Applied Soft Computing 132 (2023) 109876

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Extending machine learning prediction capabilities by explainable AI in financial time series prediction

Taha Buğra Çelik [a],[*], Özgür İcan [b], Elif Bulut [a]

[a] Faculty of Economics and Administrative Sciences, Department of Business Administration, Ondokuz Mayıs University, Samsun, Turkey
[b] Faculty of Economics and Administrative Sciences, Department of International Trade and Logistics Ondokuz Mayıs University, Samsun, Turkey

ARTICLE INFO

Article history:
Received 12 July 2022
Received in revised form 23 November 2022
Accepted 25 November 2022
Available online 2 December 2022

Keywords:
Stock market prediction
Machine learning
Deep learning
Empirical mode decomposition
Explainable machine learning
Local interpretable model-agnostic explanations

ABSTRACT

Prediction with higher accuracy is vital for stock market prediction. Recently, considerable amount of effort has been poured into employing machine learning (ML) techniques for successfully predicting stock market price direction. No matter how successful the proposed prediction model is, it can be argued that there occur two major drawbacks for further increasing the prediction accuracy. The first one can be referred as the black box nature of ML techniques, in other words inference from the predictions cannot be explained. Furthermore, due to the complex characteristics of the predicted time series, no matter how sophisticated techniques are employed, it would be very difficult to achieve a marginal increase in accuracy that would meaningfully offset the additional computational burden it brings in. For these two reasons, instead of chasing incremental improvements in accuracy, we propose utilizing an "e**X**plainable **A**rtificial **I**ntelligence" (XAI) approach which can be employed for assessing the reliability of the predictions hence allowing decision maker to abstain from poor decisions which are responsible for declining overall prediction performance. If there would be a measure of how sure the prediction model is on any prediction, the predictions with a relatively higher reliability could be used to make a decision while lower quality decisions could be avoided. In this study, a novel two-stage stacking ensemble model for stock market direction prediction based on ML, empirical mode decomposition (EMD) and XAI is proposed. Our experiments have shown that, proposed prediction model supported with local interpretable model-agnostic explanations (LIME) achieved the highest accuracy of 0.9913 when only the most *trusted predictions* have been considered on KOSPI dataset and analogous successful results have been obtained from five other major stock market indices.

Code metadata

Permanent link to reproducible Capsule: https://doi.org/10.24433/CO.1813338.v1.

1. Introduction

Financial markets, in particular stock markets, allow investors and traders (practitioners who aim to earn excess returns from short-term price movements) to earn capital gains by making the right decisions. However, the price movements in the stock markets are highly nonlinear, and it is difficult to make the right decisions consistently. As a result of the rapid developments in ML and deep learning in recent years, great progress has been made in the field of stock market prediction. Since many novel methods and techniques are being developed and applied in a combined manner, classifying research in the stock market prediction literature is very difficult. Therefore, rather than making a definite classification, relevant research can be grouped according to which methods or techniques they suggest for directly increasing the success of prediction. In the literature, studies focusing on determining/optimizing the feature space used by the model are common in order to increase the prediction success. Zhou et al. [1] have employed multiple data sources including historical transaction data, technical indicators, stock posts, news and Baidu index to predict stock market direction with support vector machine (SVM). In feature-oriented studies, novel prediction models are frequently used by hybridizing with ML or deep learning methods. These studies focus on improving the features in order to increase the success of the prediction model. Persistent homology which is a method used in topological data analysis have employed by Ismail et al. [2] for obtaining

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

[*] Corresponding author.
E-mail addresses: tahabugra.celik@omu.edu.tr (T.B. Çelik), ozgur.ican@omu.edu.tr (Ö. İcan), elif@omu.edu.tr (E. Bulut).

https://doi.org/10.1016/j.asoc.2022.109876
1568-4946/© 2022 Elsevier B.V. All rights reserved.

useful inputs for stock market prediction model. Yun et al. [3] have proposed a hybrid GA-XGBoost prediction model which employs genetic algorithms (GA) for feature selection. This approach which is an effort for increasing the predictive success of the model by focusing on the inputs are called feature engineering [4]. In order to learn the latent feature representation from stock prices, Zhang et al. [5] have proposed deep belief networks (DBN). Thakkar and Chaudhari [6] have utilized the term frequency–inverse document frequency (TF–IDF) to derive feature weight matrix from the historical stock market data and backpropagation neural network (BPNN). Hao and Gao [7] have proposed multiple time scale feature learning to predict the price trend of the stock market index. In order to learn complementary features from different sources of historical price and text data, Liu et al. [8] have proposed a recurrent convolutional neural kernel. Yang et al. [9] have combined convolutional neural network (CNN) for feature extraction and a long short-term memory (LSTM) network for prediction.

Another approach focusing on features is dimensionality reduction such as variational auto-encoders (VAE) and principal components analysis (PCA) to improve the computational efficiency of prediction models via reducing the complexity of feature set [10,11]. In addition, the use of technical analysis (TA) indicators as inputs is quite common in the literature. Yang et al. [12] have combined TA with group penalized logistic regressions to predict up and down trends of stock prices. Patel et al. [13] have used ten technical indicators in order to determine trend of the data. Nabipour et al. [14] have also used ten technical indicators as continuous inputs and followed an approach of converting these into binary data before their analysis. Lee et al. [15] have made predictions with attention-based BiLSTM fed by TA indicators. Besides dimensionality reduction, another important approach is decomposition of the complex time series into more manageable sub-components. One of the most commonly used decomposition approaches is empirical mode decomposition (EMD) [16]. EMD have been applied by Xu and Tan [17] in order to decompose stock price and, sub components have been predicted by a temporal attention LSTM. Zhou et al. [18] have introduced EMD and factorization machine based neural network to predict the stock market trend. Jin et al. [19] have proposed sentiment analysis combined with LSTM and EMD. There is an also more advanced decomposition method developed as an alternative to EMD called variational mode decomposition (VMD) [20]. Utilization of VMD along with deep learning and machine learning models for stock market prediction has been proved to provide successful results [21–23]. There are also other decomposition methods such as singular spectrum analysis (SSA), empirical wavelet transform (EWT), ensemble EMD (EEMD) employed along with prediction models [24–26].

In the studies mentioned above, in order to increase the prediction performance of a single prediction model, it has been commonly observed that additional methods and techniques are being hybridized. In the literature, there is an obvious trend of employing multiple prediction models within various settings. One such approach is called ensemble learning which is a meta approach combining multiple prediction models in order to produce a better composite prediction model. In order to achieve high prediction accuracy, consulting multiple models is a widely used approach and known as multi-classifiers, multi-classifiers combination and mixture of experts in the literature (for more details see [27]).

Ensemble methods are divided into three main categories called *bagging*, *stacking*, and *boosting*. Bagging or bootstrap aggregation include a diverse group of prediction models which are trained with different training subsets generated using random sampling. The predictions made by the ensemble members are then given to a combination scheme (such as voting, averaging or any set of other rules) to produce a final prediction value while stacking approach combines outputs of a group of prediction models as inputs to another prediction model in order to achieve higher prediction accuracy [28]. Although multiple layers of models can be utilized, two-level hierarchy is more common. Finally, in boosting ensemble models, training data is iteratively changed to focus on the misclassified instances in the previous fits. In summary, boosting approach is based on the idea of correcting prediction errors [29,30].

Plentiful of methods and techniques to increase the success of stock market prediction models exist in the literature, and new ones are constantly being developed. One can conclude that prediction success of the mentioned models in contemporary literature is already significantly high. Regardless of the method being used, it becomes more and more difficult, or even impossible, to exceed the prediction accuracy rates achieved in some of these studies such as [3] thanks to the pattern recognition capabilities of machine learning methods. Reporting of close prediction successes with different combinations of new or existing techniques indicates that it is increasingly difficult to make further progress. Therefore, instead of dealing with new techniques which increase the computational complexity, the alternative direction of how to increase the reliability of the predictions of already established techniques arises. In other words, an approach that would enable a prediction model to avoid some percentage of the failed predictions would indirectly increase the overall accuracy to a significantly higher level by merely proceeding with trustworthy predictions. The only drawback of this approach is that only certain combinations of the feature set would result in actual predictions as rest of them would lead to unreliable ones hence would be ignored. As a result of this, making predictions with very high accuracy rates would bring the cost of being abstained from making predictions for certain periods of the entire forecasting horizon. In the contemporary literature, we see that such efforts are examined under the name of explainable artificial intelligence (XAI). Recently, two different approaches have come to the fore within the scope of XAI. One of them is local interpretable model-agnostic explanations (LIME) [31], which allows any model to be interpreted by explaining each prediction on an instance-by-instance basis. Another common method is SHapley Additive Explanations (SHAP) [32].

In this study, initially, a two-stage stacking ensemble prediction model has been developed in order to predict the daily stock market closing price direction. Instead of feature engineering, we have preferred a decomposition approach by employing EMD technique. The reason for preferring data decomposition has been clearly explained in Section 2.4. It has been put forward that EMD clearly facilitates the prediction task of the ensemble model. These decomposed series, also known as intrinsic mode functions (IMF) are simply sub-components of the original time series. In the first stage, each IMF has been predicted with two distinct ANN models. The first ANN model has been used for predicting each IMF's continuous value. In other words, it predicts its quantitative values (regression prediction), so it is called "ANN Regression" (ANNR) for short. On the other hand, the latter one has been used for classification of the direction (upward and downward) and since this ANN model has been designed as a classifier model, it has been called ANNC standing for "ANN Classifier". Needless to say that, for all IMFs, these two ANN models have been trained separately and model objects have been saved for predictions. The reason behind employing these two distinct ANN models is that pre-experimental results have asserted the superiority of ANNC for predicting certain IMFs but ANNR for the rest. For this reason, the predictions of these two models have been combined in order to exploit their relative strengths in the first stage and their combined predictions have been fed to a third prediction model in the second stage. The third model in the second stage has been selected based on the comparative performances of different algorithms, namely random forest (RF) and extreme gradient boosting (XGBoost) and since RF has been provided better results, it has been preferred over XGBoost. Moreover, this prediction model has been utilized as a classifier (upward and downward direction prediction) in order to predict the direction of the original time series. To sum up, overall architecture is one of the possible stacking ensemble model configurations among myriad combinations. Finally, based on the preferred techniques our proposed ensemble model has been named as EMD-ANNRC-RF.

The point that we put forward as a contribution to the literature of this study is to investigate what contribution the LIME algorithm, one of the explainable AI approaches, can offer if it is integrated into an ensemble classifier. When the literature is examined, no research has been found to improve the prediction success by investigating the reliability of the predictions made by the model in financial time series forecasting with the "model explainability" approach. We think that a study in which such an intensively studied subject (stock market prediction) is handled with these dimensions will make a direct contribution to the literature. The possibility of integration of XAI approaches, which can be considered new in the literature, into direction prediction models should be tested and further developed in further studies. Therefore, in this study, we have presented how to benefit from the LIME algorithm, which can be integrated into prediction models. In this context, it would be helpful to explain the details of this integration.

In the literature on XAI, independent of the prediction model being used, there is an approach which tries to explain each prediction (instance by instance) called LIME. LIME explains the predictions of any classier in an interpretable and faithful manner. It offers explanations by learning an interpretable model locally around the prediction [31]. LIME adds a feature to the prediction model by providing class probabilities (since the prediction task in this study is upward and downward prediction) for each prediction. In the second stage of proposed prediction procedure, which constitutes the most important part of the study, the LIME algorithm has been integrated to the RF model. As mentioned above, by giving the outputs of ANNR and ANNC as inputs to the RF model, the upward and downward movement direction of six major stock market indices has been predicted. The main motivation of this study emerges at this stage. During the exploration stage of this research, it has been discovered that the RF model makes predictions with superior accuracy for certain output values of ANNR and ANNC. In the usual process, when any input instance is given to the RF model, it makes one of the 0 or 1 predictions. However, after the LIME algorithm is implemented, a probability is assigned to each of the 0 and 1 class labels. Thus, the probabilities of each class prediction made by the RF model can be calculated with LIME. For example, suppose that for any input set, the probability of 0.10 for the class 0 and 0.90 for the class 1 is obtained. These calculated probabilities for the classes are obtained for each prediction in the test set. Here, the class probabilities are utilized as the reliability level for the predictions. If one of the class probabilities is high enough for the decision maker, then he/she trust that prediction and make decision based on the outcome. According to the previous example, if 0.80 is high enough to be trusted, then the decision maker arrives a decision in favor of 1 class label. Here, 0.80 is the reliability level for the decision maker. On the other hand, If the decision maker would set the reliability level as 0.91, then he/she would hesitate to make any decision since 0.90<0.91. In other words, reliability condition is not satisfied. If the reliability level is set to 0.50, we simply obtain the predictions made by the RF algorithm alone as if LIME is integrated. However, as the reliability level is increased, some predictions will be avoided as the reliability condition will not be met for some of them, but a higher accuracy will be expected for trusted predictions. In other words, predictions cannot be made for all of the test set, since those with low reliability levels will be avoided. In order to test this idea, the final increase in the accuracy rate for all reliability levels ranging from 50% to 100% and the decrease in the number of predictions (accuracy and number of trusted predictions tradeoff) have been revealed as a result of the experiments. To the best of our knowledge, the implementation of LIME algorithm to a prediction to such a classifier model as we propose here has not been previously proposed in the relevant literature.

It is useful for the reader to summarize the next parts of the work. Section 2.1 to Section 2.4 describe the base methods and techniques used in the forecasting model. In Section 2.4, the details of the model we have proposed are shared. While the experimental results and evaluations are included in Section 3, discussions and conclusion and future directions are given in Section 4 and Section 5 respectively.

2. Prediction methods and framework

2.1. Empirical mode decomposition

Empirical mode decomposition (EMD) is a method for analyzing nonlinear and non-stationary data which is developed by [16]. Any complicated dataset can be decomposed into finite number of 'intrinsic mode functions' (IMF) and the decomposition is based on the local characteristic time scale of the data so that it is applicable to nonlinear and non-stationary processes [16]. Decomposition is a separation of a signal into different components and it is usually used in data analysis if some information is wanted to be extracted that cannot be obtained when considering the data as a whole. Therefore, decomposing data allows analyzing the newly obtained components to gain new insight into the features inherent to the data. Each mode function represents a portion of the complete signal. For a given signal $X(t)$, EMD algorithm is implemented as follows:

1. Determine envelops of $X(t)$ which are defined by local maxima X_{up} and local minima X_{low} separately.
2. Once the extrema are identified, all the local maxima are connected by a cubic spline line, $X_{up}(t)$, as the upper envelope.
3. Repeat the procedure for the local minima to produce the lover envelope, $X_{low}(t)$.
4. Let m_t be the mean of the upper and lower envelope such that,

$$m_t = \left(X_{up}(t) + X_{low}(t)\right)/2 \tag{1}$$

5. Then the first component, $h_1(t)$,

$$h_1(t) = X(t) - m_t \tag{2}$$

6. After the first component obtained, replace $X(t)$ with $h_1(t)$ and repeat the whole procedure until the stopping criterion is satisfied where the stopping criterion is:

$$\sum_t \frac{\left(h_i(t) - X(t)\right)^2}{X(t)^2} < \epsilon \tag{3}$$

Here ϵ is a threshold value and it is usually close to 0.2. The IMF admits to well-behaved Hilbert transform. This decomposition is not based on a predetermined mathematical basis but the data itself dictates the decomposition and applicable to non-linear and non-stationary processes.

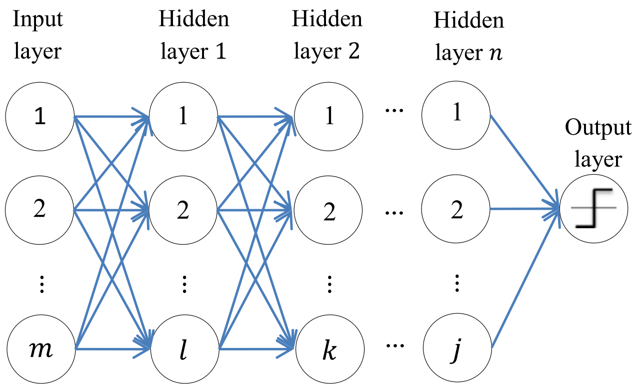

Fig. 1. General architecture of ANN models.

2.2. Artificial neural networks

The beginning of artificial neural networks (ANN) goes back to the computational model developed for neural networks called threshold logic by [33]. However, the algorithm that forms the basis of the multi-layer neural networks which enables the training of ANNs in its current sense is the backpropagation algorithm. ANN is a supervised machine learning algorithm that learns the mapping between an input and an output set. Basic ANN architecture has three *layers* and each layer consists of *nodes*. First layer is the input layer and each node refers to an input. The second layer, called as hidden layer, may include more than one layer. The last layer is called output layer and it produces the output of the model for each input instance. Based on the design, nodes are connected to each other but not necessarily all nodes are connected. The model is trained with a sample of data, to capture the relationship between inputs and outputs. In Fig. 1, a representative ANN model is depicted.

Both regressing prediction (prediction of continuous numeric values) and classification (prediction of class label) can be made with ANN. In this study, both these features of artificial neural network have been used. ANNR and ANNC models which have been mentioned before have two hidden layers with 100 nodes and using ReLU activation function. These two models have been trained for 300 epochs with a mini-batch size of 30 samples. The output layer of ANNR has a single node for predicting a numeric value with hyperbolic tangent activation function and trained to minimize the mean squared error (MSE) loss function using the Adam version of stochastic gradient descent. ANNC on the other hand, has sigmoid activation function in output layer with binary cross entropy loss function.

2.3. Local interpretable model-agnostic explanations (LIME)

Ribeiro et al. [31] proposed the LIME algorithm to provide explanations for individual predictions, allowing some degree of reliability for the predictions of any classifier or regressor. LIME provides interpretations for predictions locally for a given prediction. LIME decides whether a model is locally faithful regardless of the model and verifies how a model represents the features around a prediction. This attribution of LIME algorithm is known as local fidelity [34]. The explanation produced by LIME is obtained by the following [31]:

$$explanation\,(x) = \arg\min_{g \in G} \mathfrak{L}\,(f, g, \Pi_x\,(z)) + \Omega\,(g) \tag{4}$$

where x represents an instance and interpretable representation of an instance is a binary vector $x \in \{0, 1\}^{d'}$. Let G be a set of potentially interpretable models and $g \in G$, where g represents a machine learning model. The domain of g is $\{0, 1\}^{d'}$. The complexity of an interpretation of a model is $\Omega\,(g)$. For classification, $f\,(x)$ is the probability measure of x belong to a class. $\Pi_x\,(z)$ is proximity measurement between an instance z to x, so as to define locality around x.

2.4. Proposed model: Two-stage ensemble EMD-ANNRC-RF

There are two major prediction approaches in stock market prediction literature. The first approach is to predict actual price levels of time series also known as regression prediction. In this approach it is a common practice to compare the predicted values with observed ones with respect to known evaluation metrics such as root mean squared error (RMSE), mean squared error (MSE), mean absolute error (MAE) and mean absolute percentage error (MAPE). Recently, directional prediction accuracy (also known as hit-rate) for evaluating prediction performance is also becoming more common since making accurate movement direction prediction is vital for successful stock market predictions. Accuracy is measured as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

where TP true positive predictions, TN true negative predictions, FP false positive predictions and FN false negative predictions. Other performance evaluation metrics which are commonly employed along with accuracy are *precision*, *recall*, *F1-score* and *area under the receiver operating characteristic curve* (ROC-AUC). These evaluation metrics are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 - score = 2\frac{(Precision)\,(Recall)}{Precision + Recall} \tag{8}$$

$$ROCAUC = \int_0^1 \frac{TP}{TP + FN}\, d\frac{TP}{TP + FP} \tag{9}$$

The second approach in predicting stock market is based on time series directional prediction, also known as time series classification. In order to conduct stock market direction classification, target data is commonly labeled as 1 and 0 standing for upward and downward movements as respectively. Therefore the prediction procedure turns into a classification problem and a classifier model can be trained with respect to input and target set consisting of ones and zeros. These two major prediction approaches have their own advantages and disadvantages. In regression prediction, actual data values are predicted so that the prediction results offer "more information" compared to classification. On the other hand, results of time series classification is easier to interpret relative to regression prediction. Moreover, classification prediction is easier to handle for the prediction model as long as the target data and the feature set are related since the target data set is Boolean in contrast to continuous target sets of regression prediction. In this paper, in order to predict stock market price data, a novel two-stage stacked ensemble prediction framework is designed by exploiting the advantages of both approaches.

Before building our proposed ensemble model, data preprocessing and scaling of daily closing prices of six selected major stock market indices namely Standard & Poor's 500 Index (SP500), Nikkei 225 Index (NI225), Borsa Istanbul 100 Index (XU100), Korea Composite Stock Price Index (KOSPI), Deutscher Aktien Index (DAX) and Financial Times Stock Exchange 100 Index (FTSE100) have been conducted. As the first step of building the ensemble

Table 1
Prediction accuracies of each of the IMFs with ANNR and ANNC.

Index	Model	IMF0	IMF1	IMF2	IMF3	IMF4	IMF5	IMF6	IMF7	IMF8
SP500	ANNR	0.7330	**0.8945**	**0.9736**	**0.9744**	**0.9936**	**0.9968**	**0.9960**	**1.0000**	NA
	ANNC	**0.8826**	0.8610	0.8754	0.9169	0.9361	0.7995	0.4657	**1.0000**	NA
NI225	ANNR	0.8248	0.8725	**0.9729**	**0.9926**	**0.9942**	**0.9984**	**0.9959**	**0.9984**	NA
	ANNC	**0.8924**	**0.8915**	0.9441	0.9499	0.9745	0.9573	0.8094	0.9326	NA
XU100	ANNR	0.7894	**0.8991**	**0.9792**	**0.9888**	**0.9920**	**0.9984**	**0.9960**	**0.9976**	NA
	ANNC	**0.8928**	0.8888	0.9240	0.9400	0.9048	0.8776	0.8472	0.9960	NA
KOSPI	ANNR	0.8103	**0.8896**	**0.9657**	**0.9926**	**0.9943**	**0.9943**	**0.9984**	**1.0000**	NA
	ANNC	**0.8840**	0.8856	0.9371	0.9322	0.9175	0.7042	0.4608	**1.0000**	NA
DAX	ANNR	0.8211	0.8847	**0.9833**	**0.9873**	**0.9944**	**0.9976**	**0.9968**	**1.0000**	NA
	ANNC	**0.9039**	**0.8864**	0.9055	0.9436	0.9357	0.9325	0.6958	**1.0000**	NA
FTSE100	ANNR	0.7645	0.8789	**0.9680**	**0.9924**	**0.9933**	**0.9958**	**0.9975**	**0.9975**	**1.0000**
	ANNC	**0.8840**	**0.8950**	0.9168	0.9303	0.9529	0.9815	0.6395	0.3782	**1.0000**

model, time series data has been decomposed by employing EMD technique in order to reduce complexity of the original series so that data sets have been decomposed into more manageable sub-components (i.e. IMFs). Subsequently experiments have been conducted for the prediction of each of the IMF data sets. By the help of ANNR and ANNC, regression and classification predictions have been made respectively for each of IMFs. Our preliminary results are summarized in Table 1 below:

Prediction results show that, accuracy of classification predictions for the first IMF (IMF0) denoted by $h_0(t)$, is significantly better then regression predictions. Since $h_0(t)$ is the hardest part to predict for regression prediction, it is thought that improvement in prediction accuracy for $h_0(t)$ may contribute to the overall success of the prediction model of the original price series. On the other hand, ANNR prediction results of other IMFs are slightly better than ANNC predictions. Therefore, predicting all IMFs except $h_0(t)$ with regression based prediction model and predicting $h_0(t)$ with classification based prediction model is considered to be appropriate approach for increasing overall prediction accuracy. However, at this stage a problem arises. Prediction procedure of EMD-ANN model is traditionally made by summation of each predicted continuous valued IMFs so that the aggregated predictions of IMFs represent the prediction of original price series. On the other hand, classification prediction results of $h_0(t)$ are sigmoid function outputs (denoted by $\sigma(x)$, i.e. continuous values which is ranging between 0 and 1). ANN classification model labels predictions as 1 if $\sigma(x) > 0.5$ and 0 for $\sigma(x) \leq 0.5$. In this regard, summing classification predictions of $h_0(t)$ with the regression predictions of the rest of the IMFs is inappropriate and meaningless. In order to overcome this obstacle, a novel prediction experiment design has been proposed.

As described before, $h_0(t)$ series is predicted by ANNC. ANNC has been trained with four days lagged values[1] of $h_0(t)$ such that $h_0(t-1), \ldots, h_0(t-4)$ are fed as inputs and the movement direction of $h_0(t)$ which is denoted as $D_t(h_0(t))$ where

$$D_t(h_0(t)) = \begin{cases} 0, & \textit{if} \quad h_0(t) - h_0(t-1) \leq 0 \\ 1, & \textit{if} \quad h_0(t) - h_0(t-1) > 0 \end{cases} \tag{10}$$

as output. Since ANNC is a binary classifier, predictions, $\hat{C}_{y,t}$, are class labels that is $\hat{C}_{y,t} \in \{0, 1\}$ implied by classifier design of the artificial neural network. Activation function of ANNC is a *sigmoid function* also called as *squashing function* in machine learning terminology and denoted by $\sigma_t(x)$, which is a special form of the logistic function and ranges between 0 and 1. ANNC classifies inputs based on the predicted values of $\sigma_t(x)$ such that,

$$\hat{C}_{y,t} = \begin{cases} 0, & \textit{if} \quad \hat{\sigma}_t(x) \leq 0.5 \\ 1, & \textit{if} \quad \hat{\sigma}_t(x) > 0.5 \end{cases} \tag{11}$$

Explorative experiments for the final prediction step have shown that, predictions that have sigmoid function output closer to one or zero can more successfully predict upward and downward movements of original time series respectively. Based on this justification, $\hat{\sigma}_t(x)$ is kept as the output value of ANNC, in other words, no class label conversion has been made. On the other hand, all IMFs are also predicted by the ANNR model and aggregated as the conventional way to predict original price series for each of the t time period. Let $\hat{y}_t$ be the prediction at t time step, and *difference series* which is denoted as $d_t(\hat{y})$, where

$$d_t(\hat{y}) = \hat{y}_t - \hat{y}_{t-1} \tag{12}$$

is obtained for each day. Here $d_t(\hat{y})$ indicates the prediction direction magnitude of ANNR and indicates the *severity* of the increase or decrease in the successive predictions. The motivation behind obtaining $d_t(\hat{y})$ is analogous to using $\hat{\sigma}_t(x)$. Similarly, as $d_t(\hat{y})$ diverges from zero, upward and downward movement direction prediction accuracies are expected to increase. Stage-1 of the proposed model ends here and the outputs of this stage, $d_t(\hat{y})$ and $\hat{\sigma}_t(x)$, are considered to be the inputs of the ML model of Stage-2. The procedure and operations performed in Stage-1 in terms of training the prediction models, the related input dataset and the outputs of these models would be helpful. To sum up, ANNR and ANNC models have been trained in training set and by using these two models, predictions have been made for validation set to obtain $d_t(\hat{y})$ and $\hat{\sigma}_t(x)$ hence Stage-1 is completed. The outputs of these two models will become the inputs that would feed the prediction model of Stage-2. In the next stage a classifier, namely a random forest model, has been trained with $d_t(\hat{y})$ and $\hat{\sigma}_t(x)$ as inputs for obtaining movement direction of original price series, $D_t(y)$, at time step t as output where:

$$D_t(y_t) = \begin{cases} 0, & \textit{if} \quad y_t - y_{t-1} \leq 0 \\ 1, & \textit{if} \quad y_t - y_{t-1} > 0 \end{cases} \tag{13}$$

In final prediction step of Stage-2, in order to test aforementioned hypotheses, $d_t(\hat{y})$ and $\hat{\sigma}_t(x)$ have been obtained by the predictions of ANNR and ANNC respectively in the *test set*. Finally, $d_t(\hat{y})$ and $\hat{\sigma}_t(x)$ have been fed to random forest classifier which has 250 number of trees, 37 maximum depth of the tree to predict the movement direction of original price series for each t time step and $\hat{C}_{y,t} = \{0, 1\}$. In other words, predicted movement directions of original price series for each time step t have been obtained. Then the results have been compared with single EMD-ANNR [36] prediction model in terms of accuracy,

[1] We have tried different lag orders and four or five days seem to be appropriate as [21,35] suggested. It has been observed that using less than four days lag decreases prediction performance while no significant performance has been observed by employing more than five days. Furthermore it increases the amount of overall computation. Therefore, following [21], we have preferred to use four lays lagged values.

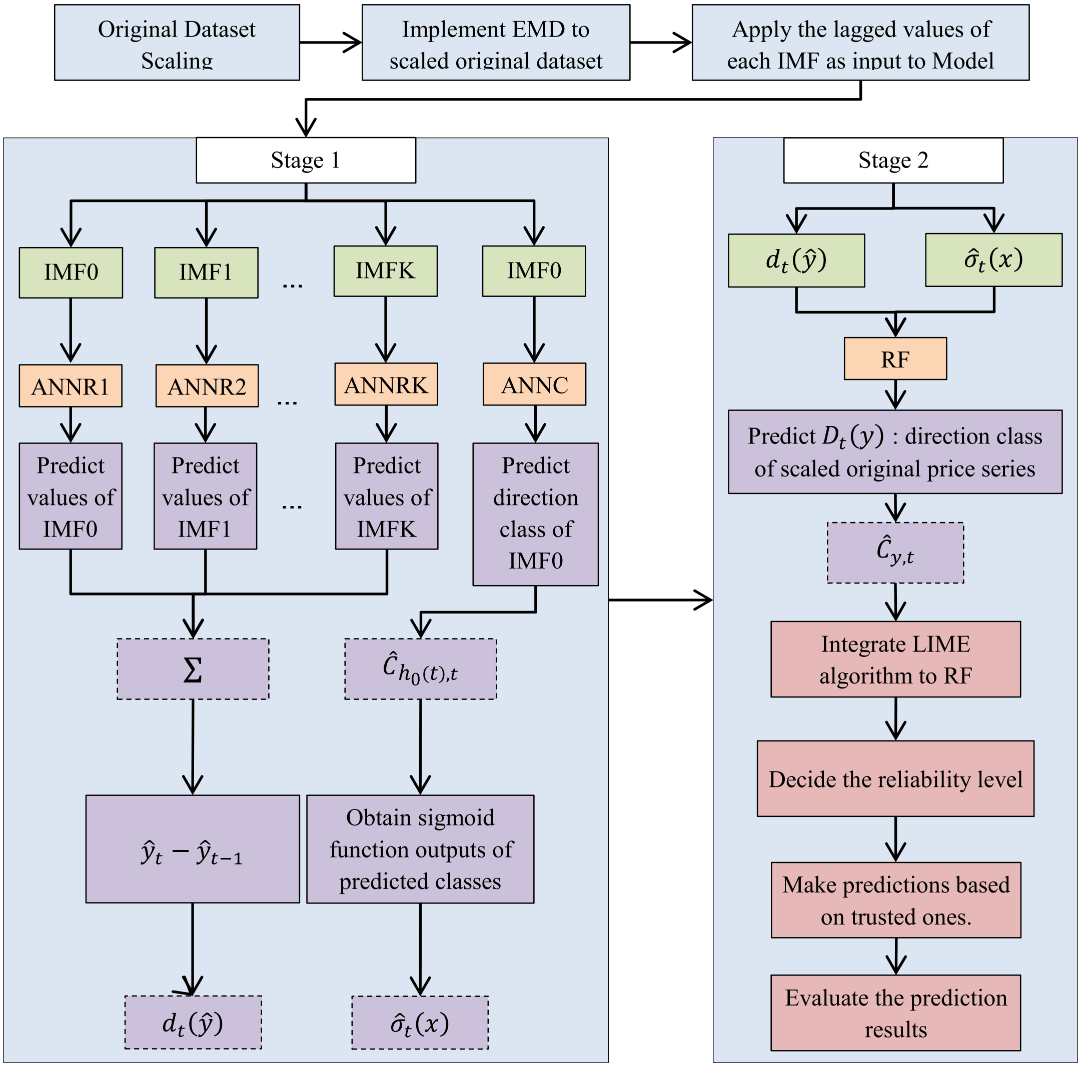
Original Dataset Scaling
Implement EMD to scaled original dataset
Apply the lagged values of each IMF as input to Model
Stage 1
IMF0
IMF1
...
IMFK
IMF0
ANNR1
ANNR2
...
ANNRK
ANNC
Predict values of IMF0
Predict values of IMF1
...
Predict values of IMFK
Predict direction class of IMF0
Σ
$\hat{C}_{h_0(t),t}$
$\hat{y}_t - \hat{y}_{t-1}$
Obtain sigmoid function outputs of predicted classes
$d_t(\hat{y})$
$\hat{\sigma}_t(x)$
Stage 2
$d_t(\hat{y})$
$\hat{\sigma}_t(x)$
RF
Predict $D_t(y)$: direction class of scaled original price series
$\hat{C}_{y,t}$
Integrate LIME algorithm to RF
Decide the reliability level
Make predictions based on trusted ones.
Evaluate the prediction results

Fig. 2. Schematic layout of proposed two stage EMD-ANNRC-RF-LIME model.

precision, recall, F1-score and ROC-AUC. The entire prediction procedure is depicted in Fig. 2.

In a nutshell, in the final version of the prediction model, the entire dataset has been decomposed by EMD, then by using ANNR and ANNC, the first prediction procedure of the ensemble model has been completed. Finally random forest has been employed for the second prediction procedure of the ensemble model and there by the final model has been called as EMD-ANNRC-RF. As the last step of Stage-2, LIME algorithm has been integrated to EMD-ANNRC-RF and the details of integrating LIME to the base model has been explained in the next sub-section.

Step by step prediction procedure of EMD-ANNRC-RF-LIME is given below:

1. Scale the original dataset.
2. Decompose the entire original dataset and obtain IMF series.
3. Divide each of the IMF dataset into training, validation and test sets.
4. Train ANN with n days lagged values of the IMF series as inputs and most recent value as output for each IMF. The ANN model utilized here is referred as ANNR for ANN regression.
5. Train another ANN with n days lagged values of IMF0 dataset, $IMF0_{t-1}, \ldots, IMF0_{t-n}$, as inputs and most recent day's movement direction of IMF0, $D_t\,(IMF0)$, as output. The ANN employed at this stage is a classifier and referred as ANNC.
6. Predict original dataset in validation set via ANNR by summation of all predicted IMFs and by calculating difference between original dataset predictions for each successive time periods t and $t-1$ obtain $d_t\left(\hat{y}\right)$ for each day.
7. Predict directional movements of IMF0 in validation set via ANNC. Obtain predicted *sigmoid function outputs*, $\hat{\sigma}_t\,(x)$, for each time period t of each classification.
8. Train a classification model with RF in validation set using $d_t\left(\hat{y}\right)$ and $\hat{\sigma}_t\,(x)$ which are obtained in steps 5 and 6 as inputs and direction of scaled original series $D_t\,(y)$ as output.
9. Run LIME algorithm on RF model object in validation set with the same input/output setup as described in previous step and obtain LIME algorithm. Determine reliability level and find out which predictions of RF satisfy this criterion enough with respect to given criteria, then predictions are made if the trusted prediction of RF is reliable enough with respect to the criteria.
10. Finally, in test set, predict original dataset with ANNR and predict IMF0 with ANNC1. Then feed prediction outputs of ANNR and ANNC1 to ANNC2 as inputs to make final

predictions with respect to pre-determined reliable class probabilities calculated by LIME algorithm.

2.4.1. Integrating LIME algorithm to the EMD-ANNRC-RF

Prediction with higher accuracy is vital for stock market prediction. Therefore, using a prediction model with a high accuracy rate will allow more successful results. However, in cases where the accuracy of the prediction model cannot be increased further, another approach can be suggested to increase the success of the predictions. Testing the reliability of the predictions by a model appears to be as an option. Any approach that can be developed as to whether a prediction is trustworthy is an effort to measure the extent to which the prediction is trusted, or with what probability it may come true.

In this case, if there was a measure of how trustworthy each prediction is, the predictions with relatively higher probability of occurrence could be used to make decisions so that they can be compared against the reliability level determined by decision maker. On the other hand, decision maker would be hesitant to make decisions for unreliable predictions (i.e. lower probability of occurrence). Thus, it would not be wrong to expect an ultimate increase in the success of the predictions, since only the predictions with a high level of reliability will be used in decision-making. In the ML literature, this approach is generally referred to as model explainability. Model explainability increases trust in a machine learning model because it allows it to be interpreted. There are two different ways to interpret a model: global and local. Global interpretation explains the whole model while local interpretation explains only predictions [37]. Global interpretation explain the complete behavior of the model while local interpretation helps to understand how the model makes decisions for a single instance and explain the individual predictions. LIME and SHAP are common algorithms for local interpretation. In this study, the LIME algorithm is used for this purpose. Since it is desired to explain each prediction made by the model and consult to the measurements related to while making decision making, local interpretability approach is employed for model explainability in this study.

Experiment design in this study has evolved around two major points. One of them is to increase the prediction success of the base model, as can be seen in the results in Section 3.2 while the other is to add another aspect to the prediction procedure using the LIME algorithm. Therefore, according to the second aspect of this study, a novel prediction procedure is proposed using the LIME algorithm. Essentially, the importance of using $\hat{\sigma}_t\left(x\right)$ as the predictive output of the ANNC model instead of the class labels directly to predict $D_t\left(y_t\right)$, and similarly, using $d_t\left(\hat{y}\right)$ instead of directly using $\hat{y}_t$ as the output of ANNR emerges at this stage. As mentioned in the previous section, albeit partially, a parallelism has been observed between the values of $\hat{\sigma}_t\left(x\right)$ and $D_t\left(y_t\right)$ in the preliminary experiments. As the values of $\hat{\sigma}_t\left(x\right)$ approach 1 or 0, the accuracy of the predictions for 1 or 0 values of $D_t\left(y_t\right)$ increases, respectively. It is observed that the increase in hit rates is similar for $d_t\left(\hat{y}\right)$. As the absolute amount of increase/decrease in the values of $d_t\left(\hat{y}\right)$ increases, the accuracy rates of the predictions made for the 1 or 0 values of $D_t\left(y_t\right)$ increase, respectively. The LIME algorithm has been utilized for benefiting from this observed phenomenon in a more systematic way. Thus, the reliability of the results of the prediction model can be measured for each value of $\hat{\sigma}_t\left(x\right)$ and $d_t\left(\hat{y}\right)$, and only the predictions that meet the reliability condition are used to predict according to a certain predetermined reliability level.

By integrating the LIME algorithm to the prediction model, each prediction of the RF model has been explained in the test set. The LIME technique enables these explanations based on the behavior of RF model in the validation set. As a result, for each prediction of RF model in the test set, prediction probabilities or explanation prediction probabilities (EPP) have been calculated for each class label. When any input instance is given to RF, since the prediction model is a binary classifier, it produces one of the values of 0 and 1 as a prediction output. After the LIME algorithm has been integrated to the base model, it has allowed a probability calculation of the probability of occurrence for each class label. In Fig. 3, seven different random prediction samples are given in Fig. 3. According to random sample 1, for class 0 (downward prediction) LIME calculates a prediction probability of 0.97 while for class 1 (upward prediction) a value of 0.03 is calculated. Corresponding input values can be observed in the right hand side of the figure. According to this, a downward movement will occur with 0.97 probability or an upward movement will occur with 0.03 probability. For random sample 2, prediction probabilities of 0 and 1 classes are 0.39 and 0.61 respectively. In this case, if the decision maker determines 0.70 as the reliability level beforehand, then he/she would trust the prediction in random sample 1 since the downward prediction probability (0.97) is greater than or equal to the reliability level. On the other hand, in random sample 2, since the prediction probability of both of the classes is less than 0.70, the decision maker would hesitate to make a prediction and refrain from making a decision.

3. Experimental results and evaluation

3.1. Data descriptions and experimental environment

All experiments in this study have been performed in Python 3.9 on Windows 10 Pro with AMD Ryzen 5900X processor and 32 GB RAM. Python libraries Pandas [38], NumPy [39], PyEMD [40], Scikit-learn [41], Keras [42], Matplotlib [43], xgboost [44], lime [45] are have been utilized. Ten years of historical data for each dataset have been used for the experiments. In order to visualize the test periods of each of the stock market index, daily closing prices have been depicted in Fig. 4. Also before conducting experiments, each data set has been scaled by maximum–minimum method, such that

$$s\left(y_t\right) = \frac{y_t - min\left(y_t\right)}{max\left(y_t\right) - min\left(y_t\right)} \tag{14}$$

where $s\left(y_t\right)$ and $min\left(y_t\right)$ stands for scaled dataset and minimum value of y_t respectively while $max\left(y_t\right)$ represents maximum value of y_t. Data normalization is a common practice for machine learning and deep learning tasks since prediction models can work more efficiently with scaled data.

Experiments have been carried out on six different stock market indices which are SP500, NI225, XU100, KOSPI, DAX and FTSE100. In order to train and predict stock market indices' upward and downward movement direction on the next day, the entire data set is divided into 50%, 25% and 25% subsets of training, validation and test sets respectively. Since four days lagged values of each IMF have been used as inputs for prediction in Stage-1, four days of data are missing. In addition, since the difference series, $d_t\left(\hat{y}\right)$, have been obtained from the predictions of ANNR, there occurs one more day loss. As a result of these, five days of data loss occur in total. Date ranges concerning the whole data sets its subset (training set, validation set and test set) sizes are given in Table 2.

Since the market calendar differs among countries, negligible differences occur between the total lengths of the data sets and the start-end dates. All of the experimental data sets have been downloaded from the tradingview website (https://tr.tradingview.com/).

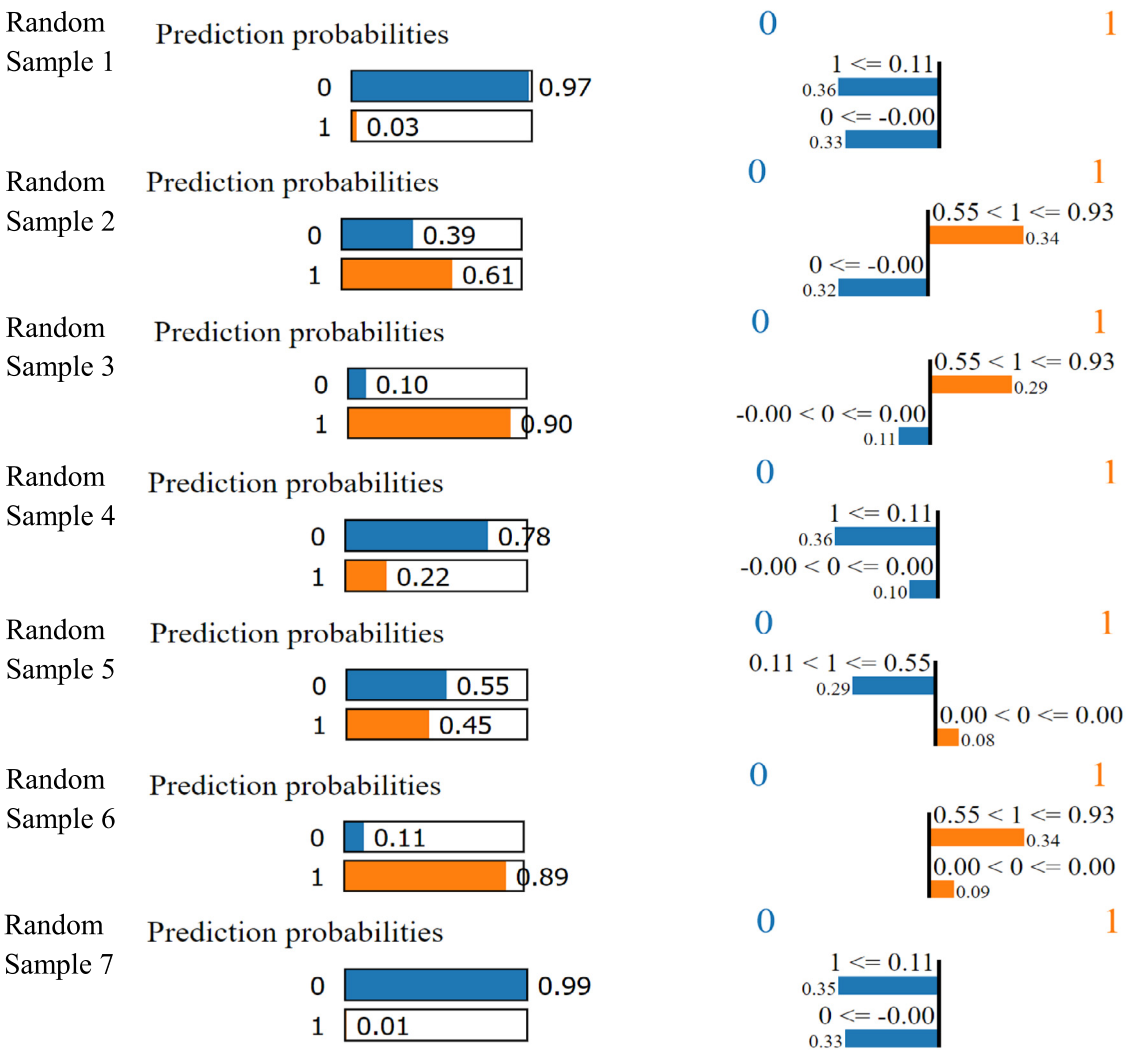

Fig. 3. Random samples from test set predictions and related instance explanations of LIME.

Table 2
Date ranges and sizes of total data, training set, validation set and test set of experiment data.

Index	Dates	Total data	Training set	Validation set	Test set
SP500	2012-01-03 ∼2021-12-31	2517	1261	625	626
NI225	2012-01-04 ∼2021-12-30	2445	1224	608	608
XU100	2012-01-02 ∼2021-12-31	2512	1258	624	625
KOSPI	2012-01-02 ∼2021-12-30	2460	1232	611	612
DAX	2012-01-02 ∼2021-12-30	2530	1267	629	629
FTSE100	2012-01-04 ∼2021-12-31	2529	1267	628	629

3.2. Results and final evaluation

All accuracy results of EMD-ANNR, EMD-ANNRC-XGBoost and EMD-ANNRC-RF models have been presented in Table 3 for each of the stock market indices. Obtained results reveal that proposed two stage ensemble prediction model is significantly better than EMD-ANNR according to performance evaluation metrics. Furthermore, EMD-ANNRC-RF is slightly better than EMD-ANNRC-XGBoost in all cases. It is possible to think that this might be due to the fact that XGBoost algorithm has more hyper parameters (the default hyper-parameters of the library [44] have been used) than RF hence it has to be fine-tuned

Comparisons with state of the art results are summarized in Table 4. In order to compare our base prediction model (EMD-ANNRC-RF), in Table 4, recent stock market prediction studies which proposes various techniques and approaches are listed. The prediction accuracies of these studies are ranging approximately between 0.60 and 0.90. Therefore, it is possible to claim that our base prediction model produces approximately close results with state of the art approaches.

As a final comment, the findings obtained by applying the LIME algorithm to the EMD-ANNRC-RF model are discussed below and the resulting model entitled as two-stage EMD-ANNRC-RF-LIME. Integrating the LIME algorithm to the prediction model makes it possible for the decision maker to rely on predictions which only satisfy the reliability condition. Therefore, at this stage, the value of the reliability level should be determined. Then, according to the determined reliability level, predictions are made during the test period and it should be calculated for which days the predictions will be made and to what extent these

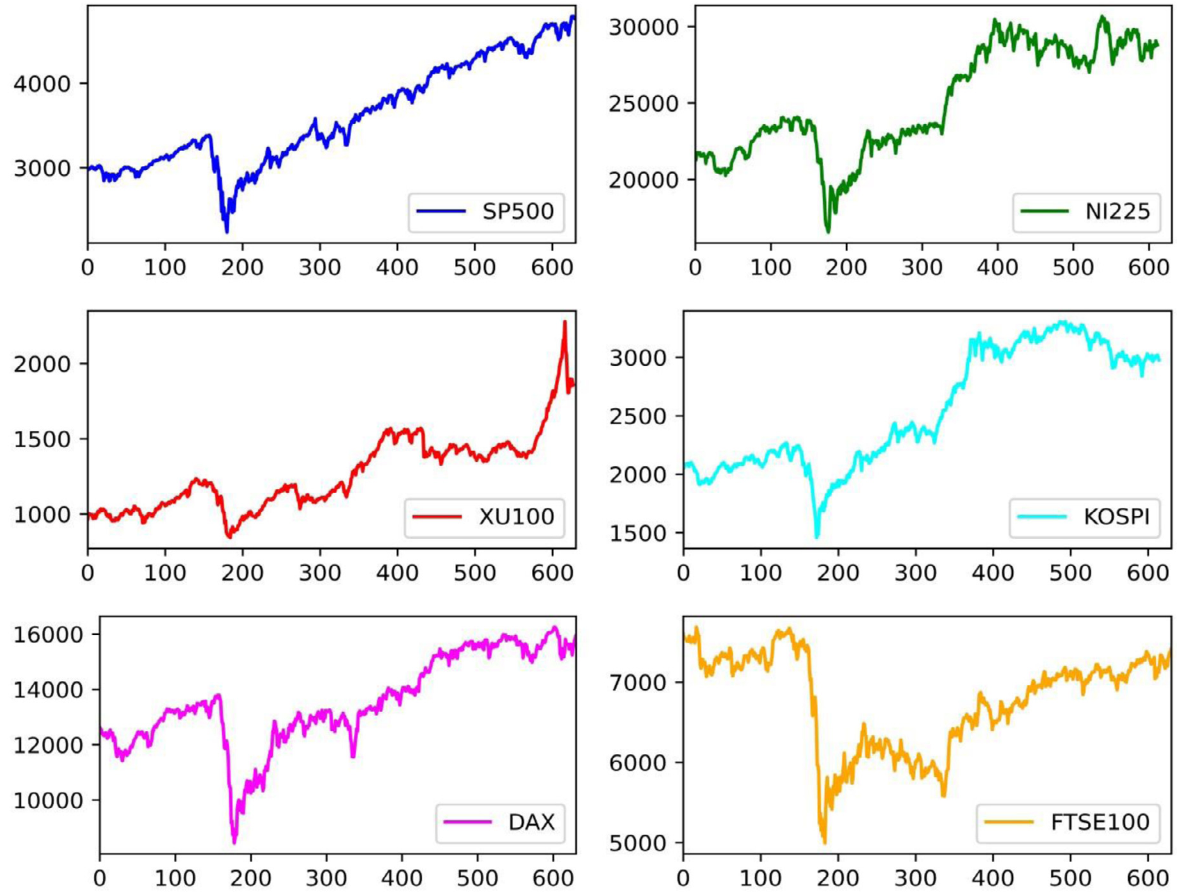

Fig. 4. Test periods of predicted indices.

Table 3
Comparison of prediction models.

		SP500	NI225	XU100	KOSPI	DAX	FTSE100
EMD-ANNR	Accuracy	0.6784	0.6903	0.7352	0.7328	0.6529	0.6476
	Precision	0.7112	0.6979	0.7519	0.7715	0.6715	0.6636
	Recall	0.7409	0.7241	0.8122	0.7514	0.6875	0.6909
	F1-score	0.7258	0.7108	0.7809	0.7613	0.6794	0.6770
	ROC-AUC	0.6675	0.6885	0.7203	0.7299	0.6503	0.6443
EMD-ANNRC-XGBoost	Accuracy	0.7796	0.7878	0.7872	0.7810	0.7727	0.7727
	Precision	0.8101	0.8065	0.8142	0.8257	0.7898	0.7699
	Recall	0.8056	0.7837	0.8209	0.7781	0.7827	0.8138
	F1-score	0.8078	0.7949	0.8176	0.8012	0.7862	0.7912
	ROC-AUC	0.7750	0.7880	0.7807	0.7815	0.7719	0.7701
EMD-ANNRC-RF	Accuracy	**0.7987**	**0.8010**	**0.7888**	**0.7925**	**0.7806**	**0.7838**
	Precision	0.8287	0.8133	0.8113	0.8313	0.7929	0.8050
	Recall	0.8194	0.8056	0.8292	0.7954	0.7976	0.7808
	F1-score	0.8240	0.8094	0.8202	0.8130	0.7953	0.7927
	ROC-AUC	0.7951	0.8007	0.7810	0.7920	0.7794	0.7840

predictions are accurate. Also, as the reliability level increases, it is necessary to test whether there is a corresponding increase in the accuracy rate. In Fig. 5, reliability level (horizontal axes) and accuracy rate (vertical axis) relation is plotted for the test sets of six different market indices. Reliability level varies between 0.5 to 1 as 0.5 reliability simply means working with EMD-ANNRC-RF model, in other words not using LIME technique at all. By increasing reliability level gradually towards 1, decision maker simply imposes the desire of getting more trustworthy predictions from the model. As previously mentioned, because of the tradeoff between the number of trusted predictions and reliability, one simply cannot expect to obtain trustworthy predictions for the whole prediction horizon. However, it is natural to think that increasing reliability level might also increase accuracy rate but having less numbers predictions. For all of the datasets, increasing reliability level also increases the hit rates in general with small deviations.

The number of days for which trusted predictions can be made and the corresponding accuracy rates are shown in Fig. 6. It should be noted that, normatively, only the predicted days are included in the calculation of hit rates. Therefore, if any of the EPP value calculated for each class label is not above the determined reliability level, the prediction will not be made, and one of the TP, TN, FP and FN values will not occur for that day.

The number of days for which a trusted prediction can be made and the corresponding hit rates are shown in Fig. 6. It should be noted that, naturally only the trusted predictions are included in the calculation of accuracy rates. Therefore, if any

Table 4
Prediction performance summary of some of the state of the art studies.

Author(s)	Methods	Dataset	Accuracy (%)
Bisoi et al. 2019 [22]	VMD and evolutionary robust kernel extreme learning machine (RKELM)	FTSE	85.83
Börjesson and Singull, 2020 [46]	Causal and dilated convolutional neural networks	S&P 500	74.96
Chung and Shin, 2020 [47]	Genetic algorithms and multi-channel convolutional neural network	KOSPI Indices	73.74
Ghorbani and Chong, 2020 [11]	Covariance information based on principal component analysis	50 stocks from USA stock markets	80
Long et al. 2020 [48]	Deep neural network and knowledge graph	CITIC Securities	73.59
Niu et al. 2020 [21]	VMD and LSTM	S&P 500	85.83
Thakkar and Chaudhari, 2020 [6]	Term frequency–inverse document frequency and neural networks	S&P 500	75
Yu and Yan, 2020 [49]	Phase-space reconstruction and LSTM	DJIA Indices	61.51
Gunduz, 2021 [10]	Variational auto-encoders and LSTM	BIST 30 Indices	68.5
Thakkar et al. (2021). [50]	Pearson Correlation Coefficient-based Neural Network	HDFC Bank stock price	78.48
Yin et al. (2021). [51]	Optimized random forest	Integrated Electronics Corporation stock price (INTC)	89
Yun et al. (2021). [3]	GA-XGBoost	KOSPI	93.82
Agrawal et al. (2022). [52]	LSTM	HDFC Bank stock price	65.64
Chandar (2022). [53]	Convolutional neural network	Bank of America corporation stock price	89.6
Lee et al. (2022) [15]	Attention-based BiLSTM	TPE0050	68.83
Yang et al. (2022). [12]	Group penalized logistic regressions	Amazon stock price	73.2

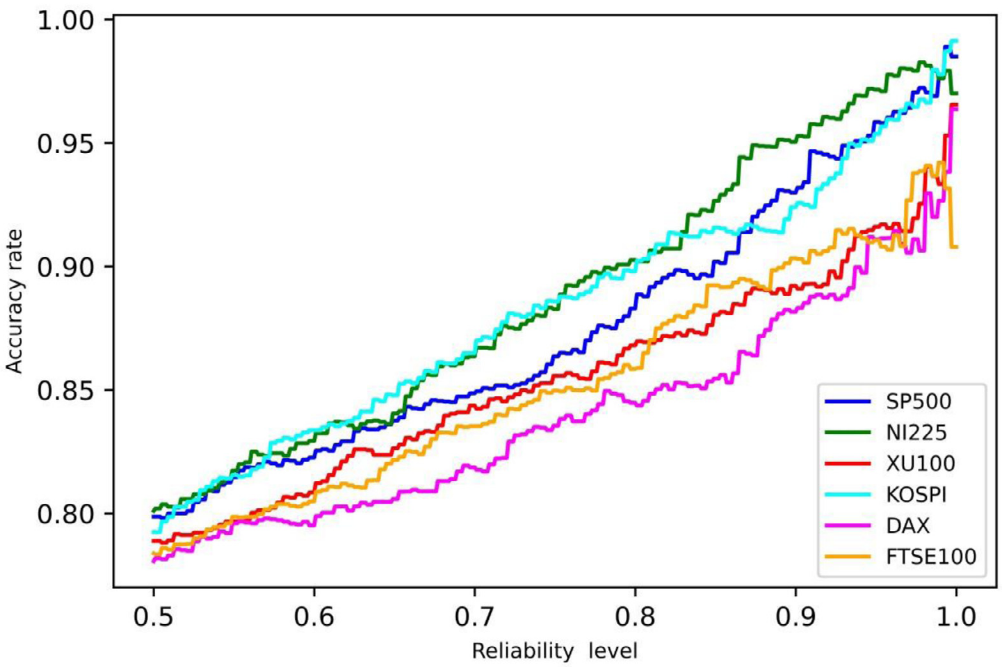

Fig. 5. Accuracy rates by reliability levels in test sets.

of the EPP value calculated for each class label is not above the determined reliability level, the model would abstain from making a prediction and none of the TP, TN, FP or FN values will occur for that day. The accuracy rates of the predictions made during the test period are between 0.7806 (DAX) for 629 trusted predictions and 0.8010 (NI225) for 608 trusted predictions at the bottom, while it ranges between 0.9079 (FTSE100) for 76 trusted predictions and 0.9913 (KOSPI) for 115 trusted predictions at the top. As the reliability level increases, the number of predictions meeting the reliability condition decreases as expected. Consequently, the accuracy rate increases as more trusted predictions are made as a result of the correlation between reliability level and accuracy rate implicitly.

For six stock market indices, the proportion of trusted predictions in test sets (*trustedpredictions*%) decreases for almost every reliability level while accuracy rate increases as can be seen in Fig. 7. On the other hand, as the reliability level exceeds approximately 0.90 level, the linear structure of the relationship starts to deteriorate and *trustedpredictions*% decreases drastically while an equivalent increase in the accuracy rate on average is not observed.

The experimental results of EMD-ANNRC-RF-LIME model are summarized in Table 5. For the cases where the reliability level is greater than or equal to each of the values %50, %85, %95 and %100, the hit rates and the corresponding number of trusted

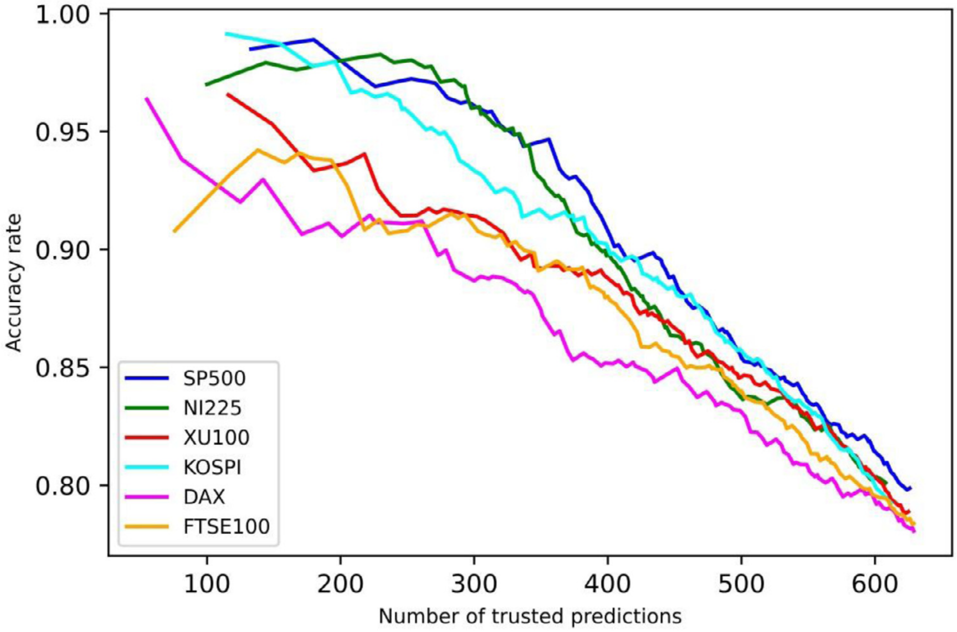

Fig. 6. Number of trusted predictions (horizontal axis) and accuracy rates (vertical axis) trade-off in test set.

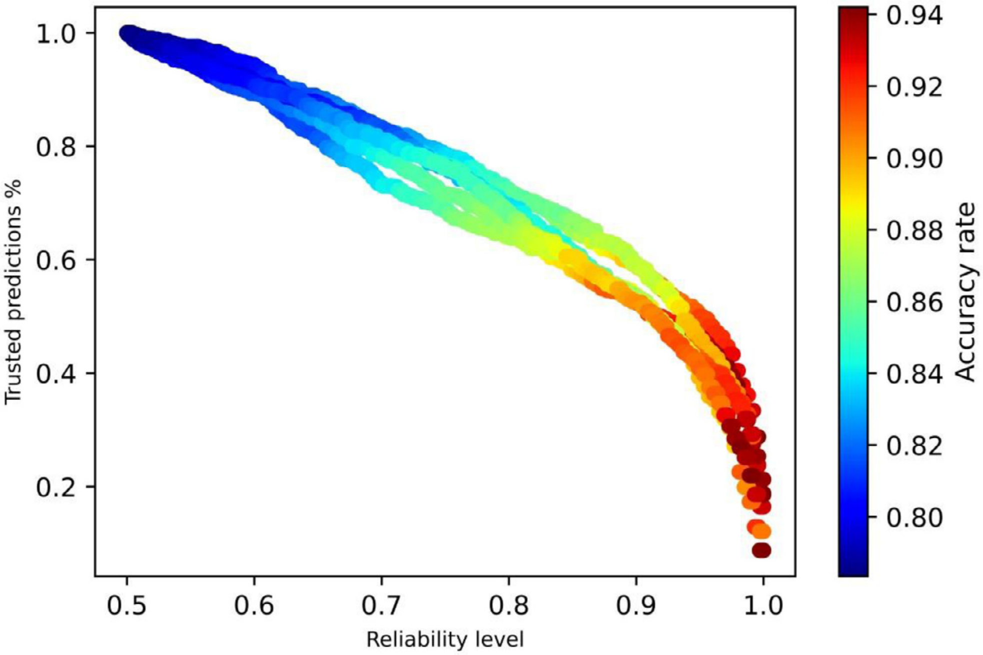

Fig. 7. Percentage of trusted predictions with respect to reliability level and accuracy rate.

predictions are given in Table 5. Notice that the number of trading days for Reliability $\geq$ 0.50 equals to whole test period length for all indices and this is due to the fact that it is equivalent act for not determining any level of reliability. In other words, it can be interpreted as predicting the entire test set by using EMD-ANNRC-RF model. The highest accuracy of the EMD-ANNRC-RF-LIME has been obtained on KOSPI data set with 0.9913 accuracy where 155 predictions are made for Reliability $=$ 1. On the other hand, lowest accuracy can be observed on FTSE100 index with 0.9079 for 76 predictions where Reliability $=$ 1.

On another note, although prediction accuracies of proposed EMD-ANNRC-RF model for six different datasets are close to each other, as the reliability level increases, the accuracy rates of six datasets begin to diverge (see Figs. 5 and 6). Therefore, it can be said that although the LIME algorithm fulfills its task as expected, EMD-ANNRC-RF-LIME model may need improvement in terms of robustness for higher levels of reliability.

4. Discussions

The fundamental approach that forms the basis of this study is to explain each prediction made by an ensemble classifier model with the LIME algorithm. In short, the predictions which are relatively more trustworthy are determined and it is ensured that only these predictions are considered in decision process. In this respect, in order to clearly demonstrate the contribution to

Table 5
Prediction result of EMD-ANNRC-RF-LIME.

Index	Reliability $= 1$		Reliability ≥ 0.95		Reliability ≥ 0.85		Reliability ≥ 0.50	
	Accuracy	Trusted predictions	Accuracy	Trusted predictions	Accuracy	Trusted predictions	Accuracy	Trusted predictions
SP500	0.9850	133	0.9585	313	0.9020	408	0.7987	626
NI225	0.9700	100	0.9715	281	0.9266	354	0.8010	608
XU100	0.9655	116	0.9161	286	0.8804	418	0.7888	625
KOSPI	0.9913	115	0.9537	259	0.9158	368	0.7925	612
DAX	0.9636	55	0.9109	247	0.8545	385	0.7806	629
FTSE100	0.9079	76	0.9105	257	0.8924	381	0.7838	629

the literature, a base model that can be a precedent for state of the art studies (which can make approximately similar successful predictions) has been created. Subsequently, the improvement made by integrating the LIME algorithm into the base model has been revealed. To be more precise, with the base model EMD-ANNRC-RF, an average hit rate of 0.7909 has been achieved in six different capital market indices, according to Table 3. Afterwards, the concept of prediction reliability level has been introduced with the EMD-ANNRC-RF-LIME model, and the prediction process has been continued with more reliable predictions by removing the predictions with relatively low reliability levels.

In this context, the decision maker is able to determine which predictions are more reliable (see sub Section 2.4.1. for more details). In Table 5, we have shared the results obtained for the 50%, 85%, 95%, and 100% reliability levels in order to summarize possible outcomes for corresponding reliability level. We have also shown the results obtained for all reliability levels in the 50%–100% range in Fig. 7. It is very clear that the point where the reliability level is determined as 50% is to trust the predictions made by the base prediction model (EMD-ANNRC-RF). On the other hand, as we have increased the reliability level, where LIME algorithm comes to the fore, predictions of EMD-ANNRC-RF model have been refined. For a better grasp of this concept, executing each prediction of the model with a reliability of 85% and above in the SP500 index eliminates those that do not meet this requirement. Thus, we can only make predictions for 408 days out of the total test period of 626 days. Therefore, some of the predictions have been avoided as sufficient reliability could not be satisfied for the 218 days of the test set. As a result, an accuracy rate of 0.9020 has been obtained for 408 predictions in the test set for the SP500 index. When we have determined the highest reliability level of 100% in six different stock market indices for the proposed model, the average predicted number of days decreased from ∼622 to ∼100 (approximately 84% decrease), while the average accuracy increased by 21.87%, resulting in 0.9639. Therefore, while the strength of this approach allows more successful predictions to be made, its weakness arises by not allowing prediction for the entire prediction process.

Although as a decomposition approach, EMD is able to provide successful results in our experiments, there are more up-to-date data decomposition methods such as ensemble empirical mode decomposition (EEMD) and VMD in the contemporary literature. Also parallel experiments carried out by employing VMD and EEMD techniques. According to our findings there was not any significant difference observed between EMD and mentioned techniques in terms of hit rates. Moreover, EMD is easier to implement than VMD since VMD requires parameter fine-tuning contrary to EMD. However, Niu et al. [21] has shown that VMD is more successful compared to EMD in stock market direction prediction. In this case, we assert that VMD does not make a difference in proposed ensemble model's context in this study. It can be taught that the reason behind this indifference might be due to the specific configuration of the proposed ensemble model. Therefore, we think that there is still room for improving obtained results by employing VMD with different settings in future studies.

ANN is sufficiently well in predicting time series but LSTM has been shown to be more successful [54]. Based on our preliminary experiment results, no significant difference has been observed between ANN and LSTM in terms of accuracy. In future studies we suggest that it might be useful to better explore LSTM algorithm with fine-tuned hyper parameters. The same inference can be applied to XGBoost model. Studies such as [3] have shown that XGBoost model can produce more successful results than RF. The reason for adoption of RF in our ensemble model's second stage is the fact that RF was observed to be slightly better than XGBoost in terms of accuracy rates. Moreover, RF has lesser hyper parameters relative to XGBoost hence it is more straightforward to use. On another note, the possible reason for XGBoost's inferior performance can be explained with the default parameter settings as there are much more hyper parameters which is need to be fine-tuned with respect to RF.

There is another important aspect to consider regarding data decomposition. In the literature including this paper, the entire dataset (training, validation and test) is to be decomposed beforehand. Therefore, decomposing the validation and test set along with training set is necessary before employing any machine learning technique. However, in practice, only the observed data can be decomposed and the next day values of the subcomponents are subject to prediction. To be more precise, if we denote the observed values of whole time series as $(t-k, t-k+1, \ldots, t)$, the values of the subcomponents at time $t+1$ can be predicted after the series of $(t-k, t-k+1, \ldots, t)$ have been decomposed. Afterwards, since $(t-k+1, t-k+2\ldots, t+1)$ series becomes the new observed original time series, it will be decomposed and the values of decomposed subcomponents at time $t+2$ will be predicted and the prediction process will continue in this way. This prediction procedure is simply referred to as simply sliding window. In short, these operations are performed one after the other for each window, and predictions are performed. It is suggested that financial time series prediction models with data decomposition should be developed with such an experimental design, especially to guide practitioners.

5. Conclusion and future directions

In the literature, LIME algorithm has not been used as suggested here in an integrated manner with machine learning techniques to make direct daily direction prediction as best of our knowledge. Therefore, integrating LIME in such a context is, in our opinion, an original contribution to the literature. Thanks to LIME algorithm's ability to avoid "unreliable" predictions of the model, and allowing relatively fewer but more reliable predictions, our proposed EMD-ANNRC-RF-LIME framework has proved to be distinctively successful.

The inevitable outcome of such a framework would be its use as a beneficial decision making tool for investors in capital markets. In a daily trading routine of a market participant usually more than one price series is to be predicted. As a result of this, when more than one price series are predicted in the same period, some predictions would be avoided for some assets while the others would be fulfilled. Therefore, instead of predicting a single asset's price, it is possible to make predictions for more than one price series at the same time and rely on assets which provide the reliability condition. Thus, in each prediction period, directional predictions are made for a certain number of assets that meet the reliability condition, while predictions that do not meet the reliability condition will be avoided and the predicted stocks will change periodically.

As a natural consequence of correlation between individual stock prices, most of the stocks move in the same direction in correspondence with the market conditions. As a final note, our findings in this paper offer a new perspective for enabling portfolio diversification while completely exploiting speculative potential of inversely correlated stocks. Therefore, for further studies, it is recommended to experiment on a large set of various assets simultaneously.

CRediT authorship contribution statement

Taha Buğra Çelik: Methodology, Conceptualization, Investigation, Software, Visualization, Data curation, Writing – original draft. **Özgür İcan:** Conceptualization, Software, Data curation, Writing – original draft. **Elif Bulut:** Supervision, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Z. Zhou, M. Gao, Q. Liu, H. Xiao, Forecasting stock price movements with multiple data sources: Evidence from stock market in China, Phys. Stat. Mech. Appl. 542 (2020) 123389, http://dx.doi.org/10.1016/j.physa.2019.123389.

[2] M.S. Ismail, M.S. Md Noorani, M. Ismail, F. Abdul Razak, M.A. Alias, Predicting next day direction of stock price movement using machine learning methods with persistent homology: Evidence from Kuala Lumpur Stock Exchange, Appl. Soft Comput. 93 (2020) 106422, http://dx.doi.org/10.1016/j.asoc.2020.106422.

[3] K.K. Yun, S.W. Yoon, D. Won, Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process, Expert Syst. Appl. 186 (2021) 115716, http://dx.doi.org/10.1016/j.eswa.2021.115716.

[4] J. Shen, M.O. Shafiq, Short-term stock market price trend prediction using a comprehensive deep learning system, J. Big Data 7 (1) (2020) 66, http://dx.doi.org/10.1186/s40537-020-00333-6.

[5] X. Zhang, N. Gu, J. Chang, H. Ye, Predicting stock price movement using a DBN-RNN, Appl. Artif. Intell. 35 (12) (2021) 876–892, http://dx.doi.org/10.1080/08839514.2021.1942520.

[6] A. Thakkar, K. Chaudhari, Predicting stock trend using an integrated term frequency–inverse document frequency-based feature weight matrix with neural networks, Appl. Soft Comput. 96 (2020) 106684, http://dx.doi.org/10.1016/j.asoc.2020.106684.

[7] Y. Hao, Q. Gao, Predicting the trend of stock market index using the hybrid neural network based on multiple time scale feature learning, Appl. Sci. 10 (11) (2020) 3961, http://dx.doi.org/10.3390/app10113961.

[8] S. Liu, X. Zhang, Y. Wang, G. Feng, Recurrent convolutional neural kernel model for stock price movement prediction, PLOS ONE 15 (6) (2020) e0234206, http://dx.doi.org/10.1371/journal.pone.0234206.

[9] C. Yang, J. Zhai, G. Tao, Deep learning for price movement prediction using convolutional neural network and long short-term memory, Math. Probl. Eng. 2020 (2020) 1–13, http://dx.doi.org/10.1155/2020/2746845.

[10] H. Gunduz, An efficient stock market prediction model using hybrid feature reduction method based on variational autoencoders and recursive feature elimination, Financ. Innov. 7 (1) (2021) 28, http://dx.doi.org/10.1186/s40854-021-00243-3.

[11] M. Ghorbani, E.K.P. Chong, Stock price prediction using principal components, PLOS ONE 15 (3) (2020) e0230124, http://dx.doi.org/10.1371/journal.pone.0230124.

[12] Y. Yang, X. Hu, H. Jiang, Group penalized logistic regressions predict up and down trends for stock prices, North Am. J. Econ. Finance 59 (2022) 101564, http://dx.doi.org/10.1016/j.najef.2021.101564.

[13] J. Patel, S. Shah, P. Thakkar, K. Kotecha, Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques, Expert Syst. Appl. 42 (1) (2015) 259–268, http://dx.doi.org/10.1016/j.eswa.2014.07.040.

[14] M. Nabipour, P. Nayyeri, H. Jabani, S.S., A. Mosavi, Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis, IEEE Access 8 (2020) 150199–150212, http://dx.doi.org/10.1109/ACCESS.2020.3015966.

[15] M.-C. Lee, J.-W. Chang, S.-C. Yeh, T.-L. Chia, J.-S. Liao, X.-M. Chen, Applying attention-based BiLSTM and technical indicators in the design and performance analysis of stock trading strategies, Neural Comput. Appl. 34 (16) (2022) 13267–13279, http://dx.doi.org/10.1007/s00521-021-06828-4.

[16] N.E. Huang, et al., The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, Proc. R. Soc. Lond. Ser. Math. Phys. Eng. Sci. 454 (1971) (1998) 903–995, http://dx.doi.org/10.1098/rspa.1998.0193.

[17] F. Xu, S. Tan, Deep learning with multiple scale attention and direction regularization for asset price prediction, Expert Syst. Appl. 186 (2021) 115796, http://dx.doi.org/10.1016/j.eswa.2021.115796.

[18] F. Zhou, H. Zhou, Z. Yang, L. Yang, EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction, Expert Syst. Appl. 115 (2019) 136–151, http://dx.doi.org/10.1016/j.eswa.2018.07.065.

[19] Z. Jin, Y. Yang, Y. Liu, Stock closing price prediction based on sentiment analysis and LSTM, Neural Comput. Appl. 32 (13) (2020) 9713–9729, http://dx.doi.org/10.1007/s00521-019-04504-2.

[20] K. Dragomiretskiy, D. Zosso, Variational mode decomposition, IEEE Trans. Signal Process. 62 (3) (2014) 531–544, http://dx.doi.org/10.1109/TSP.2013.2288675.

[21] H. Niu, K. Xu, W. Wang, A hybrid stock price index forecasting model based on variational mode decomposition and LSTM network, Appl. Intell. 50 (12) (2020) 4296–4309, http://dx.doi.org/10.1007/s10489-020-01814-0.

[22] R. Bisoi, P.K. Dash, A.K. Parida, Hybrid variational mode decomposition and evolutionary robust kernel extreme learning machine for stock price and movement prediction on daily basis, Appl. Soft Comput. 74 (2019) 652–678, http://dx.doi.org/10.1016/j.asoc.2018.11.008.

[23] Y. Yujun, Y. Yimei, Z. Wang, Research on a hybrid prediction model for stock price based on long short-term memory and variational mode decomposition, Soft Comput. 25 (21) (2021) 13513–13531, http://dx.doi.org/10.1007/s00500-021-06122-4.

[24] J. Xiao, X. Zhu, C. Huang, X. Yang, F. Wen, M. Zhong, A new approach for stock price analysis and prediction based on SSA and SVM, Int. J. Inf. Technol. Decis. Mak. 18 (01) (2019) 287–310, http://dx.doi.org/10.1142/S021962201841002X.

[25] H. Liu, Z. Long, An improved deep learning model for predicting stock market price time series, Digit. Signal Process. 102 (2020) 102741, http://dx.doi.org/10.1016/j.dsp.2020.102741.

[26] Y. Yujun, Y. Yimei, X. Jianhua, A hybrid prediction method for stock price using LSTM and ensemble EMD, Complexity 2020 (2020) 1–16, http://dx.doi.org/10.1155/2020/6431712.

[27] M. Mohandes, M. Deriche, S.O. Aliyu, Classifiers combination techniques: A comprehensive review, IEEE Access 6 (2018) 19626–19639, http://dx.doi.org/10.1109/ACCESS.2018.2813079.

[28] R. Dash, S. Samal, R. Dash, R. Rautray, An integrated TOPSIS crow search based classifier ensemble: In application to stock index price movement prediction, Appl. Soft Comput. 85 (2019) 105784, http://dx.doi.org/10.1016/j.asoc.2019.105784.

[29] D.K. Padhi, N. Padhy, A.K. Bhoi, J. Shafi, M.F. Ijaz, A fusion framework for forecasting financial market direction using enhanced ensemble models and technical indicators, Mathematics 9 (21) (2021) 2646, http://dx.doi.org/10.3390/math9212646.

[30] E.K. Ampomah, Z. Qin, G. Nyame, Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement, Information 11 (6) (2020) 332, http://dx.doi.org/10.3390/info11060332.

[31] M.T. Ribeiro, S. Singh, C. Guestrin, Why should I trust you?': Explaining the predictions of any classifier, 2016, arXiv, (Accessed: Oct. 07, 2022). [Online]. Available: http://arxiv.org/abs/1602.04938.

[32] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, 2017, p. 30, (Accessed: Oct. 08, 2022). [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

[33] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (4) (1943) 115–133, http://dx.doi.org/10.1007/BF02478259.

[34] D. Rothman, Hands-on Explainable AI (XAI) with Python: Interpret, Visualize, Explain, and Integrate Reliable AI for Fair, Secure, and Trustworthy AI Apps, Packt Publishing Ltd, 2020.

[35] H. Niu, K. Xu, University of Science and Technology Beijing, Beijing 100083, China School of Economics and Management, A hybrid model combining variational mode decomposition and an attention-GRU network for stock price index forecasting, Math. Biosci. Eng. 17 (6) (2020) 7151–7166, http://dx.doi.org/10.3934/mbe.2020367.

[36] C. Zhang, H. Pan, A novel hybrid model based on EMD-BPNN for forecasting US and UK stock indices, in: 2015 IEEE International Conference on Progress in Informatics and Computing, PIC, 2015, pp. 113–117, http://dx.doi.org/10.1109/PIC.2015.7489820.

[37] A. Heuillet, F. Couthouis, N. Díaz-Rodríguez, Explainability in deep reinforcement learning, Knowl.-Based Syst. 214 (2021) 106685, http://dx.doi.org/10.1016/j.knosys.2020.106685.

[38] J. Reback, et al., pandas-dev/pandas: Pandas 1.0.3. Zenodo, 2020, http://dx.doi.org/10.5281/ZENODO.3715232.

[39] C.R. Harris, et al., Array programming with NumPy, Nature 585 (7825) (2020) http://dx.doi.org/10.1038/s41586-020-2649-2, Art. (7825).

[40] PyEMD's documentation – PyEMD 0.2.13 documentation, 2022, https://pyemd.readthedocs.io/en/latest/index.html (accessed Dec. 02, 2022).

[41] F. Pedregosa, et al., Scikit-learn: Machine learning in Python, Mach. Learn. PYTHON, p. 6.

[42] Keras: Deep learning for humans, Keras (2022) (Accessed: Oct. 08, 2022). [Online]. Available: https://github.com/keras-team/keras.

[43] T.A. Caswell, et al., matplotlib/matplotlib: REL: v3.5.2. Zenodo, 2022, http://dx.doi.org/10.5281/ZENODO.6513224.

[44] XGBoost | Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2022, https://dl.acm.org/doi/abs/10.1145/2939672.2939785 (accessed Oct. 08, 2022).

[45] Local interpretable model-agnostic explanations (lime) — lime 0.1 documentation, 2022, https://lime-ml.readthedocs.io/en/latest/index.html, (Accessed Oct. 08, 2022).

[46] L. Börjesson, M. Singull, Forecasting financial time series through causal and dilated convolutional neural networks, Entropy 22 (10) (2020) 1094, http://dx.doi.org/10.3390/e22101094.

[47] H. Chung, K. Shin, Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction, Neural Comput. Appl. 32 (12) (2020) 7897–7914, http://dx.doi.org/10.1007/s00521-019-04236-3.

[48] J. Long, Z. Chen, W. He, T. Wu, J. Ren, An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market, Appl. Soft Comput. 91 (2020) 106205, http://dx.doi.org/10.1016/j.asoc.2020.106205.

[49] P. Yu, X. Yan, Stock price prediction based on deep neural networks, Neural Comput. Appl. 32 (6) (2020) 1609–1628, http://dx.doi.org/10.1007/s00521-019-04212-x.

[50] A. Thakkar, D. Patel, P. Shah, Pearson correlation coefficient-based performance enhancement of vanilla neural network for stock trend prediction, Neural Comput. Appl. 33 (24) (2021) 16985–17000, http://dx.doi.org/10.1007/s00521-021-06290-2.

[51] L. Yin, B. Li, P. Li, R. Zhang, Research on stock trend prediction method based on optimized random forest, CAAI Trans. Intell. Technol. (2021) http://dx.doi.org/10.1049/cit2.12067, cit2.12067.

[52] M. Agrawal, P. Kumar Shukla, R. Nair, A. Nayyar, M. Masud, Stock prediction based on technical indicators using deep learning model, Comput. Mater. Contin. 70 (1) (2022) 287–304, http://dx.doi.org/10.32604/cmc.2022.014637.

[53] S.K. Chandar, Convolutional neural network for stock trading using technical indicators, Autom. Softw. Eng. 29 (1) (2022) 16, http://dx.doi.org/10.1007/s10515-021-00303-z.

[54] D. Wu, X. Wang, J. Su, B. Tang, S. Wu, A labeling method for financial time series prediction based on trends, Entropy 22 (10) (2020) 1162, http://dx.doi.org/10.3390/e22101162.