aforementioned base classifiers, and finally, we explain our final framework. Before discussing our framework, we first establish what ensemble learning is and how it works.

3.1. Ensemble Learning

A method takes forecasts from numerous ML models or similar forecasting from the same models at various times to make them more precise results. Forecast from a solitary individual model probably will not produce that many exact outcomes, and thus we require fostering a gathering-based AI model whose prescient limit is much higher than a solitary calculation. Figure 1 depicts the common structure of an ensemble model.

Bagging and boosting are two effective techniques used in machine learning to ensemble the models. Here, in our experiment, we used the boosting technique stacking classifier method; the stacking classifier works on a two-stage process. First, it sequentially creates base learners that are used as the input for the final stage. Then, in the final stage, the stacking classifier builds a meta-classifier using the base learners, which are treated as level-1 weak learners.

3.1.1. Gradient Boosting

Gradient boosting is a type of classifier used to merge different slow learners to produce a robust forecasted model [94]. The following principle describes gradient boosting.

1. Initially, develop an error function, and that function is optimised at the time of model building.

2. Iteratively develop weak models for forecasting.
3. Finally, all the weak models are merged a create a robust model with minimising error function.

Our model-building process begins with a comparatively weak learner model (depending upon our dataset). Then, iteratively, a weak learner is converted to a better classifier F_m(x), and that classifier becomes a robust classifier.

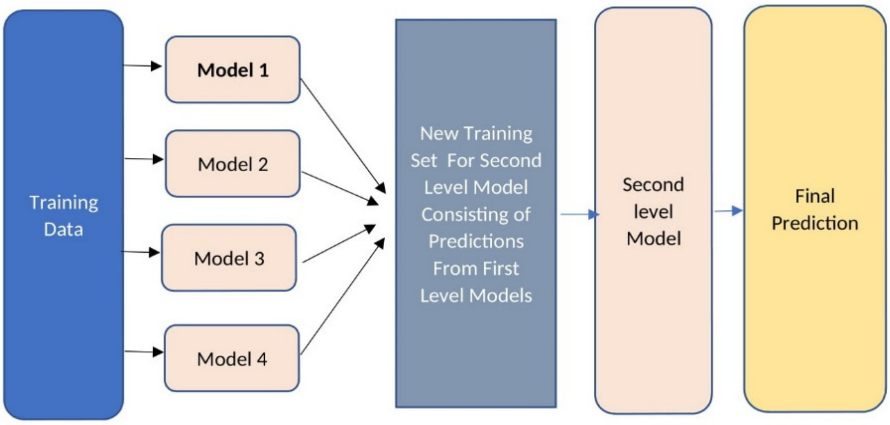

Training Data
Model 1
Model 2
Model 3
Model 4
New Training Set For Second Level Model Consisting of Predictions From First Level Models
Second level Model
Final Prediction

Figure 1. The common structure of an ensemble model [67].

3.1.2. AdaBoost

In 2003, Freund and Robert [95] developed a meta-algorithm and coined its name, AdaBoost; they were awarded a Nobel Prize for their work. According to them, AdaBoost can form an alliance with other machine learning algorithms to increase the accuracy level. Here, in this modelling technique, the weak learners are created sequentially.

During model training, an AdaBoost classifier can be represented as

$$FM_t(z) = \sum_{t=1}^{T} fm_t(z) \tag{1}$$

where the object z is used as an input for each weak learner, i.e., fm_t and a value is returned which represents an object of that class. For instance, in a binary classification problem, the predicted class object and the absolute value depend upon the outcome of the weak learner's sign. Similarly, the sample belongs to the positive class, and the Tth classifier falls into the positive class or otherwise falls into the negative class.

In the training set for each sample, a weak learner generates an output hypothesis $h(z_i)$. During each iteration t, a coefficient αt is assigned to the selected weak learner; for instance, a minimised sum of error terms Em_t will be calculated for the t-stage classifier.

$$Em_t = \sum_{i=1}^{T} Em[Fm_{t-1}(z_i) + \alpha t\ h(z_i)] \tag{2}$$

Here, $Fm_{t-1}(z_i)$ is a robust classifier that is derived from the previous stage training, $Em(Fm)$ denotes the sum of error terms, and $fm_t(z) = \alpha_t h\ (z)$ is the weak classifier which is considered for addition to the final classifier.

3.1.3. Extreme Gradient Boosting (XGBoost)

The XGBoost model is an early experimental model of a Ph.D. student at Washington University [95]. XGBoost is an improved version of the gradient boosting algorithm, which is more scalable and efficient. The features that make the XGBoost algorithm something different are provided here [63]. The automatic feature extraction can be possible. XGBoost supports the regularisation technique to avoid overfitting and has the capability to learn from non-linear datasets. Moreover, the parallelisation feature makes the XGBoost train with multiple CPU cores. It is one of the tree-based ensemble additive models that are composed of multiple base learners. In general, the XGBoost can be represented as

$$\begin{array}{l} \mathrm{F} = \{\mathrm{m1},\ \mathrm{m2},\ \mathrm{m3},\ \mathrm{m4} \ldots\ldots\ldots\ldots \mathrm{m_n}\} \text{ set of base learners} \\ \text{Predictive model} = \hat{\mathrm{y}}_\mathrm{i} = \sum_{\mathrm{t=1}}^{\mathrm{n}} \mathrm{mt(xi)} \end{array} \quad (3)$$

where $\hat{\mathrm{y}}_\mathrm{i}$ is the final predictive model, which is the combination of all weak learners, and x is the input feature for each weak learner, i.e., m.

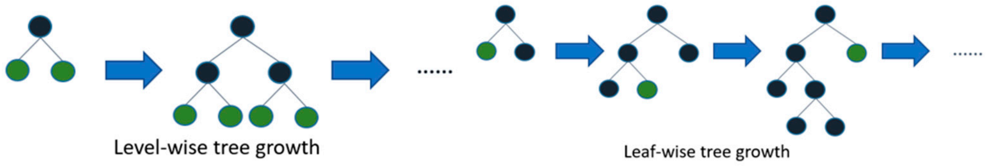
From the paper [94], we extracted the objective function for XGBoost as given below:

$$Obj(\theta) = \sum_{i=1}^{m} L(z_i, \hat{z}_i) + \sum_{t=1}^{T} \Omega(f_t) \quad (4)$$

By looking at Equation (4), we see that the objective function has two parts; the first part denotes the loss function, i.e., L denotes the training loss of either logistic or squared loss, and the second part represents the addition of each tree's complexity. z_i is actual value and $\hat{z}_i$ is the predicted value, whereas Ω is the regularisation term, T denotes the total number of trees, and f is the function.

3.1.4. The LightGBM

The LightGBM is the most diversely used boosted model that supports parallel training such as extreme gradient boosting [96]. When dealing with multi-dimensional datasets, the LightGBM works much better than the traditional boosting algorithms or XGBoost. Typically, the boosting algorithms split the tree structure horizontally (i.e., level-wise growth), whereas the LightGBM increases the tree structure vertically. Figure 2 shows the growth of tree division between level-wise and leaf-wise.



(**a**) (**b**)

Figure 2. Tree division level-wise and leaf-wise.

3.1.5. CatBoost

CatBoost is the first Russian machine learning algorithm, developed in the year 2017 by the researchers of Yandex [97]. It is one of the tree-based boosting algorithms. CatBoost stands for categorical boosting, but it deals with absolute values and with other matters (not only with category features but also with regression problems and automatic feature engineering also possible for data). Therefore, compared to the training time of different gradient boosting algorithms, CatBoost takes less time to train. Generally, in a boosting technique, we follow a standard GBT technique for the construction of decision trees, but CatBoost follows two ways of constructing the tree. One is an ordered technique, and the other is a basic technique.

A random permutation technique is applied in the ordered mode for training that follows n number of supporting models, i.e., M1 . . . Mn, such that the Mi is trained with the help of the earliest I samples in the permutation. In each iteration, for obtaining the residual of the jth sample, the MJ-1 model is used.

3.1.6. Histogram Gradient Boosting

The GBDT framework takes a longer time to train the model as the number of datasets increases, and sometimes the average accuracy level also decreases. Thus, histogram-based gradient boosting is very much effective when there is a large size of datasets [98]. Moreover, this technique reduces training time without losing the accuracy level. Thus, we can say histogram-based gradient boosting is a technique for training faster decision trees used in the gradient boosting ensemble. The splitting principle of histogram-based gradient boosting is as follows: instead of finding the split points on the sorted feature values, the histogram-based algorithm buckets continuous feature values into discrete bins and uses these bins to construct feature histograms during training. Since the histogram-based algorithm is more efficient in both memory consumption and training speed, we developed our work on its basis.

3.2. Dimensionality Reduction Technique

The dimensionality reduction technique is used to minimise the number of features during the training dataset. Dimensionality reduction techniques are applied to the ML models to avoid overfitting issues. As the number of dimensions decreases, the corresponding training parameters also decrease, making the model more straightforward and indicating the degree of freedom. If the number of parameters increases, then the degree of freedom is also high, leading to overfitting the model. That means our model will perform better when the training dataset is provided, but it may not serve better when we provide a test dataset. It is a data preparation technique that will be applied first on the dataset, but it must be remembered that the dataset must be cleaned and scaled before use [99].

3.3. Evaluation Matrices

To examine our proposed model's performances, we used the performance matrices, i.e., accuracy, ROC curve (AUC), and F-score are used. Thus, we took the blend of matrices rather than a solitary one to check the framework's performance. The performance matrices are given below.

$$Precision = \frac{t_p}{t_p + f_p} \tag{5}$$

$$Recall = \frac{t_p}{t_p + f_n} \tag{6}$$

$$Fscore = 2\frac{Precision * Recall}{Precision + Recall} \tag{7}$$

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \tag{8}$$

where

t_p represents the total number of true positive values;
t_n represents the total number of true negative values;
f_p represents the total number of false positive values;
f_n represents the total number of false negative values.

According to Sokolova et al. [100], the area under curve is an appropriate evaluation matrix for classification problems; when the AUC value increases, the prediction performance of the model also increases.

3.4. Tools and Technologies Used

For the technical analysis, Python environment with Anaconda and Google collab was used for model development. Our total model development procedure was executed with an Intel processor (core-i5-1035G1, 1.19 GHz) with 8 GB of memory and a 64-bit Windows operating system.

3.5. Dataset

For our novel approach, we used five different datasets of four different countries, i.e., S&P 500 Index (S&P 500) USA, Dow Jones Industrial Average (DJIA) HONG KONG, Hang

Seng Index (HSI) USA, NIKKEI 225 JAPAN, and DAX PERFORMANCE INDEX UNITED KINGDOM. The data on equity indices are daily. All datasets have the equity indices of each trading day.

The period in which the dataset was taken and tested in our model was from 3 January 2000 to 1 July 2019 for DJIA; for S&P 500 Index, the HSI data were from 2 January 2002 to 1 July 2019; the DAX index data were from 12 December 1987 to 18 August 2021; and the NIKKEI 225 data were from 5 January 1965 to 20 August 2021. The Dow Jones Industrial Average index had 4372 tuples, the S&P 500 index had 4904 tuples, the HSI had 4304 tuples, the DAX index had 8495 tuples, and the NIKKEI 225 index had tuples initially. Preliminarily, the records of the dataset had the existing features, i.e., trading volume, high, close, open, and low with the corresponding trading data. 'Yahoo/Finance' portal is a reliable source to download our dataset.

Data Pre-Processing

In previous related studies, there were no specific rules for selecting related input features to forecast the flow direction of the index. Hence, without hesitation, we can say that each technical feature has its hidden behaviour. Using this covert behaviour, the investors try to analyse the current situation and decide whether to buy or sell, according to Weng et al. [18]. Finally, given their conclusion on the technical indicators that analyse the hidden behaviour of these input indicators, it can forecast the monthly closing price of major U.S. indices. Therefore, we used technical indicators and some other features to predict stock index movement in this research.

Once our raw dataset was received, the data needed to be preprocessed. During data preprocessing, we had to follow the following steps:

(a) Generally, the index extracted from the web portal has some existing features that are open, close, low, high, etc. Now, looking at the dataset, we had to handle the null and missing values.

(b) In the second step, we extracted 23 technical indicators using the preexisting dataset described in the previous point. Apart from the technical indicators, we extracted two more features, i.e., the difference between the open and close price, which reflects the increase and decrease of stock value on that day. Another one is to find out the volatility, i.e., the difference between high and low stock prices.

(c) Label generation: In label generation, we constructed a response predicted variable, the binary feedback variable, i.e., $Z_t \in \{0, 1\}$, for individual trading days of our stock. The feedback variable that will be forecast on the T'th day is calculated as

```
If $\mathrm{Open}_t < \mathrm{Close}_t$
    Then
        $Z_t = 1$
    Else
        $Z_t = 0$
End If
```

Z_t is the forecast label labelled as 'TREND' that is used as our predicted variable, Open is the opening price of the index on the day, and Close_t is the closing price of the index on the day. Here, we assume that when the Z_t value returns '1', the stock price will increase, and when the Z_t value returns '0', we consider the stock price to have decreased.

(d) Although we are dealing with technical indicators representing a stock's hidden behaviour, we must find a perfect combination of input features with no multi-collinearity issues. The problems with multi-collinearity from a mathematical viewpoint are that the coefficient gauges themselves will, in general, be untrustworthy and that variable is not measurably critical; because of these disadvantages, we ought to consistently check for multi-collinearity in our dataset. For checking multi-collinearity, we have to create a corelation matrix with the help of the correlation function corr(·)

(which is used to find the pairwise correlation of all columns in the dataframe) function. This function creates a matrix with a correlation value with the combination of each variable. Therefore, when we diagonally check the matrix, we will obtain correlation values. By looking at the matrix, we have to remove the features whose values are more than 50. Thus, quickly looking at the matrix, we can easily identify the highly correlated values, which should be released. During this feature selection process, we used 23 technical indicators along with seven standard features. After successful correlation testing, we found only seven input features perfectly combined and ready to train our model.

Finally, we found four technical indicators and two derived features, as well as one pre-existing feature, whose descriptions are given below for our experiment. For our technical analysis purposes, we used (Ta-Lib) library, popularly used by traders and researchers for calculating technical indicators. This library can be downloaded from the www.ta-lib.org (accessed on 1 January 2021) website [101–103]. In this study, we employed technical indicators with other dummy variables as our input features for our ensemble models.

Average Directional Movement Index (ADX)

This indicator is used to find out the movement of price in a positive or negative direction.

$$\text{ADXtd} = ((\text{ADXtd-1} * (\text{n\_day} - 1)) + \text{DXtd})/\text{n\_day}$$

Triple Exponential Average (TRIX)

The goal of TRIX is to find out the change of price percentage between two triple SMEA.

$$\text{Trix} = (\text{ema3n} - \text{ema3n-1})/\text{ema3n-1}$$

where ema3n denotes the previous n period's ema.

Percentage Price Oscillator (PPO)

The PPO can be calculated as the difference between the moving averages on n different lengths, i.e., slow-moving average and fast-moving average, and divided by slow-moving average.

$$\text{PPO} = ((\text{F\_ma} - \text{S\_ma})/\text{S\_ma}) * 100$$

where

- F_ma (fast-moving average for a short period);
- S_ma (slow-moving average for a long period).

Ultimate Oscillator (ULT)

ULT indicates whether our stock is oversold or overbought with the goal that we can produce whether we purchase or sell the stocks.

Open

This is a pre-existing feature that denotes any stock price during the opening of the stock in each day.

Open–close

This is an extraction of one feature that indicates the difference between daily transactions' opening and closing values.

High–low

This is also an extracted feature that finds the volatility of each trading day. It is calculated as the difference between the high price and low price value of that day.

(e) After obtaining the useful features, we divided our database into two parts: 75% of the data reserve for training and 25% of the data to test our predictive model.

(f) Finally, in the data processing step, we implemented the scaling technique to normalise our features, which are to be inputted into our model. The statistical description of our three datasets are provided below in Tables 2–6, in which we exploit each of the features of max, min, mean, and standard deviation values of all three datasets.

Table 2. The DJIA index.

Feature	Max	Min	Mean	Standard Deviation
Open	26,833.470700	6547.009766	14,058.426953	4872.465559
Open–Close	1041.839840	−1020.718750	−3.178378	133.911740
ADX	69.127234	8.407117	23.798045	9.135691
TRX	0.208359	−0.456629	0.021683	0.101179
ULT	84.329733	23.405777	54.456530	10.764429
High–Low	1596.648440	25.150390	160.771935	116.553829
PPO	5.590482	−9.709044	0.157680	1.486415

The Dow Jones Industrial Average index had 4372 tuples.

Table 3. The HSI.

Feature	Max	Min	Mean	Standard Deviation
Open	33,335.480470	8351.589844	20,240.157879	5563.590816
Open–Close	1356.910160	−1441.719730	9.175004	198.295114
ADX	57.009177	8.688415	23.420749	8.575097
TRX	0.512277	−0.716833	0.022766	0.157158
ULT	86.389832	18.160742	52.695513	10.444615
High–Low	2060.559570	0.000000	260.297083	177.965756
PPO	7.019466	−11.705714	0.148573	2.112400

The HSI had 4304 tuples.

Table 4. The S&P 500 index.

Feature	Max	Min	Mean	Standard Deviation
Open	2952.709961	679.280029	1539.047669	5563.590816
Open–Close	104.010010	−104.57983	−0.194698	14.922201
ADX	55.046212	8.137028	22.807377	7.958571
TRX	0.235341	−0.569180	0.014379	0.110814
ULT	87.234912	21.561675	54.523968	10.889013
High–Low	125.219971	2.900025	18.132874	12.235494
PPO	7.019466	−10.804408	0.100939	1.587514

The S&P 500 index has 4904 tuples.

Table 5. The NIKKEI 225 index.

Feature	Max	Min	Mean	Standard Deviation
Open	38,921.648438	1020.48999	12,985.66006	8041.057993
Open–Close	1977.890625	−2676.550781	3.268797	158.894006
ADX	75.0401	5.06951	24.674705	10.25368
TRX	0.441637	−0.679374	0.023049	0.142088
ULT	100	4.24892	53.45467	15.33835
High–Low	4206.3	0.00000	142.392	181.764
PPO	9.277873	−14.0134	0.162396	1.948208

The NIKKEI 225 index had 13927 tuples.

Table 6. The DAX index.

Feature	Max	Min	Mean	Standard Deviation
Open	15,948.15	1211.24	6084.276	3693.168
Open–Close	702.86	−508.54	0.680938	73.83413

Table 6. *Cont.*

Feature	Max	Min	Mean	Standard Deviation
ADX	59.474862	7.034225	22.703418	8.590422
TRX	0.399542	−0.562701	0.031402	0.150613
ULT	94.791195	5.549906	53.478123	12.127739
High–Low	921.060546	0.00000	89.470937	79.640647
PPO	6.215173	−16.024076	0.220511	2.126011

The DAX index had 8495 tuples.

4. Proposed Framework

The objective of our study was to develop a framework in which different ensemble models are combined together to form a single predictive model. In a standard stacking framework, the training database is fitted by the base learners known as first-level classifiers. These base-level classifiers after training are used as the input features for the second-level meta classifiers. However, sometimes in ensemble techniques, the level-1 models show overfitting problems. Thus, we simply introduce the dimensional reduction technique to prepare necessary inputs for the first-level classifiers that may not create any over-fitting issues.

We trained all the base-level classifiers in our fusion-based work and converted them into a forecast model; these are used as input for our stacking framework. Hence, we gathered each base classifier's predictive output in our model development process, which is treated as a new set of data for our final model. Thus, categorically, we divided the model into two phases. In the first phase, XGBoost, Adaboost, GB, LightGBM, CatBoost, and Hamming Gradient Boosting are used as level-1 classifiers. The stacking classifiers will be used as a level-2 classifier which is called a meta-classifier and is used to extract the preprocessed hidden features from the level-1 classifiers and combine the level-1 classifiers to make a strong level-2 classifier. The developed framework and pseudo-code are given in Figure 3 and Algorithm 1.

Algorithm 1. Pseudocode for the proposed framework.

```
Input: Training data $T_d = [x_k,\ y_k]_{k=1}^{n}$
Output: A boosted stacking meta classifier model denoted as M
Step 1: Implementation of dimensional reduction technique to prepare a training set for base-level classifiers.
Step 2: Initialise E to 6 (number of base-level classifiers)
  For i < −1 to E do
    Read the baselevel classifier $m_i$
Prepare a training set for first level classifiers
    For t < −1 to $T_d$ do
Train the model
     $m_i$ (t)
     end for
     $fm_i < -m_i$
   end for
 Step 3: Read the level-2 meta classifier $[M]_{j=1}^{6}$
 Train the meta-classifier with cross-validation technique
   For j < −1 to E do
           For i < −1 to E do
               Ds = Ds + $fm_i$
           End for
 Create a meta-classifier
           M ($D_s$)
   End for
 Step 4: Return (M)
```

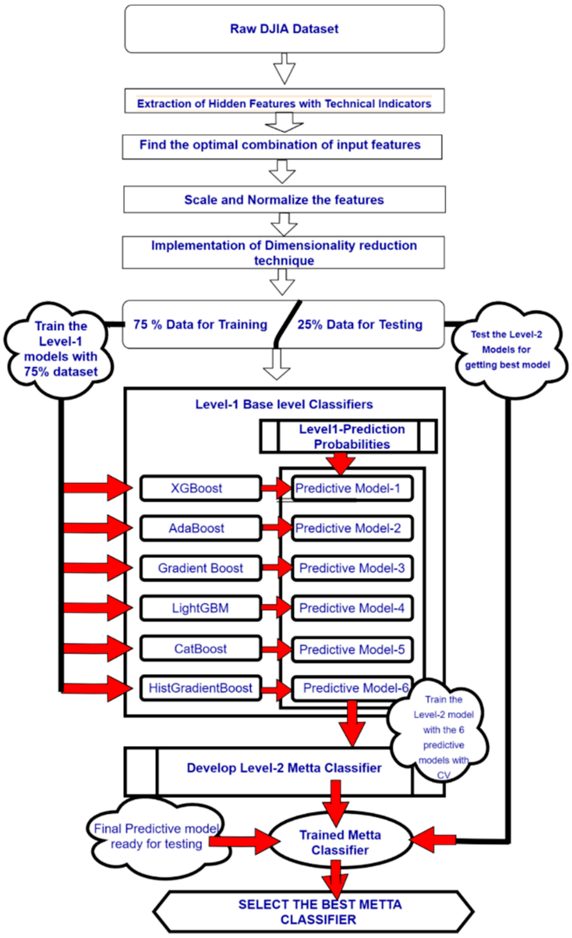

Raw DJIA Dataset
Extraction of Hidden Features with Technical Indicators
Find the optimal combination of input features
Scale and Normalize the features
Implementation of Dimensionality reduction technique
Train the Level-1 models with 75% dataset
75 % Data for Training
25% Data for Testing
Test the Level-2 Models for getting best model
Level-1 Base level Classifiers
Level1-Prediction Probabilities
XGBoost
AdaBoost
Gradient Boost
LightGBM
CatBoost
HistGradientBoost
Predictive Model-1
Predictive Model-2
Predictive Model-3
Predictive Model-4
Predictive Model-5
Predictive Model-6
Train the Level-2 model with the 6 predictive models with CV
Develop Level-2 Metta Classifier
Final Predictive model ready for testing
Trained Metta Classifier
SELECT THE BEST METTA CLASSIFIER

Figure 3. Proposed framework.

We developed a fusion-based ensemble model in this investigation, wherein we used six ensemble algorithms, i.e., XGBoost, Adaboost, LightGBM, GB, CatBoost, and Hamming Gradient Boosting.

Here, we implemented the two most advanced techniques for avoiding overfitting issues during training the models. We took three different datasets and seven input features (detailed description given in Section 3.2). In any model building, process data pre-processing takes an essential role in generating a better predictive model. Thus, in our experiment, we went through the data pre-processing steps; this is elaborated upon in Section 3. After data pre-processing, we considered our models, which were to be trained. During our model building process, we used the stacking framework for the hybridisation of models. In a stacking framework, it takes two steps to develop a hybrid model. Thus, in step 1, it first finds out what base-level classifiers there are; using the base-level classifiers can prepare a training dataset for the second level. Thus, here, we took all our above-mentioned ensemble models as base level classifiers that are labelled as the level-1 classifier.

However, the real challenge came to our notice when we trained the level-1 classifiers with our seven input features. When we trained our base-level classifiers, we simply obtained overfitted models. Thus, to avoid overfitting issues for level-1 models, we implemented the dimensional reduction technique. Different types of dimensional techniques are present; after the successful implementation of this technique, we found the LDA technique was the perfect one for avoiding overfitting in our level-1 models. After implementing the dimensional reduction technique, we successfully developed six ensemble models whose results are given in Tables 7–11.

These base-level classifiers are used as a training dataset for second-level classifiers, a meta-classifier. However, again, another challenge came to our attention in that when we trained the meta classifiers with the help of level-1 classifiers, again, we faced overfitting issues. Thus, to avoid overfitting in our meta-classifier, we implemented the cross-validation technique. Therefore, after the implementation of the cross-validation technique, not only did we avoid overfitting, but we also obtained a generalised model. Tables 13, 15, 17, 19, and 21 and Tables 12, 14, 16, 18, and 20 show the performance level of our meta classifier with cross-validation and without cross-validation, respectively.

5. Results and Discussion

In this segment, we focus on our experimental results during our development process.

5.1. Performance of Base-Level Classifiers

Here, we implemented six ensemble-based boosting classifiers, namely, XGBoost, Adaboost, Gradient Boosting, LightGBM, CatBoost, and HistoGradient Boosting.

These ensemble models were trained with the three datasets, whose accuracy measures are given below in Tables 7–11.

Table 7. The forecast performances of the base level classifiers of DJIA index.

	XGBoost	AdaBoost	GradientBoost	LGBM	CatBoost	Histogr. Boost
Training accuracy	95.33	94.70	94.3	94.27	93.93	94.42
Testing accuracy	93.18	92.99	92.71	92.43	92.99	93.09
Accuracy dif.	2.15	1.71	1.62	1.84	0.94	1.33

Table 8. The forecast performances of the base level classifiers of HSI.

	XGBoost	AdaBoost	GradientBoost	LGBM	CatBoost	Histogr. Boost
Training accuracy	97.24	96.83	96.83	96.48	95.63	96.64
Testing accuracy	95.16	94.97	95.06	95.25	94.21	94.78
Accuracy dif.	2.08	1.86	1.77	1.23	1.42	1.86

Table 9. The forecast performances of the base level classifiers of S&P-500 index.

	XGBoost	AdaBoost	GradientBoost	LGBM	CatBoost	Histogr. Boost
Training accuracy	97.36	96.45	96.09	95.48	95.54	95.04
Testing accuracy	95.01	95.34	94.85	94.43	95.18	94.43
Accuracy dif.	2.35	1.11	1.24	1.05	0.36	0.61

Table 10. The forecast performances of the base level classifiers of DAX PERFORMANCE index.

	XGBoost	AdaBoost	GradientBoost	LGBM	CatBoost	Histogr. Boost
Training accuracy	90.78	90.06	92.06	88.70	89.32	89.40
Testing accuracy	88.94	89.08	88.50	87.98	89.18	89.37
Accuracy dif.	1.84	0.98	3.56	0.72	0.14	0.03

Table 11. The forecast performances of the base level classifiers of NIKKEI 225 index.

	XGBoost	AdaBoost	GradientBoost	LGBM	CatBoost	Histogr. Boost
Training accuracy	91.44	90.54	92.50	90.11	90.63	90.66
Testing accuracy	89.33	89.30	88.69	88.55	89.42	89.30
Accuracy dif.	2.11	1.24	3.81	1.56	1.21	1.36

We remarked that we used six ensemble boosted models whose training accuracy and test accuracy are given by looking at Table 7 above. Hence, the model XGBoost showed a training accuracy of 95.33 and a testing accuracy of 93.18. The model Adaboost showed a training accuracy of 94.70 and testing accuracy of 92.99, and Gradient Boosting showed a training accuracy of 94.33 and testing accuracy of 92.71; LightGBM showed a training accuracy of 94.27 and testing accuracy of 92.43; CatBoost showed a training accuracy of 93.93 and testing accuracy of 92.99; and finally, the HistoGradient Boosting model showed a training accuracy of 94.42 and testing accuracy of 93.09.

Observing the cells of accuracy difference shows the difference in accuracy between the training accuracy set and the testing accuracy set. For our information purpose, it is always a good practice that develops a model as much as a generalised model. A generalised model is a model with little or no difference between the training and testing accuracies. Hence, Table 7 shows a small variation, i.e., 0.94 to 2.15, between training and testing accuracy; the less indifference in the model is more common. Here, our CatBoost and HistoGradient boosting models were more generalised than the other four models. After this, these six models' predictive outputs were used as input features for the meta-classifier.

We remarked that we used six ensemble boosted models whose training accuracy and test accuracy are given in Table 8. The model XGBoost showed a training accuracy of 97.24 and a testing accuracy of 95.16; the model Adaboost showed a training accuracy of 96.83 and a testing accuracy of 94.97; Gradient Boosting showed a training accuracy of 96.83 and testing accuracy of 95.06; LightGBM shows training accuracy 96.48 and testing accuracy 94.21; CatBoost showed a training accuracy of 95.63 and testing accuracy of 94.21; and finally, the HistoGradient Boosting model showed a training accuracy of 96.64 and testing accuracy of 94.78. From here, all these models were ready to train our meta-classifier, which was our primary goal in terms of developing a generalised model.

As shown in Table 9, we found six models were developed whose training and testing accuracy were given. The model XGBoost showed a training accuracy of 97.36 and a testing accuracy of 95.01; the model Adaboost showed a training accuracy 96.45 and a testing accuracy of 95.34; Gradient Boosting showed a training accuracy of 96.09 and testing accuracy of 94.85; LightGBM showed a training accuracy of 95.48 and testing accuracy of 94.43; CatBoost showed a training accuracy of 95.54 and testing accuracy of 95.18; and finally, the HistoGradient Boosting model showed a training accuracy of 95.04 and a testing accuracy of 94.43. From here, all these models were ready to train our meta-classifier, which was our primary goal in terms of developing a generalised model.

We used six ensemble boosted models whose training accuracy and test accuracy are shown in Table 10. The model XGBoost showed a training accuracy of 90.78 and a testing accuracy of 88.94; the model Adaboost showed a training accuracy of 90.06 and a testing accuracy of 89.08; Gradient Boosting showed a training accuracy of 92.06 and a testing accuracy of 88.50; LightGBM showed a training accuracy of 88.70 and a testing accuracy of 87.98; CatBoost showed a training accuracy of 89.32 and a testing accuracy of 89.18; and finally, the HistoGradient Boosting model showed a training accuracy of 89.40 and a testing accuracy of 89.37. Here, the HistoGradient Boosting provided the lower accuracy difference, i.e., 0.03 whereas the Gradient Boost provided the highest accuracy difference, i.e., 3.56.

We used six ensemble boosted models whose training accuracy and test accuracy are given in Table 11. The model XGBoost showed a training accuracy of 91.44 and a testing accuracy of 89.33; the model Adaboost showed a training accuracy of 90.54 and a testing accuracy of 89.30; Gradient Boosting showed a training accuracy of 92.50 and a testing accuracy of 88.69; LightGBM showed a training accuracy of 90.11 and a testing accuracy of 88.55; CatBoost showed a training accuracy of 90.63 and a testing accuracy of 89.42; and finally, the HistoGradient Boosting model showed a training accuracy of 90.66 and a testing accuracy of 89.30. Here, the CatBoost provided the lowest accuracy difference, i.e., 1.21, whereas the Gradient Boost provided the highest accuracy difference, i.e., 3.81.

5.2. Performances of Fusion-Based Meta-Classifiers

In this section, we attempt find out the best combination of meta-classifiers. In the development of the meta-classifiers, we used the stacking cross-validation technique to combine level-1 predictive outputs and finally developed the metaclassifiers. The aim of our experiment was to find out a generalised model. The model was generalised when the training accuracy and testing accuracy were both are close to each other.

5.2.1. Performances of DJIA Index

Table 12 shows meta-classifiers' structure without cross-validation and Table 13 shows the construction of meta-classifiers with cross-validation.

Table 12. Construction of predictive meta-classifiers without cross-validation of DJIA index.

Models	Training Accuracy	Testing Accuracy	Accuracy	Difference
Meta-XGBoost	95.51	93.37	93.37	2.14
Meta-AdaBoost	95.33	93.28	93.28	2.05
Meta-G.B	95.48	93.18	93.18	2.3
Meta-LightGBM	95.45	93.28	93.28	2.19
Meta-CatBoost	95.11	93.28	93.28	1.84
Meta-H.G.boost	95.48	93.37	93.37	2.11

Table 13. Construction of predictive meta-classifiers with cross-validation of DJIA index.

Models	Training Accuracy	Testing Accuracy	Accuracy	Difference
Meta-XGBoost	93.99	92.99	93	1
Meta-AdaBoost	93.33	92.71	92.72	0.62
Meta-GB	94.55	93.09	93.09	1.46
Meta-Light GBM	93.33	93.27	93.28	0.06
Meta-CatBoost	93.96	92.99	93.00	0.97
Meta-H.G.Boost.	93.96	93.46	93.46	0.5

By observing Tables 12 and 13, we can say that the accuracy value of all models in Table 12 without C.V and in Table 13 with C.V seemed to be the same, with a negligible difference. However, when we came to find a generalised model with good predictive power, then Table 13 with cross-validation provides a promising result.

Table 13 found the six meta-classifiers, i.e., XGBoost, AdaBoost, GB, LightGbm, CatBoost, and HistoGradient Boosting. Out of the six meta classifiers, some of them had good accuracy. Some of them were better, but the LightGBM and HistoGradient Boosting provided good, promising results as they tend to prove themselves as generalised models with good predictive power. The model was generalised when the training accuracy and testing accuracy results were nearer to each other with a small difference or no difference. The meta classifier LightGBM provided training accuracy of 93.33 and testing accuracy of 93.27, with a negligent difference, i.e., 0.06. The meta-classifier HistoGradient Boosting provided a training accuracy of 93.96 and a testing accuracy of 93.46 with a difference of 0.5.

When we compared both models, we found that the meta classifier HistGrandBoosting showed the highest predictive accuracy, i.e., 93.46, whereas LightGbm showed 93.27 as its predictive accuracy; this means in terms of comparison of accuracies, HistGrandboosting proved itself to be more promising than LightGbm. However, if we were aiming for a generalised model with reasonable accuracy, then we can say that LightGbm is the perfect one because the training and testing accuracy difference was only 0.06, providing excellent promising power in prediction with both training and testing accuracy, inducing us to consider it as the best model among all.

5.2.2. Performance of HSI

In Table 14, we show the six meta classifiers that were developed without cross-validation technique; the purpose of this development was to bring about a conclusion that instead of using a single model, if we were to use a combination of models for prediction, then definitely our combined approach can predict better than an individual one.

Table 14. Construction of predictive meta-classifiers without cross-validation of HSI.

Models	Training Accuracy	Testing Accuracy	Accuracy	Difference
Meta-XGBoost	97.53	94.97	94.97	2.56
Meta-AdaBoost	96.90	95.16	95.16	1.74
Meta-G.B	97.53	94.97	94.97	2.56
Meta-LightGBM	97.46	94.97	94.97	2.49
Meta-CatBoost	96.71	95.44	95.44	1.27
Meta-H.G.boost	97.53	94.97	94.97	2.56

By looking at Table 1, we can see that the meta-classifiers were more generalised when the cross-validation technique was applied. For example, from this table, the Meta-H.G Boost and Meta-LightGBM showed remarkable performance, whereas the Meta-H.G Boost provided the highest testing accuracy, i.e., 95.35, and Meta-LightGBM provided 94.97 as the testing accuracy. However, by looking at the differences in accuracies, we found that the Meta-H.G Boost provided a difference of 0.31, whereas Meta-LightGBM provided a difference of 0.12, which was the slightest difference among all six meta-classifiers. Thus, the Meta-LightGBM is the more generalised model among all the models, which are shown in Table 15.

Table 15. Construction of predictive meta-classifiers with cross-validation of HSI.

Models	Training Accuracy	Testing Accuracy	Accuracy	Difference
Meta-XGBoost	95.41	94.30	94.30	1.11
Meta-AdaBoost	95.47	94.78	94.78	0.69
Meta-G.B	96.55	95.16	95.16	1.39
Meta-LightGBM	95.09	94.97	94.97	0.12
Meta-CatBoost	96.48	95.25	95.25	1.23
Meta-H.G.boost	95.66	95.35	95.35	0.31

5.2.3. Performance of S&P-500 Index

As shown in Table 16, we found six meta-classifiers that were developed without using the cross-validation technique. From Table 16, we can see that the meta-classifiers Meta-G.B, LightGBM, and CatBoost had testing accuracies of 95.01, and Meta-XGBoost was 94.76; Meta-AdaBoost had a training accuracy of 95.34 and Meta-H.G.boost had a training accuracy of 94.18.

Table 16. Construction of predictive meta-classifiers without cross-validation of S&P-500 index.

Models	Training Accuracy	Testing Accuracy	Accuracy	Difference
Meta-XGBoost	97.53	94.76	94.76	2.77
Meta-AdaBoost	96.87	95.34	95.34	1.53
Meta-G.B	97.47	95.01	95.01	2.46
Meta-LightGBM	97.36	95.01	95.01	2.35
Meta-CatBoost	97.45	95.01	95.01	2.44
Meta-H.G.boost	97.00	94.18	94.18	2.82

As shown in Table 17, we found that the meta-classifiers were more generalised when the cross-validation technique was applied. Table 17 shows that the Meta-AdaBoost, Meta-GB, and Meta-CatBoost showed the highest testing accuracies, i.e., at 95.43, whereas the Meta-LightGBM provided 94.68 as the testing accuracy. However, by looking at the accuracy difference, we found that Meta-XGBoost had a difference of 0.28, whereas Meta-LightGBM was 0.05, which was the slightest difference among all six meta-classifiers. Thus, Meta-LightGBM was the more generalised one among all.

Table 17. Construction of predictive meta-classifiers with cross-validation of S&P-500 index.

Models	Training Accuracy	Testing Accuracy	Accuracy	Difference
Meta-XGBoost	95.54	95.26	95.26	0.28
Meta-AdaBoost	96.17	95.43	95.43	0.74
Meta-G.B	96.81	95.43	95.43	1.38
Meta-LightGBM	94.73	94.68	94.68	0.05
Meta-CatBoost	96.28	95.43	95.43	0.85
Meta-H.G.boost	95.54	95.18	95.18	0.36

5.2.4. Performance of DAX Index

As shown in Table 18, we found six meta-classifiers that were developed without using the cross-validation technique. Here, by looking at the table, we found the training accuracy of all models appeared same with small differences, such as that we observe the testing accuracy, all model results looked the same and only a small variation was found. However, our result finding revealed that the accuracy difference was increased in DAX index, and therefore there was a chance of underfitting of the model.

Table 18. Construction of predictive meta-classifiers without cross-validation of DAX index.

Models	Training Accuracy	Testing Accuracy	Accuracy	Difference
Meta-XGBoost	92.41	88.26	94.76	4.15
Meta-AdaBoost	92.06	88.50	95.34	3.56
Meta-G.B	92.41	88.26	95.01	4.15
Meta-LightGBM	92.37	88.31	95.01	4.06
Meta-CatBoost	92.41	88.26	95.01	4.15
Meta-H.G.boost	92.41	88.26	94.18	4.15

As shown in Table 19, we found that the meta-classifiers were more generalised when the cross-validation technique was applied. In Table 19, by looking at the accuracy difference, we see that Meta-AdaBoost provided the highest difference, i.e., 0.47, whereas Meta-LightGBM provided 0.10, which was the smallest difference among all six meta-classifiers. Thus, the Meta-LightGBM was the more generalised one among all models. When we applied the cross-validation technique in our model, we found the models were more generalised and the training and testing accuracy difference was significantly less.

Table 19. Construction of predictive meta-classifiers with cross-validation of DAX index.

Models	Training Accuracy	Testing Accuracy	Accuracy	Difference
Meta-XGBoost	88.66	88.22	94.30	0.44
Meta-AdaBoost	89.70	89.23	94.78	0.47
Meta-G.B	88.92	88.60	95.16	0.32
Meta-LightGBM	85.10	85	94.97	0.10
Meta-CatBoost	96.47	96.25	95.25	0.22
Meta-H.G.boost	96.14	95.93	95.35	0.21

5.2.5. Performance of NIKKEI 225 Index

As is shown in Table 20, we found six meta-classifiers that were developed without using the cross-validation technique. Here, we found the training accuracy of all models appeared the same with a small difference such that that if we looked at the testing accuracy, all model results appeared the same, with only a small variation found. However, our result findings revealed that the accuracy difference was increased in NIKKEI 225 index when there was no cross-validation technique applied, which may lead to underfitting of a model.

Table 20. Construction of predictive meta-classifiers without cross-validation of NIKKEI 225 index.

Models	Training Accuracy	Testing Accuracy	Accuracy	Difference
Meta-XGBoost	92.65	88.67	94.76	3.98
Meta-AdaBoost	92.53	88.81	95.34	3.72
Meta-G.B	92.65	88.67	95.01	3.98
Meta-LightGBM	92.65	88.67	95.01	3.98
Meta-CatBoost	92.65	88.67	95.01	3.98
Meta-H.G.boost	92.65	88.67	94.18	3.98

As we can see in Table 21, we found that the meta-classifiers were more generalised when the cross-validation technique was applied. By looking at the accuracy difference, we found that the Meta-XGBoost provided the highest difference, i.e., 1.63, whereas Meta-LightGBM provided 0.22, which was the smallest difference among all six meta-classifiers. Thus, Meta-LightGBM was the more generalised one among all models. When we applied cross-validation technique in our model, we found the models were more generalised and the training and testing accuracy differences were significantly less.

Table 21. Construction of predictive meta-classifiers with cross-validation of NIKKEI 225 index.

Models	Training Accuracy	Testing Accuracy	Accuracy	Difference
Meta-XGBoost	90.84	89.21	94.30	1.63
Meta-AdaBoost	90.62	89.62	94.78	1
Meta-G.B	90.34	89.62	95.16	0.72
Meta-LightGBM	88.60	88.38	94.97	0.22
Meta-CatBoost	90.55	89.71	95.25	0.84
Meta-H.G.boost	89.58	88.67	95.35	0.91

5.3. Evalution Matrices of Meta-LightGBM

From the above experiment and discussion, we found the Meta-LightGBM model to be a more generalised model. Thus, in this paragraph, we explain the performance measurement technique of Meta-LightGBM of different datasets used in our experiment. If we are to access the accuracy of a classification model and the quality of prediction, then we must extract the classification report of that model. This classification report provides the results of metrices, i.e., recall, precision, and f1-score on the basis of class.

Table 22 shows the classification report of meta-classifier LightGBM, and the model had an AUC score of 93.36. When our model worked on the DJIA index, the performance of the metrics for class 0 was 0.93, recall was 0.93, and the F1-score was 0.93, whereas for class 1, the precision was 0.94, recall was 0.93, and the F1-score was 0.94. As we found, the AUC score was much closer to 1, which indicates that our model performance was good enough for this dataset.

Table 22. Classification report of Meta-LightGBM for DJIA index.

Precision	Recall	F1-Score
0.93	0.93	0.93
0.94	0.93	0.94

Table 23 shows the classification report of meta-classifier LightGBM, and the model had an AUC score of 95.43. When our model worked on the S&P 500 Index, the performance of the metrics of class 0 was 0.97, recall was 0.94, and the F1-dcore was 0.95, whereas for class 1, the precision was 0.94, recall was 0.97, and the F1-score was 0.95. As we see, the AUC score was much closer to 1, which indicates that our model performance was good enough for this dataset.

Table 23. Classification report of Meta-LightGBM in S&P 500 index.

Precision	Recall	F1-Score
0.97	0.94	0.95
0.94	0.97	0.95

Table 24 shows the classification report of Metaclassifier LightGBM, and the model had an AUC score of 95.26 for HSI. When our model worked on the HSI, the performance of the metrics of class 0 was 0.95, recall was 0.96, and the F1-score was 0.96, whereas for class 1, the precision was 0.96, recall was 0.94, and the F1-score was 0.95. As we see, the AUC score was much closer to 1, which indicates that our model performance was good enough for this dataset.

Table 24. Classification report of Meta-LightGBM for HSI.

Precision	Recall	F1-Score
0.95	0.96	0.96
0.96	0.94	0.95

Table 25 shows the classification report of meta-classifier LightGBM, and the model had an AUC score of 84.71 for DAX index. When our model worked on the DAX index, the precision metrics of class 0 was 0.86, recall was 0.87, and the F1-score was 0.87, whereas for class 1, the precision was 0.83, recall was 0.82, and the F1-score was 0.83. In comparison to the previous three datasets, the AUC scored somewhat less but was still good enough for a model performance.

Table 25. Classification report of Meta-LightGBM for DAX index.

Precision	Recall	F1-Score
0.86	0.87	0.87
0.83	0.82	0.83

Table 26 shows the classification report of meta-classifier LightGBM, and the model had an AUC score of 84.41 for NIKKEI 225 index. When our model worked on the NIKKEI 225 index, the precision metrics of class 0 was 0.90, recall was 0.94 and the F1-score was 0.92, whereas for class 1, the precision was 0.84, recall was 0.75, and the F1-score was 0.79. In comparison to the previous three datasets, the AUC scored somewhat less but was still good enough for a model performance.

Table 26. Classification report of Meta-LightGBM for NIKKEI 225 index.

Precision	Recall	F1-Score
0.90	0.94	0.92
0.84	0.75	0.79

The Figure 4 shows a graphical presentation of AUC scores, and Table 27 shows the AUC scores of the Meta-LightGBM model of different datasets.

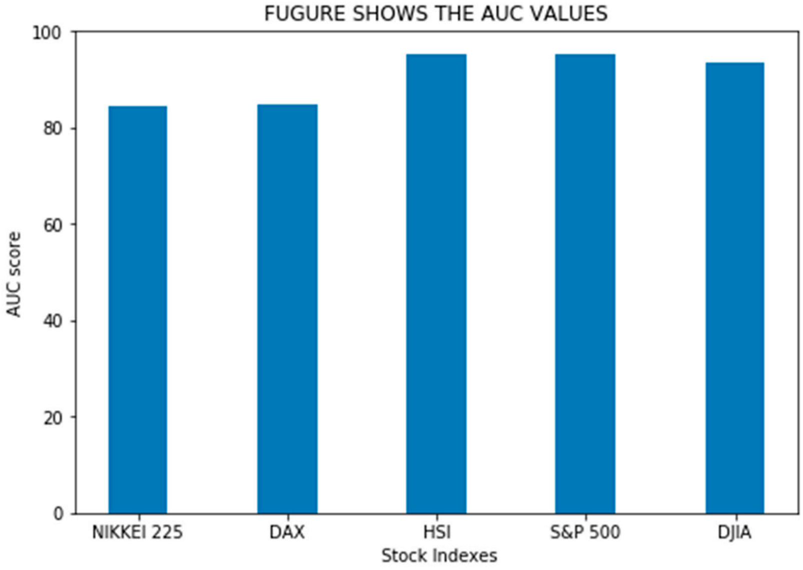


Figure 4. Bar graph shows the AUC score of Meta-LightGBM of all datasets.

Table 27. The AUC scores of all datasets of Meta-LightGBM model.

INDEX	AUC SCORE
NIKKEI 225	84.41
DAX	84.71
HSI	95.26
S&P 500	95.43
DJIA	93.36

5.4. Forecast Accuracy Comparison with Past Work

For a comparative study, we took the suggested model of Qiu et al. [66], i.e., WLSTM+Attention with our proposed model, i.e., Meta-LightGBM. As a benchmark, we took the mean absolute error (MAE) of both the models. The model which gave less MAE value was the best predictive model [40]. As shown in Table 28, our proposed model showed a lower MAE value than the WLSTM+Attention model. Thus, we can say the proposed model can perform better than the WLSTM+Attention model.

Table 28. Comparison of our suggested model with that past work.

Name of the Index	Our Proposed Model (Meta-LightGBM)	WLSTM+Attentation
	MAE	MAE
S&P 500	0.0481	0.1935
HSI	0.0464	0.2453
DJIA	0.0672	0.1569

5.5. Practical Implications

Every investor's 'dream' is to be able to properly anticipate the stock price and, as a result, compute the expected return. The proposed method has the ability to provide investors with useful information. Nowadays, the ML-based tools provide recommendations about specific stocks for the investors so that the investor gains a preliminary idea and can minimise the losses on investment. Artificial intelligence has a genuine effect on monetary exchange by mining significant data and providing modest and effectively accessible apparatuses that advantage everybody, not simply corporates. The speculation choices made by AI will be determined, exact and fair, not at all like those made by humans, who are evidently excessively enthusiastic about the exchange of securities. The proposed model may be used to develop new trading techniques or to manage stock portfolios by changing equities on the basis of trend predictions. It will help different financial institutions to gather information about the movement of the stock so that they can guide their investors to book profits and minimise the losses. Furthermore, it also provides a new direction for future researchers on how ensemble models are hybridised with different combinations of technical indicators and what the outcome will be when different parameters are tuned. Our experimental outcome revealed that the meta-classifier LightGBM had less error differences between training and testing accuracy that made our model more generalised. With the help of our model, any investor can minimise the losses during trading.

6. Conclusions and Future Scope

This study is based on fusion of ensemble models with technical indicators and extracted features to develop an evolutionary ensembled framework for forecasting stock market swings. During the model building process, our goal was to select a generalised model whose training and testing accuracy difference was minimal, instead of finding a model which provides the highest accuracy so that the investor and financial decision-makers can minimise the biased result using this proposed model. For our novel approach, we randomly selected five different indexes of four different countries, using seven features: four technical indicators, two derived features from the pre-existing elements, and the open price of the stock indices. Six ensemble models were used as base classifiers in layer one: XGBoost, LightGBM, AdaBoost, Gradient Boosting, CatBoost, and HistoGradient Boosting. During the individual modelling process (layer one modelling), we used a dynamic reduction algorithm, i.e., LDA, to generate probable input for the next layer classifier to prevent overfitting.

Our fact-finding results revealed that when we fused the ensemble models and developed a meta-classifier without using cross-validation technique, the fusion models training and testing accuracy difference was not good in comparison to the fusion models that were based on cross-validation technique. Sometimes, we found a fusion model provided better performance than a single predictive model, but the fusion models which were trained with cross validation technique showed promising results. During our experiment, we found that when the data size was increased, the performance of the model sometimes decreased. Our goal was to find a fusion model that offered a minimal overfitting and underfitting level, i.e., the training and testing accuracy difference should be significantly less. By looking at Table 29, we can summarise that the meta-classifier Meta-LightGbm is a model that had minimal training and testing accuracy difference in all the indexes. The recorded accuracy differences are shown in Table 29. Instead of only focusing on accuracy if we consider both accuracy and generalised acceptance, the meta-classifier Meta-LightGBM with cross-validation was shown to be more promising than all other predictive models. The run time of the model (Metta-LightGBM) on different datasets depended upon different constraints, such as the hardware configuration of the system in which the model was run though Googlecolab platform and the running time of our proposed model on S&P 500 index was 5 s, DJIA was 4 s, HSI was 4 s, NIKKEI 225 index was 40 s, and DAX index was

14 s in order to train the model. From this, we found that the execution time may differ as the number of datasets increases.

Table 29. Performance summary of Meta-LightGBM of all datasets.

	Training Accuracy	Testing Accuracy	Difference	Dataset
Meta-LightGBM	94.73	94.68	0.05	S&P-500 Index
Meta-LightGBM	95.09	94.97	0.12	HSI Index
Meta-LightGBM	93.33	93.27	0.06	DJIA Index
Meta-LightGBM	85.10	85.00	0.10	DAX Index
Meta-LightGBM	88.60	88.38	0.22	NIKKEI 225 Index

Finally, we can say that the fusion of ensemble models can be more generalised when we apply a cross-validation technique; not only will we improve the predictive accuracy, but also we will obtain a generalised model whose training and testing accuracy are very much closer to each other. Thus, we obtain a model which is neither overfitted nor underfitted.

Limitations and Future Work

Despite our proposed methodology's great predictive performance, there are some constraints which may be worked over in future. Our current study examined only one day ahead in terms of stock direction prediction, and therefore the study must be extended in future for long-term prediction. Our current study focused only on four countries' stock exchanges, but it may be examined and extended to study more stock exchanges of different countries. The research group did not consider any other information sources such as fundamental analysis or sentiment analysis, and thus this dataset must be considered in future experiments. In future work, we will use the factorisation machines and observe how they help in predicting the stock market behaviour [104–107].

Author Contributions: Conceptualisation, D.K.P., N.P., M.F.I., A.K.B. and J.S.; data curation, D.K.P. and N.P.; formal analysis, D.K.P., J.S., A.K.B. and M.F.I.; funding acquisition, M.F.I. and J.S.; investigation, D.K.P., N.P. and A.K.B.; methodology, D.K.P., N.P., M.F.I., A.K.B. and J.S.; project administration, M.F.I., A.K.B. and J.S.; resources, M.F.I., A.K.B. and J.S.; software, D.K.P. and N.P.; supervision, N.P., M.F.I., and A.K.B.; validation D.K.P., N.P. and A.K.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Sejong University Research fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analysed in this study. This data can be found on the 'Yahoo/Finance' portal. The data used to support the findings of this study are available from the first author upon request.

Acknowledgments: Jana Shafi would like to thank the Deanship of Scientific Research, Prince Sattam bin Abdul Aziz University, for supporting this work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CLAHE	Contrast-limited adaptive histogram equalisation
PSNR	Peak signal-to-noise ratio
PCA	Principal component analysis
ICA	Independent component analysis
FF	Firefly
HO	Hybrid optimisation
MAE	Mean absolute error

References

1. Adam, K.; Marcet, A.; Nicolini, J.P. Stock Market Volatility and Learning. *J. Financ.* **2016**, *71*, 33–82. [CrossRef]
2. Kunze, F.; Spiwoks, M.; Bizer, K. The usefulness of oil price forecasts—Evidence from survey predictions. *Manag. Decis. Econ.* **2018**, *39*, 427–446. [CrossRef]
3. Agustini, W.F.; Affianti, I.R.; Putri, E. Stock price prediction using geometric Brownian motion. *J. Phys. Conf. Ser.* **2018**, *974*, 012047. [CrossRef]
4. Dinh, T.A.; Kwon, Y.K. An empirical study on importance of modeling parameters and trading volume-based features in daily stock trading using neural networks. *IEEE Inform.* **2018**, *5*, 36. [CrossRef]
5. Mehdizadeh, S.; Sales, A.K. A comparative study of autoregressive, autoregressive moving average, gene expression programming and Bayesian networks for estimating monthly streamflow. *Water Resour. Manag.* **2018**, *32*, 3001–3022. [CrossRef]
6. Zhang, Y.; Song, W.; Karimi, M.; Chi, C.H. Kudreyko, AFractional autoregressive integrated moving average and finite-element modal: The forecast of tire vibration trend. *IEEE Access* **2018**, *6*, 40137–40142. [CrossRef]
7. Li, Q.; Cao, G.; Wei, X. Relationship research between meteorological disasters and stock markets based on a multifractal detrending moving average algorithm. *Int. J. Mod. Phys.* **2018**, *32*, 1750267. [CrossRef]
8. Petukhova, T.; Ojkic, D.; McEwen, B.; Deardon, R.; Poljak, Z. Assessment of autoregressive integrated moving average (ARIMA), generalized linear autoregressive moving average (GLARMA), and random forest (RF) time series regression models for predicting influenza a virus frequency in swine in Ontario, Canada. *PLoS ONE* **2018**, *13*, e0198313. [CrossRef]
9. Wang, D.; Liang, Z. Afuzzy set-valued autoregressive moving average model and its applications. *Symmetry* **2018**, *10*, 324. [CrossRef]
10. Rui, R.; Wu, D.D.; Liu, T. Forecasting stock market movement direction using sentiment analysis and support vector machine. *IEEE Syst. J.* **2018**, *13*, 60–770.
11. Nabipour, M.; Nayyeri, P.; Jabani, H.; Mosavi, A.; Salwana, E.S.S. Deep Learning for Stock Market Prediction. *Entropy* **2020**, *22*, 840. [CrossRef]
12. Jothimani, D.; Surendra, S.Y. Stock trading decisions using ensemble based forecasting models: A study of the Indian stock market. *J. Bank. Financ. Technol.* **2019**, *3*, 113–129. [CrossRef]
13. Zhong, X.; Enke, D. Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financ. Innov.* **2019**, *5*, 24. [CrossRef]
14. Yu, L.; Hu, L.; Tang, L. Stock Selection with a Novel Sigmoid-Based Mixed Discrete-Continuous Differential Evolution Algorithm. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 1891–1904. [CrossRef]
15. Shen, J.; Shafiq, M.O. Short-term stock market price trend prediction using a comprehensive deep learning system. *J. Big Data* **2020**, *7*, 66. [CrossRef] [PubMed]
16. Qiu, M.; Song, Y.; Akagi, F. The case of the Japanese stock market. *Chaos Solitons Fractals* **2016**, *85*, 1–7. [CrossRef]
17. Salim, L. A Technical Analysis Information Fusion Approach for Stock Price Analysis and Modeling. *World Sci. Res.* **2018**, *17*, 1850007.
18. Weng, B.; Martinez, W.; Tsai, Y.-T.; Li, C.; Lu, L.; Barth, J.R.; Megahed, F.M. Macroeconomic indicators alone can predict the monthly closing price of major U.S. indices: Insights from artificial intelligence, time-series analysis and hybrid models. *Appl. Soft Comput.* **2018**, *71*, 685–697. [CrossRef]
19. Fama, E.F. Random walks in stock market prices. *Financ. Anal. J.* **1965**, *21*, 55–59. [CrossRef]
20. Malkiel, B.G.; Fama, E.F. Efficient capital markets: A review of theory and 810 empirical work. *J. Financ.* **1970**, *25*, 383–417. [CrossRef]
21. Kahneman, D.; Tversky, A. Prospect Theory: An Analysis of Decision under Risk. *Econometrica* **1979**, *47*, 263–291. [CrossRef]
22. Glogowski, J. Keeping Up with the Quants: Your Guide to Understanding and Using Analytics by Thomas, H. Davenport and Jinho Kim. *J. Bus. Financ. Librariansh.* **2014**, *19*, 86–89. [CrossRef]
23. Rosenberg, B.; Reid, K.; Lanstein, R. Persuasive Evidence of Market Inefficiency. *J. Portf. Manag.* **1985**, *13*, 9–17. [CrossRef]
24. Sanjoy, B. Investment Performance of Common Stocks in Relation to Their Price-Earnings Ratios: A test of the Efficient Markets Hypothesis. *J. Financ.* **1977**, *32*, 663–682.
25. Nti, I.K.; Adekoya, A.F.; Weyori, B.A. A systematic review of fundamental and technical analysis of stock market predictions. *Artif. Intell. Rev.* **2020**, *53*, 3007–3057. [CrossRef]

26. Shiller, R.J. From efficient markets theory to behavioral finance. *J. Econ. Perspect.* **2003**, *17*, 83–104. [CrossRef]
27. Thaler, R.H. The end of behavioral finance. *Financ. Anal. J.* **1999**, *55*, 12–17. [CrossRef]
28. Hsu, P.H.; Kuan, C.M. Reexamining the profitability of technical analysis with data snooping checks. *J. Financ. Econom.* **2005**, *3*, 606–628. [CrossRef]
29. Nguyen, T.H.; Shirai, K.; Velcin, J. Sentiment analysis on social media for stock movement prediction. *Expert Syst. Appl.* **2015**, *42*, 9603–9611. [CrossRef]
30. Dreman, D.N.; Berry, M.A. Overreaction, Underreaction, and the Low-P/E Effect. *Financ. Anal. J.* **1995**, *51*, 21–30. [CrossRef]
31. Anbalagan, T.; Maheswari, S.U. Classifcation and prediction of stock market index based on fuzzy metagraph. *Procedia Comput. Sci.* **2014**, *47*, 214–221. [CrossRef]
32. Ghaznavi, A.; Aliyari, M.; Mohammadi, M.R. Predicting stock price changes of tehran artmis company using radial basis function neural networks. *Int. Res. J. Appl. Basic Sci.* **2014**, *10*, 972–978.
33. Agarwal, P.; Bajpai, S.; Pathak, A.; Angira, R. Stock market price trend forecasting using machine learning. *Int. J. Res. Appl Sci. Eng. Technol.* **2017**, *5*, 1673–1676.
34. Schmeling, M. Investor sentiment and stock returns: Some international evidence. *J. Empir. Financ.* **2009**, *16*, 394–408. [CrossRef]
35. Hu, Y.; Liu, K.; Zhang, X.; Su, L.; Ngai EW, T.; Liu, M. Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review. *Appl. Soft Comput.* **2015**, *36*, 534–551. [CrossRef]
36. Khan, H.Z.; Alin, S.T.; Hussain, A. Price prediction of share market using artifcial neural network "ANN". *Int. J. Comput. Appl.* **2011**, *22*, 42–47.
37. Ballings, M.; Dirk, V.P.; Nathalie, H.; Ruben, G. Evaluating multiple classifiers for stock price direction prediction. *Expert Syst. Appl.* **2015**, *42*, 7046–7056. [CrossRef]
38. Schapire, R.E. Explaining AdaBoost. In *Empirical Inference*; Schölkopf, B., Luo, Z., Vovk, V., Eds.; Springer: Berlin/Heidelberg, Germany, 2019.
39. Su, C.H.; Cheng, C.H. A hybrid fuzzy time series model based on ANFIS and integrated nonlinear feature selection method for forecasting stock. *Neurocomputing* **2016**, *205*, 264–273. [CrossRef]
40. Padhi, D.K.; Padhy, N. Prognosticate of the financial market utilizing ensemble-based conglomerate model with technical indicators. *Evol. Intel.* **2021**, *14*, 1035–1051. [CrossRef]
41. Marwala, T.; Hurwitz, E. *Artificial Intelligence and Economic Theory: Skynet in the Market*; Springer: London, UK, 2017.
42. Malkiel, B.G. The efficient market hypothesis and its critics. *J. Econ. Perspect.* **2003**, *17*, 59–82. [CrossRef]
43. Smith, V.L. Constructivist and ecological rationality in economics. *Am. Econ. Rev.* **2003**, *93*, 465–508. [CrossRef]
44. Nofsinger, J.R. Social mood and financial economics. *J. Behav. Financ.* **2005**, *6*, 144–160. [CrossRef]
45. Bollen, J.; Mao, H.; Zeng, X. Twitter mood predicts the stock market. *J. Comput. Sci.* **2011**, *2*, 1–8. [CrossRef]
46. Henrique, B.M.; Sobreiro, V.A.; Kimura, H. Literature review: Machine learning techniques applied to financial market prediction. *Expert Syst. Appl.* **2019**, *124*, 226–251. [CrossRef]
47. Zhang, Y.; Wu, L. Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network. *Expert Syst. Appl.* **2009**, *36*, 8849–8854.
48. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.
49. Chong, E.; Han, C.; Park, F.C. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Syst. Appl.* **2017**, *83*, 187–205. [CrossRef]
50. Avery, C.N.; Chevalier, J.A.; Zeckhauser, R.J. The CAPS prediction system and stock market returns. *Rev. Financ.* **2016**, *20*, 1363–1381. [CrossRef]
51. Prasad, V.V.; Gumparthi, S.; Venkataramana, L.Y.; Srinethe, S.; SruthiSree, R.M.; Nishanthi, K. Prediction of Stock Prices Using Statistical and Machine Learning Models: A Comparative Analysis. *Comput. J.* **2021**, 8. [CrossRef]
52. Zong, E. Forecasting daily stock market return using dimensionality reduction. *Expert Syst. Appl.* **2017**, *67*, 126–139. [CrossRef]
53. Hiransha, M.; Gopalakrishnan, E.A.; Menon, V.K.; Soman, K.P. NSE stock market prediction using deep-learning models. *Procedia Comput. Sci.* **2018**, *132*, 1351–1362.
54. Shynkevich, Y.; McGinnity, T.M.; Coleman, S.A.; Belatreche, A. Forecasting movements of health-care stock prices based on different categories of news articles using multiple kernel learning. *Decis. Support Syst.* **2016**, *85*, 74–83. [CrossRef]
55. Huang, B.; Huan, Y.; Da Xu, L.; Zheng, L.; Zou, Z. Automated trading systems statistical and machine learning methods and hardware implementation: A survey. *Enterp. Inf. Syst.* **2018**, *13*, 132–144. [CrossRef]
56. Usmani, M.; Adil, S.H.; Raza, K.; Ali, S.S.A. Stock market prediction using machine learning techniques. In Proceedings of the 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, 15–17 August 2016; pp. 322–327.
57. Atsalakis, G.S.; Valavanis, K.P. Forecasting stock market short-term trends using a neuro-fuzzy based methodology. *Expert Syst. Appl.* **2009**, *36*, 10696–10707. [CrossRef]
58. Chourmouziadis, K.; Chatzoglou, P.D. An intelligent short term stock trading fuzzy system for assisting investors in portfolio management. *Expert Syst. Appl.* **2016**, *43*, 298–311. [CrossRef]
59. Arévalo, A.; Niño, J.; Hernández, G.; Sandoval, J. High-Frequency Trading Strategy Based on Deep Neural Networks. In *Intelligent Computing Methodologies*; Huang, D.S., Han, K., Hussain, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; p. v9773.

60. Guresen, E.; Kayakutlu, G.; Daim, T.U. Using artificial neural network models in stock market index prediction. *Expert Syst. Appl.* **2011**, *38*, 10389–10397. [CrossRef]
61. Paik, P.; Kumari, B. Stock market prediction using ANN, SVM, ELM: A review. *IJETTCS* **2017**, *6*, 88–94.
62. Murekachiro, D. A review of artifcial neural networks application to stock market predictions. *Netw. Complex Syst.* **2016**, *6*, 3002.
63. Ampomah, E.K.; Qin, Z.; Nyam, G. Evaluation of Tree-Based Ensemble Machine Learning Models in Predicting Stock Price Direction of Movement. *Information* **2020**, *11*, 332. [CrossRef]
64. Leung, M.T.; Daouk, H.; Chen, A.-S. Forecasting stock indices: A comparison of classification and level estimation models. *Int. J. Forecast.* **2000**, *16*, 173–190. [CrossRef]
65. Omer, B.S.; Murat, O.; Erdogan, D. A Deep Neural-Network Based Stock Trading System Based on Evolutionary Optimized Technical Analysis Parameters. *Procedia Comput. Sci.* **2017**, *114*, 473–480.
66. Jiayu, Q.; Bin, W.; Changjun, Z. Forecasting stock prices with long-short term memory neural network based on attention mechanism. *PLoS ONE* **2020**, *15*, e0227222.
67. Xiao, Y.; Xiao, J.; Lu, F.; Wang, S. Ensemble ANNs-PSO-GA Approach for Day-ahead Stock E-exchange Prices Forecasting. *Int. J. Comput. Intell. Syst.* **2014**, *7*, 272–290. [CrossRef]
68. Huang, C.L.; Tsai, C.Y. A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. *Expert Syst Appl.* **2009**, *36*, 1529–1539. [CrossRef]
69. Ecer, F.; Ardabili, S.; Band, S.; Mosavi, A. Training Multilayer Perceptron with Genetic Algorithms and Particle Swarm Optimization for Modeling Stock Price Index Prediction. *Entropy* **2020**, *22*, 1239. [CrossRef] [PubMed]
70. Shah, D.; Isah, H.; Zulkernine, F. Stock Market Analysis: A Review and Taxonomy of Prediction Techniques. *Int. J. Financ. Stud.* **2019**, *7*, 26. [CrossRef]
71. Zhang, G.; Patuwo, B.E.; Hu, M.Y. Forecasting with artificial neural networks: The state of the art. *Int. J. Forecast.* **1998**, *14*, 35–62. [CrossRef]
72. Basa, S.; Kar, S.; Saha, S.; Khaidem, L.; De, S. Predicting the direction of stock market prices using tree-based classifiers. *N. Am. J. Econ.* **2019**, *47*, 552–567.
73. Milosevic, N. Equity Forecast: Predicting Long Term Stock Price Movement Using Machine Learning. *arXiv* **2016**, arXiv:1603:00751.
74. Choudhury, S.S.; Sen, M. Trading in Indian stock market using ANN: A decision review. *Adv. Model. Anal. A* **2017**, *54*, 252–262.
75. Boonpeng, S.; Jeatrakul, P. Decision support system for investing in stock market by using OAA-neural network. In Proceedings of the Eighth International Conference on Advanced Computational Intelligence (ICACI), Chiang Mai, Thailand, 14–16 February 2016; pp. 1–6.
76. Yang, R.; Yu, L.; Zhao, Y.; Yu, H.; Xu, G.; Wu, Y.; Liu, Z. Big data analytics for financial Market volatility forecast based on support vector machine. *Int. J. Inf. Manag.* **2019**, *50*, 452–462. [CrossRef]
77. Yun, K.K.; Yoon, S.W.; Won, D. Prediction of stock price direction using a hybrid GA-XG Boost algorithm with a three-stage feature engineering process. *Expert Syst. Appl.* **2021**, *186*, 115716. [CrossRef]
78. Yang, B.; Zi-Jia, G.; Wenqi, Y. Stock Market Index Prediction Using Deep Neural Network Ensemble. In Proceedings of the 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 3882–3887.
79. Wang, J.J.; Wang, J.Z.; Zhang, Z.G.; Guo, S.P. Stock Index Forecasting Based on a Hybrid Model. *Omega* **2012**, *40*, 758–766. [CrossRef]
80. Chenglin, X.; Weili, X.; Jijiao, J. Stock price forecast based on combined model of ARI-MA-LS-SVM. *Neural Comput. Applic.* **2020**, *32*, 5379–5388.
81. Tiwari, S.; Rekha, P.; Vineet, R. Predicting Future Trends in Stock Market by Decision Tree Rough-Set Based Hybrid System with Hhmm. *Int. J. Electron.* **2010**, *1*, 1578–1587.
82. Stankovic, J.; Markovic, I.; Stojanovic, M. Investment strategy optimization using technical analysis and predictive modeling in emerging markets. *Proc. Econ. Financ.* **2015**, *19*, 51–62. [CrossRef]
83. Atsalakis, G.S.; Valavanis, K.P. Surveying stock market forecasting techniques—Part II: Soft computing methods. *Expert Syst. Appl.* **2009**, *36*, 5932–5941. [CrossRef]
84. Bodas-Sagi, D.J.; Fernández-Blanco, P.; Hidalgo, J.I.; Soltero, D. A parallel evolutionary algorithm for technical market indicators optimization. *Nat. Comput.* **2012**, *12*, 195–207. [CrossRef]
85. de Oliviera, F.A.; Zarate, L.E.; de Azevedo Reis, M.; Nobre, C.N. The use of artificial neural networks in the analysis and prediction of stock prices. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Anchorage, AK, USA, 9–12 October 2011; pp. 2151–2155.
86. Nguyen, T.-T.; Yoon, S.A. Novel Approach to Short-Term Stock Price Movement Prediction using Transfer Learning. *Appl. Sci.* **2019**, *9*, 4745. [CrossRef]
87. Bisoi, R.; Dash, P.K.; Parida, A.K. Hybrid variational mode decomposition and evolutionary robust kernel extreme learning machine for stock price and movement prediction on daily basis. *Appl. Soft Comput.* **2018**, *74*, 652–678. [CrossRef]
88. Naik, N.; Mohan, B.R. Intraday Stock Prediction Based on Deep Neural Network. *Proc. Natl. Acad. Sci. USA* **2020**, *43*, 241–246. [CrossRef]
89. Salim, L. Intraday stock price forecasting based on variational mode decomposition. *J. Comput. Sci.* **2016**, *12*, 23–27.

90. Royo, R.C.; Francisco, K. Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data. *Expert Syst. Appl.* **2015**, *42*, 5963–5975. [CrossRef]
91. Kara, Y.; Boyacioglu, M.A.; Baykan, Ö.K. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert Syst. Appl.* **2011**, *38*, 5311–5319. [CrossRef]
92. Creighton, J.; Farhana, H.Z. Towards Building a Hybrid Model for Predicting Stock Indexes. In Proceedings of the IEEE International Conference on Big Data, Boston, MA, USA, 11–14 December 2017; pp. 4128–4133.
93. Singh, J.; Khushi, M. Feature Learning for Stock Price Prediction Shows a Significant Role of Analyst Rating. *Appl. Syst. Innov.* **2021**, *4*, 17. [CrossRef]
94. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
95. Yan, D.; Zhou, Q.; Wang, J.; Zhang, N. Bayesian regularisation neural network based on artificial intelligence optimisation. *Int. J. Prod. Res.* **2017**, *55*, 2266–2287. [CrossRef]
96. Dorogush, A.V.; Ershov, V.; Gulin, A. CatBoost: Gradient boosting with categorical features support. *arXiv* **2018**, arXiv:1810.11363.
97. Patel, J.; Shah, S.; Thakkar, P.; Kotecha, K. Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Syst. Appl.* **2015**, *42*, 259–268. [CrossRef]
98. Guryanov, A. Histogram-Based Algorithm for Building Gradient Boosting Ensembles of Piecewise Linear Decision Trees. In *Analysis of Images, Social Networks and Texts AIST*; Springer: Cham, Switzerland, 2019; p. 11832.
99. Xuan, H.; Lei, W.U.; YinsongYe, A. Review on Dimensionality Reduction Techniques. *Int. J. Pattern Recognit. Artif. Intell.* **2019**, *33*, 950017.
100. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
101. TA-LIB: Technical Analysis Library. Available online: www.ta-lib.org (accessed on 10 June 2021).
102. Cai, X.; Hu, S.; Lin, X. Feature extraction using Restricted Boltzmann Machine for stock price prediction. In Proceedings of the IEEE International Conference on Computer Science and Automation Engineering, Zhangjiajie, China, 25–27 May 2012; pp. 80–83.
103. Huang, W.; Nakamori, Y.; Wang, S.Y. Forecasting stock market movement direction with support vector machine. *Comput. Oper. Res.* **2005**, *32*, 2513–2522. [CrossRef]
104. Ticknor, J.L. A Bayesian regularized artificial neural network for stock market forecasting. *Expert Syst. Appl.* **2013**, *40*, 5501–5506. [CrossRef]
105. Knoll, J.; Stübinger, J.; Grottke, M. Exploiting social media with higher-order factorization machines: Statistical arbitrage on high-frequency data of the S&P 500. *Quant. Financ.* **2019**, *19*, 571–585.
106. Lawrence, R. Using neural networks to forecast stock market prices. *Univ. Manit.* **1998**, *333*, 2006–2013.
107. Stübinger, J.; Walter, D.; Knoll, J. Financial market predictions with Factorization Machines: Trading the opening hour based on overnight social media data. *Econ. Financ. Lett.* **2019**, *5*, 28–45. [CrossRef]