Performance Analysis of Machine Learning Based Ensemble Methods: Stacking, Voting and Blending for Predicting Financial Market Trends

By

Fiza Noor

CIIT/FA21-RMT-019/ISB

MS Thesis

In

Mathematics

COMSATS University Islamabad, Islamabad - Pakistan

Spring, 2023

COMSATS University Islamabad

Performance Analysis of Machine Learning Based Ensemble Methods: Stacking, Voting and Blending for Predicting Financial Market Trends

A Thesis Presented to

COMSATS University Islamabad

In partial fulfillment
of the requirement for the degree of

MS (Mathematics)

By
Fiza Noor
CIIT/FA21-RMT-019/ISB

Spring, 2023

Performance Analysis of Machine Learning Based Ensemble Methods: Stacking, Voting and Blending for Predicting Financial Market Trends

A Post Graduate Thesis submitted to the Department of Mathematics as partial fulfilment of the requirement for the award of Degree of MS (Mathematics).

Name	Registration Number
Fiza Noor	CIIT/FA21-RMT-019/ISB

Supervisor:

Dr. Sajjad Khan,
Associate Professor, Department of Mathematics,
COMSATS University Islamabad,
Islamabad, Pakistan

Co-Supervisor:

Dr. Nadeem Javaid,
Professor, Department of Computer Science,
COMSATS University Islamabad,
Islamabad, Pakistan

Final Approval

This thesis titled

Performance Analysis of Machine Learning Based Ensemble Methods: Stacking, Voting and Blending for Predicting Financial Market Trends

By

Fiza Noor

CIIT/FA19-RMT-019

has been approved

For the COMSATS University Islamabad, Islamabad

External Examiner:______________________________

Dr. Muhammad Saleem

Department of Science and Humanities

Sir Syed CASE Institute of Technology, Islamabad Pakistan

Supervisor:______________________________

Dr. Sajjad Khan

Department of Mathematics

COMSATS University Islamabad, Islamabad Campus

Co-Supervisor:______________________________

Prof. Dr. Nadeem Javaid

Department of Computer Science

COMSATS University Islamabad, Islamabad Campus

HoD:______________________________

Prof. Dr. Shams-ul Islam

Department of Mathematics

COMSATS University Islamabad, Islamabad Campus

Declaration

I Fiza Noor (Registration No. CIIT/FA21-RMT-019/ISB) hereby declare that I have produced the work presented in this thesis, during the scheduled period of study. I also declare that I have not taken any material from any source except referred to wherever due that amount of plagiarism is within acceptable range. If a violation of HEC rules on research has occurred in this thesis, I shall be liable to punishable action under the plagiarism rules of the HEC.

Date: July, 2023

Fiza Noor
CIIT/FA21-RMT-019/ISB

Certificate

It is certified that Fiza Noor (Registration No. CIIT/FA21-RMT-019/ISB) has carried out all the work related to this thesis under my supervision at the Department of Mathematics, COMSATS University, Islamabad and the work fulfils the requirement for award of MS degree.

Date: July, 2023

Supervisor:

Dr. Sajjad khan
Associate Professor, Department of Mathematics

Co-Supervisor:

Dr. Nadeem javaid
Professor, Department of Computer Science,

Head of Department:

Dr. Shamsul Islam
Department of Mathematics

DEDICATION

$\mathcal{D}$edicated

to my mentor Shamsul Islam, loving Parents, who equipped me with pearls of knowledge and showed me the way of spiritual and personal enlightenment in this world and the world hereafter.

ABSTRACT

Performance Analysis of Machine Learning Based Ensemble Methods: Stacking, Voting and Blending for Predicting Financial Market Trends

The financial stock market is the backbone of every country's economy. Beforehand prediction of inflation periods, shocks, closing price of a stock, and forecasting of up and down trends are the most challenging tasks. The type of input data used to feed the model has a significant impact on how well the model performs. This study deals with highly volatile and non-linear Toronto stock exchange (TSE) signal data to predict the closing trend within a day. Stock signals are quite complex with abrupt frequencies. Although proposed systems of literature can struggle to capture non-linear dependencies of signal data, which they lack to perform over diversified input sets. We conduct a comparative analysis of three diverse heterogeneous ensemble methods. The purpose of this integrated study is to investigate the question that which ensemble method generates generalized results proven by maximum performance checkpoints. Input data is explored by missing values and class imbalance check and features importance is calculated using random forest (RF) in predicting trends. Randomly selected training and testing data is fed into all three proposed pipelines to get comparative outcomes. Moreover, each pipeline is lined up with 4 differently skillful base classifiers (QDA, XGB, EXT, NB). Then, finally, Adaboost is employed as a meta-classifier to ensemble the outcomes of base classifiers to improve classification accuracy. For the robustness check, we conducted an extensive performance analysis using 14 distinct checkpoints to ensure the market's liquidity. Empirical results reveal that stack ensemble yields highly generalized and unbiased predictions due to its well-designed training of input data and thus can be a preferred tool for stock market trend prediction (STP) on economic forums.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Abbreviations

Table 1: List of Abbreviations

Abbreviation	**Full Form**
ADABOOST	Adaptive Boosting
API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
AUCROC	Area Under ROC Curve
BM	Benchmark Model
BPNN	Back-Propagation Neural Network.
CCI	Commodity Channel index
CNN	Convolutional Neural Network
CSMAR	China Stock Market And Accounting Research
DC	Directional Change
DCO	Dynamic Creative Optimization
DA	Decision Accuracy
DL	Deep Learning
DLNN	Deep Learning Neural Network
DM	Data Mining
DT	Decision Tree
EEMD	Ensemble Empirical Mode Decomposition
EMH	Efficient Market Hypothesis
ENET	Elastic Net
EUR	Euro
FRPCA	Fuzzy Robust Principle Component Analysis
GARCH	Generalized Auto-regressive Conditional Heteroscedasticity
GBM	Gradient Boosting Machine
GP	Genetic Programming
KNN	K-nearest Neighbour
KPCA	Kernel Based Principle Component Analysis

LASSO	Least Absolute Shrinkage and Selection Operator
LM	Linear Model
LR	Logistic Regression
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLP	Multi-Layer Perceptron
MSR	Markov Switching Regression
MACDH	Moving Average Convergence/Divergence
NB	Naive Bayes
OHLC	Open High Low Close
OLS	Ordinary Least Square
PCA	Principle Component Analysis
POP	Point-Of-Sale
RCSNet	Residual-CNN-Seq1Seq
RF	Random Forest
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SampEn	Sample Entropy
SMOTE	Synthetic Minority Over Sampling Technique
SOE	Estate Owned Enterprises
STOCKNET	Stock Temporally-Dependent Movement Prediction Model
SVC	Support Vector Classifier
SVR	Support Vector Regression
TS	Trading Strategy
VAR	Multivariate Vector Autoregressive
WAPE	Weighted Average Percentage Error
WMGEPSVM	Weighted Multi-category Generalized Eigenvalue Support Vector Machine
XGB	Extreme Gradient Boosting
ZCR	Zero-Crossing Rate

Chapter 1

Introduction

1.1 Introduction

In this section of thesis, the background and motivation, introduction of Financial Stock Market, contributions and organization of thesis have been presented.

1.1.1 Financial Stock Market

The stock market, also named the equity market, is a marketplace that regulates the securities between buyers and sellers. There are a variety of stock markets based on the types of stocks involved. Markets comprise bonds, shares, property, goods, and services. The stock market is an independent entity that runs under formalized rules set by a self-regulatory organization. Its information is secured and cannot be influenced by the government or external authority. This adds to its unpredictable and imperceptible trends' behavior. For instance, forecasting stock trends, prediction of shock in stock signal, and estimation of the closing price of the stock are major concerns to date [1]. Decisions made by the public based on market position, expenses, and growth rates are not practical. The strategies and technologies are required to have the capability of analyzing direct and indirect factors affecting stock signals. For instance, the entrance and leaving time of investors from the market, companies' public image, corporate news, and political situation. The proposed models will perform best both for classification and regression on all data granularities. In the last decade, researchers have modified their mode of action against classical approaches, which were totally based upon public opinions to data driven approaches. [12].

1.1.2 Background and Motivation

The emergence of data science has drastically uplifted the research areas in almost every field of life e.g., weather forecasting, detection of conical diseases, image recognition, fraud detection, and many others. Along with that, the studies done in [20] prove that financial forecasting is a heated debate among investors, retailers, and traders. Many studies have been done and many are in progress to provide fruitful implications regarding inflation points, recession periods, and buy/hold/sell strategies. Among these strategies, machine learning, deep learning, and hybrid strategies are known to provide meaningful and influential benefits [31]. However, this road has many bumpy potholes on its way.
The chaotic nature of pricing data, non-linear dependencies among features, volatility, limited information, and unequal dominance between institutional and retail investors are all constraints towards accurate prediction [35],[36]. These can be overcome by cleaning input data from noise, removing redundant features, and balancing minority and majority classes. A huge difference in terms of accuracy is created by employing the data pre-processing step before passing it to the model. Normalization, applying technical indicators, filling of missing values, removal of outliers, dimensionality reduction, segregation of noise and informative data features are some advanced tools used to reduce overfitting, computational complexity, and biases of results [37].
In our case study, we emphasized on machine learning tools under different ensemble combinations to predict stock trend and conduct a comparative analysis in between.

1.1.3 Thesis Contributions

Our main contributions are discussed below:

- A novel combination of base and meta classifiers are used for comparative analysis of each ensemble method.
- A novel comparative machine learning study is carried out on Toronto stock exchange (TSE) for predicting its market trend.
- Our proposed stacking model has beaten its competitors with 3% higher accuracy along with higher values in other applied performance checkpoints.
- This study provides an elaborate rooms of study of pros and cons of diverse machine learning algorithms and ensemble methods.
- Moreover, it has highlighted the importance of accurate prediction of 'down trend' in saving the economic world.

1.1.4 Organization of Thesis

The remainder of the thesis is organized as follows: related studies are presented in Chapter 2. System model and mathematical formulations along with proposed methodology are demonstrated in Chapter 3. Chapter 4 describes the performance metrics employed for the robustness check of ensemble methods. Finally, the findings and simulation resuts are depicted in Chapter5, along with their comparative analysis. Finally, conclusion of this work along with future directions are presented in Chapter 6.

Chapter 2

Literature review and problem statement

2.1 Literature Review

The literature review is done on the basis of divisions of continents of the world. It covers the continental study of America, Asia, Europe, and Hybrid, given in different subsections. The hybrid includes the research, which is done on the stock data of a combination of the above three continents. Further, these subsections are subdivided into two streams: prediction and forecasting. Prediction represents those studies in which authors have conducted regression for estimating continuous values while forecasting represents those studies in which classification of up and down trends of stock signals is done. Overall, this section provides an extensive and in-depth study catering to multiple kinds of limitations with their proposed solutions and performance analysis with different classification and regression metrics. In the end, a concise summary is given in the form of a table with the following columns or attributes: country, dataset indices, limitations, proposed models, performance metrics, and future work. Following figure depicts the flow of related work.

2.1.1 Continent Based Categories

2.1.1.1 America

a) Forecasting

Stock signals constantly fluctuate between recession, contraction and expansion periods. Recession periods occur rarely throughout the stock signals data as compared to expansion periods. This is because recession periods exist for short time period as compared to other contraction and expansion periods. It represents decline of economic condition for few months. This scarcity causes class imbalance issue leading to inaccurate predictions and excessive computational power. In previous researches, this issue was ignored. However, the authors in [1] highlighted this gap. Moreover, traditionally used binary logit, probit function and gradient boosting failed to predict recessions. While in [2], the role of noise in the trend prediction of stock signals is critically analyzed. The term noise refers to discontinuous signals which cause biases in the prediction results. It is crucial to detect their location and remove them from the dataset before passing them to the model. Previous researches employed deep architectures, recurrent neural network (RNN) and long short term memory (LSTM), for prediction. However, they could not generate accurate results. The major reason is that noise signals are forwarded instead of original input signals, which deteriorate the models' performance.

Authors in [3], predict daily stock market returns. However, satisfactory results have not been provided due to incomplete and inefficient data mining. In high dimensional datasets, correlated variables retard the model's performance in predicting new behavior of stock market movements. This paper handles the following constraints, psychological factors, overfitting and curse of dimensionality by removing complications in finding the least eigenvalue from eigenvalue problem. However, [4] addressed these ignored facts that stock price time series data is comprised of not only linear components but, also non-linear dependent components. Along with that non-linear component of data contains both long and short patterns of signal's data. Therefore, there is a need to formulate an architecture that could capture these temporal and non-linear components of signal data.

The authors in [5] emphasized on the following two prediction trend affecting factors: external and internal. External factors include economic and political situation of the country, image of a country portrayed on international media and natural disasters. Along with internal factors like government policies, market portfolio and opinions of investors based on the history, all of them are interlinked with the performance of a model. Unstructured, volatile and complex data needs to be filtered from input data in [6]. These are created due to interconnected global economic and political conditions, and their sudden impact on financial stock markets. Moreover, mass media impact is not explored in the generalization of erratic signals. All of these constraints are required to be taken into account by the proposed model. The study in[7] highlighted that considering only opening and closing stock index for accurate prediction is not enough. Rather, sign of their difference is crucial in generating appropriate results. Previous studies ignored this fact.

Forecasting of United States (US) recession using high dimensional financial and macroeconomic datasets is done in [1] by using weighted gradient boosting method (wGBM).

Stock signal data have scarce recession periods which create class imbalance issue [2]. To resolve it, the author used a novel approach empirical mode decomposition (EMD) and sample entropy (sampEn), for disintegrating stock signals into a series of small components, known as intrinsic mode function (IMF). It makes calculations easier. The model uses S&P500 stocks data from the year January 2018 to April 2020. SampEn is used for accessing the complexity of psychological time series signals. These proposed techniques separate informative signals from noise. High value of SampEn decreases prediction of accurate results. Active noise reduction (ANR), is used for removing noise by generating such sound, which eradicates its effect. The authors in [3] proposes artificial neural network (ANN) to forecast the trend of stock market return. Although, ANN is a powerful tool for handling volatile stock signals training data, it is oblivious to overfit due to high dimensions of testing input data. Therefore, it is necessary to rearrange the data by extracting the most relevant and informative features. For this purpose, three dimensionality reduction techniques including principal component analysis (PCA), fuzzy robust principal component analysis (FRPCA), and kernel-based principal component analysis (KPCA) are employed. Each of above technique transforms the original data into twelve new sets.Collectively, these thirty six segregated and cleaned sets are given input to ANN to produce generalized results.

In order to capture non-linear components of stock signals, a new combination of deep models residual-cnn-seq2seq (RCSNet), is formulated in[4]. It is composed of autoregressive integrated moving average (ARIMA) model, which filters linear components from input data. It is followed by convolutional neural network (CNN) to segregate different length patterns of signals. Then, sequence-to-sequence (Seq2Seq) LSTM layer is imported to generate non-linear forecast results. Finally, both linear and non-linear forecasts are connected through a fully connected layer to give the final output. This task utilizes American stock dataset, S&P500 dataset, spanning 16 year's time period. Further, it is randomly divided into training and testing, which contains 70% and 30% data, respectively. Study [5] tackles the effect of internal and external factors affecting predictions by employing a hybrid model, termed as STOCKNET, and optimizing the hyperparameters by trial and error approach. Technical indicators are selected from original data using support vector machine (SVM), random forest (RF) and L1-regularized logistic regression (L1-LR). Further, results are compared with baseline models. Two evaluation metrics, accuracy (AUC) and mathew's correlation coefficient (MCC), are used for performance analyses in this research.

The author in [6] proposes a hybrid deep model composed of Word2vec and LSTM. Word2vec learns the patterns of words having the capability to detect synonymous words and complete the partial sentences. While LSTM has the capability to memorize the previous locations of words of input data and decide which information to store and delete. Their combination makes robust predictive analysis using news headlines and closing prices of stocks. It is tested on datasets of Apple, PepsiCo, NRG, American Public Education (APEI) and American Telephone and Telegraph (AT&T) generating binary outcomes. In study [7], data pre-processing along with machine learning techniques are considered in order to fill literature gaps. It includes first order normalization and zero-crossing rate (ZCR) to handle volatility of data and to remain unaffected from unbalanced data effects. LSTM model is employed to predict opening, closing price and difference between opening and closing price (DCO). Price data is taken from Dow Jones and the S&P500 indices from the Yahoo of 5540 days from 2/1/2005 to 12/30/2016. Evaluation metrics RMSE, MAE, prediction of the sign (POS), true positive rate (TPR) and true negative rate (TNR) are utilized to check the robustness of proposed model.

The authors in [1] acknowledge that use of different kinds of interest rates pave the path for accurate prediction of US recession periods. Using cost-sensitive GBM model, gives predictions of future recession periods for short and medium time period. It is validated by area under curve (AUC), area under receiver operating characteristic curve (AUROC) and receiver operating characteristic curve (ROC). Moreover, results show that proposed wGBM give generalized results for both in sample and out sample data with different time horizons. It outperforms previous benchmark models (binary logit, probit function, gradient boosting model) significantly. Results in [2] show that value of mean and weight absolute percentage error (MAPE and WAPE) are lowered by 18% with increase of 31.52% in direction accuracy (DA). This difference is obtained through the comparison of model's accuracy using original signal and transformed signal. Further, LSTM shows improved results with 31% increase in stocks trend direction prediction. Moreover, this study is applicable in following time series domains: weather forecasting, traffic signals, video and speech analysis.

Results in [3], show that after applying dimensionality reduction techniques, ANN generated similar values in both training and validation sets. That is, there is no over fitting. Moreover, PCA outperforms than other two variants kernel principal component analysis (KPCA) and fuzzy robust principal component analysis (FRPCA) while FRPCA gives better performance than KPCA. Hence, data mining plays an important role in reducing complexity of computations and achieving profitable predictions. Evaluation metrics, root mean square error (RMSE) and mean absolute error (MAE), are used in [4] for performance analysis while performing prediction for 1, 3, 7, 14 hours (length of time steps) for four models. Results show that proposed hybrid model Seq2Seq LSTM and CNN outperforms baseline models on given input dataset.

In [5], proposed techniques SVM, RF and L1-LR filters the following parameters as highly informative features: CO, MACDH and TRANGE while LL, CCI, and MACDH as least informative features. Further, STOCKNET gives best performance with mean accuracy 57% and mean MCC 0.00475 as compared to other baseline models. Moreover, combination of STOCKNET with multi filter feature selection enhances its mean accuracy and mean MCC values at 59% and 0.1030 respectively. Result in [6] shows that proposed combination of model outperforms with accuracy 65.4% than other models used in literature. It handled the temporal dependencies of news titles and their effects on stocks behavior efficiently. Further, its comparison with different companies have proved its efficacy and robustness. Study [7] show that the comparison of baseline models SVM, LSTM and proposed technique through evaluation metrics TPR, TNR and the POS. It reveals that proposed model gives higher POS value for trading strategy. Empirical results show that though model does not achieve 100% accuracy however, it is profitable yet. In comparison to earlier studies, the proposed method yields better results for the DCO signs with reduced regression error.

b) Prediction

Low generalization ability of traditional financial models in [8] had constrained their performance to capture complex market movements. Its main reason is that previous time series models only focused on linear features of data while ignoring non linearity of stocks data. A model with capability of handling non-linear, complex and chaotic nature of dataset with robust competence, can detect shocks within a day. The world of Bitcoin is emerging in the financial stock market rapidly so that massive algorithmic trading-bots are growing each day. In [9], over 60% of trading volume estimation depends on these bots, due to which making profitable trading is tedious within a short time period. Huge amount of cryptocurrencies data is difficult to handle for preprocessing and computations. This study aims to build a bot system which helps in making profitable trading decisions for maximum returns in comparison with US dollar. The work in [10] presents the the complex and non-linear nature of stock data. It shows that many attempts have been made to generate good results for investors. However, markets volatility is increasing day by day which has made it difficult for a predictive model to forecast company success.

The authors in [11] discovered that by using data of a company's success would generate frequently biased (or failure) results. Such a strategy is a prime illustration of the look-ahead bias. It produces extremely positive test results however, any attempt to apply such a strategy in a real-world situation may have disastrous effects.The work in [12] ensures that the demand of both linear and non-linear components of stock data should be fed to machine learning (ML) models. So, that more generalized results could be generated by the combination of algorithms. Sequence for Bitcoin is composed of complex and highly unstable behavior of signals. Previous studies focused on linear dependency evaluation between Bitcoin exchange rate and predictive variables without verifying its linear or non-linear characteristic[13]. This leads to inaccurate predictions. It is humanly impossible to inculcate each informative factor from dynamic and complex trading data in study. Previously used supervised learning techniques encountered some limitations that is portfolio optimization problem (POP) or portfolio selection problem (PSP) and minimization of prediction error without considering risk factor, transaction cost and illiquidity [14].

Complex data needs to be explored and partitioned in such a way that only informative features are captured and given to the proposed model in [15]. The work in [16] tackles the gap of limited literature of cryptocurrencies except Bitcoin. It ignored the impact of high frequency stock signals on the prediction of daily stock market returns. Study [8] uses ML algorithms on high frequency stock signals to predict daily market returns. American stock market pricing data, provided by S&P 500, on the basis of 5 minute time intervals is utilized in this study. A hybrid machine learning model named ARMA-GARCH is employed to predict the number of shocks along with the time of their occurrence in a day. Further, this model extracts informative features using forward selection with minimal redundancy maximal relevance (FSMRMR) criterion, which increases the model's accuracy. This step drastically reduces the correlation and redundancy of input features of the dataset. Also, a novel technique, nearest-k cross-validation (NK-CV), is applied for the performance check of a neural network (NN) model. It utilizes the threshold technique to decide whether to make a prediction or not. After all these stages, ensemble voting is performed at the end to generate the final output.

Paper [9] implements and compares the proposed models to figure out a suitable combination for the day traders on the basis of short time intervals (5 minutes) for Bitcoin price estimation. This process involves the following models: autoregressive integrated moving average (ARIMA), random forest (RF), the prophet (from Facebook), and multi-layer perceptron (MLP) from neural networks. Open-high-low-close chart (OHLC) data with a 5 minutes time interval is utilized in this study. In [10], a comparison of the performance of proposed techniques, LSTM and ARIMA, is performed for forecasting. LSTM minimizes the problem of investors in the appropriate selection of shares for profitable prediction results. Following evaluation metrics, RMSE, MAE, MSE, and Regression scores, are applied for performance evaluation. Auto correlation coefficient and partial correlation coefficient curves are plotted to show inter-dependency among variables at different time steps. Dell's daily stock data is utilized for this purpose. Paper [11] compares the following three algorithms: gradient boosting classifier, support vector machine, and logistic regression. This study uses business information data containing 213171 businesses in its training group, known as Church-base. Input data pre-processing is done by feature scaling techniques using logistic regression (min-max normalization) and SVM (standardization). The selected subset includes the data of businesses founded between 1995 and 2015. In order to prevent over fitting in the validation set, cross-validation is employed on the training set. Further, accuracy, precision, recall, and F1 score are calculated for each fold of a validation set.

Study [12] proposes an approach, named the Chained Paasche approach, based on single imputation, for predicting real estate prices. This approach is applied at different time intervals. At each interval, hyperparameter tuning is done using a five k cross-validation. This effectively reduces overfitting and generates more generalized results in testing data. Paper [13] uses public stock market data, which is obtained from API and websites. Input data is pre-processed by data cleaning, data splitting, and min-max scaling. This study is conducted in two steps. Firstly, informative signals are screened out using machine learning models RF and ANN. In the second step, these collected factors are combined using LSTM and passed into a combination of these four models: ARIMA, SVR, adaptive neuro-fuzzy inference system (ANFIS), and LSTM. They generate the final prediction of the daily exchange rate for Bitcoin. In the end, a comparison of the proposed technique with prior models is done, using the previous exchange rate, to show its effectiveness.

The prime objective of the study [14] is to influence investors by minimizing their investment loss thereby maximizing profit. For this purpose, reinforcement Learning (RL) is employed to cope with the drawbacks of supervised learning. Markov decision processes (MDP) are employed to cater time dependency of financial data. Twin delayed deep deterministic policy gradient algorithms (TD3) is used to reduce errors during the learning process of the model. Moreover, these ten famous indicators are used which are given as follows: Relative strength index (RSI), Simple moving average (SMA), Exponential moving average (EMA), Stochastic oscillator (%K), Moving average convergence/divergence (MACD), Accumulation/Distribution oscillator (A/D), On-balance volume indicator (OBV), Price rate of change (ROC), William's %R and Disparity index. Further, the Yahoo Finance dataset is used to retrieve historical 615 market daily prices in this study.

Study [15] employs LSTM for the detection of stock signals and the prediction of their prices. The main objective of this study is to handle aggressive and defensive portfolios of the market. For this purpose, stock data is divided into two categories: aggressive and defensive signals. Then, their pairs are converted into new portfolios using an integration test and fed into the LSTM model for finding profits. Yahoo Finance, America's daily basis dataset taken from 2008 to 2017 is used in this study. Further, data normalization and transformation are done in a pre-processing step. Paper [16] utilizes the dollar-denominated crypto-currency data from the Bitfinex exchange. It contains twelve liquid crypto-currencies data from 2013 to 2018. This study employs the following machine learning techniques on the dataset: Logistic regression, Artificial neural network, Random forest, Support vector machine, and Multi-layer perceptron. It uses past price data along with technical indicators to extract non-linear dependencies between input features.

Results in [8] show that the proposed method, Minimal relevance and minimal redundancy (MRMR) with nearest-k cross-validation, gives higher accuracy for upward and downward trend prediction of stock signals. Further, these results are tested on datasets containing data of two scenarios, upward (Sep to Oct 2007) and downward (June to July 2008) trends, respectively. Moreover, it outperforms the baseline model, conditional mutual information maximization (CMIM) with random and k-nearest cross validation, with notable results in terms of accuracy. While in the other half same tests are re-conducted on another stock dataset spanning from 2009 to 2010 for a robustness check. Findings show that the proposed hybrid model, ARMAGARCH-NN, generates more consistent results than LSTM. Hence, it can be used as an application for trading signal shock prediction to ensure profitable outcomes. Results in [9] portray that combination of MLP and LSTM has surpassed the function of ARIMA and Prophet model with 54.09% accuracy. While results in [10] shows that, performance of ARIMA in short term trend forecasting is outstanding as compared to LSTM, which shows better results for long term forecasting. Further, actual and predicted curves, are lined up portraying negligible overfitting. Moreover, evaluation metrics show that ARIMA has higher F2score than LSTM.

The model proposed in [11] shows that non-linear models are better for complex and scattered data in case of real estate market than linear ones. These methods are flexible enough to best fit the patterns of input data. Moreover,findings of evaluation metrics show that non-linear models SVR, XGBT and avNNet generate low RMSE and high R-2 value, which supports the above argument. The authors in [17] present, a novel study for inflation forecasting of the Central bank of the republic of turkey (CBRT) dataset using machine learning techniques. These techniques use informative indicators, which minimize loss functions, to generate outstanding results in out-of-sample data. In the end, a comparison of the following machine learning techniques: Ridge, LASSO, ElasticNet, and adaLASSO is carried out with the following benchmark models: Vector autoregression model (VAR) and ARIMA. Overall, in [12], the comparison of non-linear machine learning techniques and linear ordinary least square methods shows that non-linear techniques generate less volatile auto correlation of profit returns with higher accuracy for highly dimensional data.

The comparison of the proposed model with baseline models is performed using different performance metrics like RMSE, MAPE, MAE, and DA in [13]. Numeric figures and box plots clearly reveal that LSTM generate higher DA with significantly low regression metrics using the economic and technology predictors. Two back-testing approaches are proposed to evaluate the performance of the proposed technique in order to validate robustness of model using 10 different assets to ensure its ability to solve trading problem [14]. Two metrics, cumulative sum of rewards and annualized sharpe ratio, are used for performance analysis. The work in [15] utilizes sharpe ratio, MSE and MAPE for performance analysis. Findings reveal that the proposed model generates higher annual sharpe ratio with 2.68 value and lower error rates. This reduction in values is created due to inclusion of technical indicators and sentiment scores of news in analysis. Moreover, LSTM handles arbitrage opportunities to make profits and its predictions give generalized results for real time situations in both aggressive and defensive portfolios. Average classification accuracy is used in [16] to check performance of these four models: RF, ANN, LR and SVM. All proposed techniques have generated values 50% above than threshold. It shows that these techniques can be used for predictability upto a certain degree in cryptocurrency markets. Particularly, SVM outperforms than all other techniques.

2.1.1.2 Asia

a) Forecasting

Inflation is a dilemma for economic world, since actions intended to lower inflation may exacerbate unemployment[17]. Major obstacles in accurately predicting the inflation periods includes the curse of dimensionality and lack of informative variables in input. This study is crucial because prior prediction of inflation periods save investors and market from huge loss. The work in [18] highlights Covid-19 affected intensive trade areas of Turkish stock market. During Covid period, from 2021 to 2022, Russian invasion on Ukraine adds more critical situations in economic world. Due to these crises, Borsa Istanbul, Turkeys' capital stock market had collapsed. The authors in [19] highlight the difficulties in analyzing the sequence of stock signals in time series data and calculation of their fractional differences. All of these complexities are created due to non-linearity of input price data.

The authors in [17] present a novel study for inflation forecasting of the Central Bank of the Republic of Turkey (CBRT) dataset using machine learning techniques. These techniques use informative indicators that minimize loss functions to generate outstanding results for out-of-sample data. In addition, a comparison of different machine learning techniques, Ridge, LASSO, ElasticNet, and adaLASSO, is carried out using Vector autoregression model (VAR) and ARIMA. The empirical study done in [18] presents a novel method to calculate the impacts of Covid-19 and the Russian invasion on the Turkish stock market. A hybrid model, Elastic Net regression based on empirical mode decomposition, is employed on stock data collected from the beginning of Covid-19, i.e., from March 2020 to May 2022. Also, Markov switching regression is applied for the robustness evaluation of the proposed models.

Study [19] employed a hybrid novel approach ARFIMA-LSTM to detect nonlinear dependencies of PSX stock market data. It minimizes the volatility problem of stock data and overfitting issues. Results of [17]show that ElasticNet and LASSO algorithms generate better results than conventional methods for Turkish inflation prediction. Findings in [18] reveal that the political invasion of Russia and the natural epidemic of Covid-19 had drastic effects on the Turkish market. Moreover, the value of the Turkish lira dropped to negative values. Evaluation metrics MAPE, RMSE, and MAE are used in [19] to check the performance of the proposed non-linear combination model. Findings show that hybrid ARFIMA-LSTM generates the lowest values of 0.0539 RMSE and 0.002% MAPE as compared to LSTM, ARFIMA, and ARIMA models. Moreover, graphical results depict that the proposed one perfectly aligns the predicted curve with the actual one. That is, its accuracy is improved by 80% as compared to traditional models.

b) Prediction

In [20], [21], [22], [23], [24], [25], following factors are reason for research. Sparsely available datasets, huge retailers, centrally controlled market (State bank controlled by the government in the case of China), limited history of short sell stocks, and layman decisions. Whereas, instability of the stock market is due to its complexity, non-linearity, and correlation between market behavior and investment psychology. For making long-term stock predictions, two models/strategies are used: The capital asset pricing model (CAPM) and arbitrage pricing theory (APT). As a linear model approach for stock market predictions is outdated, a non-linear model is applied for extracting relations between stock returns and input models. Furthermore, investors in China are facing investment loss due to a lack of information and an inability to predict stocks trend. Therefore, an effective model has to be constructed on the basis of multiple feature extraction, which improves investors' predictability. As data is growing into complex large volumes, predicting intraday stock jumps is also quite challenging. Related works on intraday stock jump predictability remain scant due to its spontaneous and volatile nature. Many studies using automatic feature extraction models and time series forecasting techniques have been used individually. However, their combined effect needs to be checked.

Study [26] illustrates that, use of stock price charts and multi-layered deep neural network is an alternative approach to attain accurate predictions. Stock price prediction depends upon various factors, e.g., political events, global economic situations, interest rate, investor sentiments directly or indirectly [27]. The work done in [28] employed linear models while stock market signals are time dependent having temporal dimension. This non-linearity factor remains ignored in the previous studies while making predictions. The inherent non-linear, non-stationary and irrational behavior of stock signals is becoming complex day by day increasing its unpredictability [29].

Previous studies incorporated single models without hybridization of multiple models [30]. Due to which performance is constrained to generate affluent results having low bias-variance factors. Moreover, XGBoost is mostly used for prediction as a classifier instead of forecasting and chaotic strategy firefly algorithm (FA) for hyperparameter tuning. These traditional approaches strained boosting the portfolio of companies. The authors in [31] addressed the issue that labeled data is less effective for prediction, as it decreases the model's accuracy.

The authors in [32] solved the issue that stock price movements are difficult to predict up and down movements within a day due to its speculated nature. In [33], with the aid of stock price chart images and multiple layer deep neural network, an efficient converging attempt is made to get accurate results. Authors in [20] build a generalized model that provides an insight into the connection between predictability and retail investors. It follows the standard approach for hyperparameter tuning, model selection, and performance analysis. Input data is split into three portions given in temporal order: training data (from 2000 to 2008), validation data (from 2009 to 2011), and testing data (from 2012 to 2020). An extensive study is conducted including small and large shareholders, small and large stocks, and state-owned and non-state-owned enterprises. Long-short strategies used in the US and European market are less realistic for the Chinese market. Here, long-only portfolios are analyzed, which are more useful for investors' decisions. Moreover, a subgroup comparison of small and large stocks is conducted to evaluate heterogeneity in the predictions of the model.

A comparative study between nine machine learning techniques and two deep learning models is conducted in[21]. These techniques are listed as RF, NB, AdaBoost, LR, ANN, SVC, KNN, RNN, and LSTM. Moreover, ten technical indicators are applied to both continuous and binary data with best-tuned parameters. The authors in [22] use all A-share market stock data of China, ranging from 2010 to 2018, to predict individual stock trends. Firstly, data is split into training and testing sets, using the sliding window method. Then, data pre-processing is done by the filling method and the normalization technique. For the selection of top features from data, these two techniques are employed: SVM-RFE and RF. Finally, a fully connected network ANN with three layers is employed to predict the trend of stock price. Furthermore, all A-Share Chinese stocks are divided into ten groups to analyze the profitability.

In [23] these ML techniques are employed for price prediction in Taiwan: SVM, RF and auto regression algorithm. Further, this study uses long-term memory models to forecast the Taiwanese stock market with configuration on different time-steps using present and past data of input variables. Time series-based prediction models forecast the future value of a particular variable by analyzing past and present data of those variables. Morever, authors have analyzed the impact of different parameters of RF and kernels of SVM on prediction performance. The technical assistance and information exchange instrument dataset of over 15 years is taken under consideration. Evaluation metrics, root mean square error and mean absolute error, are employed for performance check of learning models.

Study done in [24] proposes these data-driven approaches to predict jumps in stock signals in a day: SVM, RF, KNN, and ANN. Moreover, this study utilizes technical indicators and liquidity measures. Each input stock signal of trading day is partitioned into 5 minute intervals. These transformed signals are fed to ML techniques for prediction of jump in next 5 minutes interval in such a way that binary classification is done for jump prediction while trinary classification is carried out to separate non-jumping, upward and downward signals. Dataset is obtained from Shenzen stock exchange of China to validate the proposed approach.

Authors in [25] uses these ML models: perceptron, logistic regression and SVM. NIFTY 50 dataset is utilized, ranging form 2013 to 2018 along with these technical indicators: moving average convergence divergence, average true range, exponential moving average and relative strength index. Their values are calculated keeping the closing price in view as it is a crucial part of the daily stock price. Study [26] predicts the closing price of NIFTY 50 by employing a hybrid CNN-LSTM model. It extracts informative features from rich feature data of twenty trading days for next-day prediction of stock price movement.

Study [27] inculcates return prediction factor used in previous time series models for portfolio evaluation using the following ML and DL models: RF, LSTM, SVM, CNN, and DMLP. Further, the China Securities 100 datasets spanning over 9 years are used for investigation. Moreover, stochastic gradient descent is employed as an early stopping parameter during the training of LSTM in order to reduce overfitting. In order to select optimum hyperparameters the work in [28] proposes a generative adversarial network-based hybrid prediction algorithm (GAN-HPA) along with Bayesian optimization and reinforcement learning. This hybrid model is a novel reinforcement learning to dynamically tune the hyperparameters. Also, CNN and LSTM are used as discriminators and generators in Stock-GAN for optimum feature extraction from price data. Stacked autoencoders are applied to remove noise and select final features. For stock price prediction, a multi-model generative adversarial network hybrid prediction algorithm is employed.

In order to capture hidden patterns of stock data in [29], a hybrid GA-XGBoost algorithm is proposed for achieving accurate results. Then, KOSPI data is passed to the 3-stage feature engineering engine. In the first stage, technical indicators are generated to expand the feature set. In the second step, normalization and cleaning of data are done to prepare data and feed it to the hybrid GA-XGBoost model for informative feature selection. 67 new technical indicators are generated and added to the original price data for its expansion. This transformation leads to the benefit of dimensionality in which large data proves to be helpful for solving high dimensional stock problems. Embedded with the concepts of genetics and natural selection, the genetic algorithm (GA), is employed as a fitness function for the evaluation of the XGBoost algorithm.

A novel hybrid approach IFA-XGBoost is designed in [30] to perform two tasks. First stock trend prediction is done by ML model and second to construct a portfolio using the mean-variance approach. To be specific, XGBoost hybrid with improved firefly algorithm (IFA) is employed for trend prediction of the next time period. IFA algorithm eradicates the traditional strategy of selection of hyperparameters based on experience while it performs grouped analysis to choose relevant hyperparameters for enhancement of performance. A random selection of 24 stocks of the SSE 50 index ranging from November 2009 to November 2019 along with 19 indicators comprising four technical indicators and 15 lagged return observations are utilized in this study for predictive analysis. In [31], the proposed model follows the framework of data partitioning and then initialization by inclusively generating the random number from 1 to 3 (1 for buy, 2 for sell, and 3 for monitor) categories. It is followed by search optimization done by hill climbing and aggregation of labeled partitions. It utilizes the stock market dataset of Saudi Arabia containing daily records ranging, from 1994 to 2017 of 200 listed companies.

The authors in [32] use National Stock Exchange (NSE), India, a dataset from the years 2008 to 2018. It is fed in a combined form to the candlestick data for performing predictions on a daily basis while technical indicators are used for feature extraction. Ten technical indicators are used and fed to DNN for up and down trend prediction using rectified linear unit (ReLU) as an activation function. Accuracy and F-measure values are evaluated from the confusion matrix to confirm the proposed model's robustness. Furthermore, tenfold cross-validation is employed along with the combination of technical indicators and image spacing. Paper [33] is a contribution to the literature on stock price trend prediction. As it implements a new idea of predicting stock prices in short term by taking input in the form of charts and images. For this purpose, data from all A-share companies in China, ranging from 2009 to 2020 time period, is taken under consideration. It is divided into training and validation sets. Moreover, a deep neural network with the activation function ReLU is employed. This combination overcomes the restriction of layer-by-layer time-consuming training of a model. Furthermore, the output shows the status of each stock in terms of the following labels: profitable or unprofitable in the future.

Monthly prediction of out-of-sample $\mathbb{R}^2$ value in [20] shows that non-state-owned enterprises (nSOE) have more profit in stock returns than state-owned Enterprises (SOE). Additionally, the bottom 30% are earning more profit from stock returns in the short run than the top 70%. It is favorable for retailers in the short run. Annual prediction of the out-of-sample value of $\mathbb{R}^2$% for all types of data shows that top 70% stocks are more strongly predictable than 30% due to improved predictability of SOEs. Average prediction of out-of-sample $\mathbb{R}^2$% for monthly and annual evaluation shows that Chinese small stocks are ten times higher than US small stocks. On the other hand, Chinese large stock is two times higher than US stocks. Moreover, the group of nSOEs performs better than SOEs on monthly basis period data. Tree models, Gross business receipts tax (GBRT) and RF potentially detect non-linear dependencies among macroeconomic variables and stock features. Relative variable importance is calculated for 11 macroeconomic variables, which show that ntis variable has the largest importance among others. In this study, neural networks give better results than prior literature techniques. A similar study is performed while removing small stocks from data. This shows that both Long-short and long-only portfolios achieve lower monthly returns on average, sharpe ratio, and standard deviations.

Three evaluation metrics, F1 score, accuracy, and area under the ROC Curve (ROCAUC) are calculated in [21] for tree-based models, neural network-based models, and supervised models using both continuous data. Results show that the prediction accuracy of continuous data is 67% and binary data is 83%. The two deep learning models RNN and LSTM have outperformed all the ML models with both continuous and binary data. The area under the curve (AUC) is calculated for different integrated models and RF turned out to be the best model for trend forecasting [22]. RF-RF integrated model generates the highest annual return, win rate, profit loss ratio, and sharp ratio as compared to other hybrid approaches. Input data is divided into ten equal groups. Then, each group is analyzed for its profit evaluation. Group 1 has the best performance with the highest total return, annualized return, sharpe ratio, and the lowest peak of max drawdown among the ten groups. Empirical results are promoting investors to invest in the capital market by increasing its vitality. RF-RF with a long-short trading strategy based on RF-RF integrated model practically solved the investor's problem by buying the stocks of Group 1 and selling the stock of Group 10 at the same time.

RF model is evaluated in [23] using RMSE and MAE metrics with the following number of leaf nodes N: 10, 30, 50, 70, and 90. The model performs the best for 11 years among 9, 10, and 11-year time periods. Moreover, RF, autoregression, and LSTM are applied for stock price prediction. outperforms LR and autoregression on both training and testing datasets. Results in [24] show that RF performs better in both binary and trinary classification in terms of predicting jumps and their directions than other ML techniques. Accuracy and F-measure curves portray the highest peak of RF compared to other techniques while predicting both jumps and their direction. RF performance increases using a set of both liquidity measures and technical indicators. Further, this study can be applied to other stock markets.

The authors in [25] report that perceptron, LR, and SVM achieve an average accuracy of 75.88%, 86.98%, and 87.35% respectively. Further, conversion of unlabeled data into labeled data has much improved the accuracy of these proposed models given as LR with 89.93%, SVM with 89.93%, and perceptron with 76.68% accuracies. Also, k-fold cross-validation and train-test strategies are employed on both time series and non-time series supervised learning. Further, values of TypeI and TypeII errors show that SVM outperforms among proposed techniques. The main reason is that a binary classification problem was involved in this study. The proposed model's performance in [26] on training and testing between real and predicted nifty values clearly depicts that there is no overfitting. A novel stop-loss strategy is used which protects capital against fluctuations in signals that might not be predicted by the proposed model. Further, CNN and LSTM are enriched with the capability of recognizing signal patterns to generate robust results.

Evaluation metrics F1 score and accuracy show that the proposed model performs better than previous literature. Paper [27] employs these five evaluation metrics for proposed techniques deep multilayer perceptron (DMLP), ARIMA, CNN, LSTM, RF, and SVR: Total Hit rate (HR), for positive predicted rate HR+, for negative predicted rate HR-, MAE and MSE. Results illustrate that LSTM surpasses deep models with low standard deviation and error metrics. Moreover, RF and LSTM give low values of MSE, MAE, and standard deviations as compared to other proposed models. Also, the HR value of RF is higher than the other two models. For a more detailed robustness check, the proposed model RF is hybridized with mean–variance (MV) and omega portfolio i.e.; MVF and OF models. Then, these combinations are applied to data. The following results are depicted: In the first portion, among MVF combinations, LSTM+MVF gives the highest information ratio and excess return while CNN+MVF gives the highest total return rate.

Moreover, RF+MVF generates the highest information ratio, excess return, and total return than SVM+MVF. In the second portion, among OF combinations, LSTM+OF and CNN+OF outperform the DMLP+OF model. While RF+OF generates relatively satisfactory and stable results throughout the testing period. Hence, it is proved that RF+MVF and RF+OF perform the best in all combinations.

The work in [28] utilizes MSE and MAE for the performance check of the proposed hybrid model Generative Adversarial Network based Hybrid Prediction Algorithm (GAN-HPA). Lower error values show closeness between predicted and actual stock price values for all Bharat heavy electricals limited (BHEL), Maruti Suzuki India Limited (Maruti), tata consultancy services (TCS), and Western India palm refined oils limited (Wipro) datasets.

Study [29] presents that the proposed hybrid model, GA-XGBoost, gives a 94.08% F1 score, 94.41% recall, 93.79% accuracy, and 93.75% precision. Moreover, for its generalizability check, it is compared with the following 6 deep models: 1-Layer LSTM, 2-Layer LSTM and 3-Layer LSTM, 1-Layer BiLSTM, 2-Layer BiLSTM, and 3-Layer BiLSTM. GA-XGBoost surpasses all 6 DL models by acquiring the highest accuracy of 93.68% on the validation set and 93.82% on the testing set with default hyperparameters. Moreover, GA-XGBoost is also compared with these feature extraction techniques: RF, DT, Extra tree, KNN with GA, and SVM with RBF. GA-XGBoost outperforms all baseline models. In addition, it provided an accurate prediction that, there is a 98% probability that the price trend for the 345th day will decrease while the remaining predicted an increase with 2%. This prediction matches with actual results, which is a downward price trend.

These regression metrics are used for checking accurate forecasting results in [30]: RMSE, MSE, MAPE, and MAE. The proposed model referred to as IFAXGBoost+MV generates lower error indexes as compared to these single and hybrid models, LSTM, ANN, SVR, XGBoost, extreme learning machine(ELM), Particle swarm optimization XGBoost (PSOXGBoost), hyperparameters of XGBoost optimized by firefly document (FAXGBoost) and hyperparameters of XGBoost optimized by salp swarm algorithm XGBoost (SSXGBoost) elicits its effectiveness. Moreover, the Improved Firefly algorithm with mean–variance XGBoost (IFAXGBoost+MV) generates the highest values of Sortino ratio, Sharpe ratio, and annual return rate. IFAXGBoost+MV performs better in terms of risk-return metrics, return, and risk characteristics. Graphical results and box plots depict that it generates the highest cumulative returns without and with transaction costs. Hence, IFAXGBoost+MV proves to be the best approach for both assigned tasks.

In [31], the objective function is used as an evaluation metric, which shows that automatic labeling using optimization search outperforms traditional approaches. Comparative results for automatic labeling without data partitioning elicit that hill climbing performs the best in labeling sequence, gaining the highest value on the objective function than simulated annealing, and manual labeling.

The proposed 5-layered neural network in [32], with candlestick and technical indicator approaches, gives outstanding results than the literature. Moreover, results depict that housing development finance corporation bank (HDFC), Reliance stock, and Infosys generate 0.8211, 0.8361, and 0.8178 accuracy values. The strategy proposed in [33] generates 62% accuracy on training data and 55% accuracy on validation data. This decrease in validation data highlights the overfitting issue. Moreover, the single-layer neural network gives worse results on validation datasets with 50% accuracy as compared to the deep-layered neural network. This difference can be noted at both extreme ends of time intervals. This means DLNN has the capability of capturing the patterns of trend graphs and mining the informative features from them. Hence, DLNN is considered an effective statistical tool for technical analysts and investors to predict prices using graphical data.

2.1.1.3 Europe

a) Prediction

Covid-19 caused disastrous impacts on the global economy. Researchers faced issues in the prediction of financial stock market trends due to its abrupt behavior caused by the influence of rising volatility and uncertainty in this time period. In correlated European stock markets, detecting the major cause that either inside or outside the European stock market played an important role in fluctuations is quite a challenging task [34]. The main objective of [35] is to handle the complexity of multiple real estate market data.

The authors in [34] propose an ML technique, Least Absolute Shrinkage and Selection Operator (LASSO), to perform regularized analysis by reducing the issue of limited data. This technique also filters outliers and deals with the multi-colinearity of data. This study uses data in two categories: the first is before the World Health Organization (WHO) announcement of the pandemic (Jan to March) and the second is after the WHO Covid-19 declaration (March to June). Data samples are taken from EURO STOXX 50 index dataset. LASSO can handle both default sample and subsample periods using coefficient shrinkage. In [35], three ML techniques, XGBoost, ANN and ElasticNet, are employed for house price prediction in two cities of Italy: Brescia and Varese. Dataset of house prices of these cities is collected from Application programming interface(API) protocol. Exploratory Data Analysis (EDA) is employed to undercover the underlying structure of data and extract important variables. Moreover, this is the first study that compares the proposed ML techniques in house price prediction domain. Results in [34] reveal that the following indices mostly affect European stock market: S&P 500, Spain, Germany, Switzerland and France. As LASSO selects the informative variables while droppping irrelevant ones, it aids in giving accurate predictions. In addition, LASSO coefficent paths clearly depict that France and Germany are the top positive predictors for analyzing European market trends in both default periods: before and after pandemic. Empirical results of [35] are obtained by taking MAE for performance analysis of XGBoost, ANN and ElasticNet. Findings reveal that ANN outperform other two models with 5% less MAE value than XGBoost, which gives the second best performance. Hence, it is demonstrated that when small to medium sized houses are included in the dataset, than employing artificial intelligence (AI) techniques would be effective.

2.1.1.4 Hybrid

a) Forecasting

Many studies have been performed to build an optimal model for making an automatic trading decision in financial stock markets. However, with the passing of time, stock markets are becoming more complex, dynamic, and unpredictable. Due to this, these issues emerged: imbalanced data, a high correlation among input parameters while a low correlation between input and output parameters, and cost-sensitive performance analysis. These factors drastically affect the performance of the prediction model. Therefore, an optimal model with low generalization error needs to be developed vigorously [36]. It is found in [37] that data used for research contains an intrinsic component of noise along with useful information. The authors in [38], address the issue that Bitcoin price data contain unstable fluctuations and noise.

Paper [36] cater its limitations by proposing a hybrid model RF-WMGEPSVM which is the combination of weighted multicategory generalized eigenvalue support vector machine (WMGEPSVM) and RF. It handles imbalanced data efficiently and generates buy/hold/sell signals for investors to help them in taking profitable steps. RF is an efficient technique for optimum feature selection. Here, it increases the model's generalizability and efficacy. The following five datasets are involved in this study: NIFTY 50, DOW JONES, S&P 500, NIFTY BANK, and NASDAQ. Further, this study introduced a 'walk forward' approach to predicting daily automatic trading decisions. For the calculation of the importance score of input features, out-of-bag (OOB) data samples are utilized. Moreover, 55 technical indicators are selected from the literature through an exhaustive study. This extraction is taken place on the basis of their application in technical analysis, which helps in getting promising results.

Authors in [37] employs deep neural networks for optimum feature selection by changing their architecture and hyperparameters to select the best one for accurate trend prediction. Further, 124 technical indicators, picked from recent literature and trading websites with high hit rates, are included in the investigation domain. Then, these selected features are filtered to remove redundancy and noise. This task is performed by these three methods: LASSO, sequential forward floating selection (SFFS), and tournament screening (TS). Daily basis data collected from 7 global stock markets, ranging from 2008 to 2019, is used in this study. Moreover, proposed neural networks are tested by frequently changing the number of hidden layers and dropout rates, which reveals the networks' generalizability.

The authors in [38] employ a deep feed-forward neural network (DFFNN) to forecast the trend of high-frequency Bitcoin price data. DFFNN is applied with different combinations of the following methods: resilient algorithm, Levenberg Marquardt algorithm, and conjugate gradient. Moreover, 5-minute interval data, from 2016 to 2018, containing 65,535 samples is taken under consideration. Paper [36] shows that the hybrid approach WMGEPSVM outperforms both the Buy and Hold technique and variants of SVM preference-enhanced support vector machine (PSVM), balanced multicategory support vector machine (BMPSVM), and Least squares twin support vector machines (LSTSVM). The proposed model produces high metric values of return on revenue (ROR) and percentage profitability (PP) with low maximum drawdown (MDD) on 500 trading days from all five datasets. A clear depiction of improved performance using RF hybridization with different prediction models on the same test dataset is achieved. Hence, it is proved that RF-WMGEPSVM is effective in all bearish, bullish, or flat-trend real market scenarios. Results of [37] show that BUY/HOLD technique outperform trading strategies using both literature and market-producing transaction costs and trading profitability for assets of the DAX-30 Index.

Importance score of both literature and market technical indicator is calculated. Evaluation metric RMSE in [38] depicts that the combination of DFFNN with Levenberg-Marquardt algorithm outstands than DFFNN with resilient algorithm and DFFNN with Powell-Beale. Moreover, it is found that as resilient algorithm works rapidly and effectively. Hence, they can be used in online trading domain.

b) Prediction

The authors in [39] highlight that time is consumed by forecasting techniques to study price movements. Discontinuous data has lost some of important information which results in defective and inaccurate predictions. Stock market prediction is a challenging problem for complex non-stationary data with high heteroscedasticity [40].

Paper [39] proposes directional change (DC) strategies on 10 minute interval data from 20 pairs of Forex currency spanning over 10 months. This study conducts comparison of 7 benchmarks including DC and non-DC based strategies and 5 thresholds including tuned and untuned DC datasets for an elaborative study. The authors in [40] present three main hypothesis on which prediction of stock market depends: efficient market hypothesis wherein past data is insufficient for prediction of future markets, adaptive market hypothesis (AMH) wherein according to AMH wherein markets are not irrational rather they are run by greed and fear. AMH believes in evolution wherein pricing evolves according to time and predictable patterns appear over time and technical versus fundamental wherein the former looks over the pricing trends while the latter investigates the economic variable behind a certain movement of the graph. This paper proposes a unique study where technical analysis and ML techniques are jointly used to gain a profitable trade. Four ML techniques are used: LR, RF, ANN and SVR. Two technical analysis tools are used in the study given as moving average convergence and divergence (MACD) and triple ex-moving average tool (TEMA). The proposed strategy is tested on 3 indices, IBEX, DAX, and DJI, on the data ranging from 2011 to 2019. Four evaluation metrics are used: RMSE, MAE, SMAPE and MAPE.

Empirical evidence in [39] shows that the proposed models are effective in capturing the trend reversal and providing an alternative signal to traders to improve their decision making. Furthermore, regression models successfully capture the point where trend of signal is expected to be reversed. Combining regression and classification models considerably lowers the error rate as compared to other DC based approaches. Moreover, findings reveal that the proposed approach generates lower risk with increased profit and generates good results with different performance metrics.

In [40], paper ranked the four ML techniques based on performance (LM, ANN, SVR, RF). It shows that linear model is the best while RF is the worst among all. Also, four technical indicators are ranked based on high profit factor DJI (highest profit factor), DAX (normal profit factor), IBEX (lowest profit factor). MACD has high profit factor for DAX and IBEX while TEMA has high profit factor for DJI, TEMA, MACD turns out to be profitable and competitive for the DJI. Hybridization improved the performance of all the indices and even of those who are already having good results. Therefore, using the hybrid MACD is better than using simple MACD or TEMA. Hybrid MACD outperforms hybrid TEMA on DAX, DJI and underperforms on IBEX.

TABLE 2.1: Related work

Country	**Dataset**	**Objective**	**Proposed model**	**Perform-ance metrics**	**Limitation**
United States	Standard and poor's 500 (S and P 500)	To solve class imbalance of recession periods [1]	wGBM	AUC, ROC and AUROC	Appropriate weights should be selected to reduce computational overhead
United States	S and P 500	To remove noisy components from input signal [2]	EMD and sampEn	MAPE, WAPE and DA	EMD and sampEn is sensitive to noisy data and parameter selection.
United States	S and P 500	To remove curse of dimensionality [3]	PCA, KPCA, FRPCA and ANN	Mean of daily return and Sharpe ratio	PCA assumes linear dependency among features of non-linear stock data.
United States	S and P 500	To combine linear and nonlinear components in input signal [4]	RCSNet, ARIMA, CNN, Seq2Seq LSTM	RMSE, MAE	RCSNet couldn't handle non-linear complexities of time series stock market data while ARIMA utilizes previous input values for future prediction, which is not effective. Moreover, LSTM requires more memory for learning.
America	NASDAQ	To extract features from complex data [5]	L1–LR, SVM, RF and STOCKNET	Accuracy and MCC	Lack of interpret-ability to explore underlying factors
America	PepsiCo, NRG, Telegraph (ATandT), APEI and American Telephone	To handle unstructured data [6]	Word2Vec and LSTM	Accuracy	Word2Vec is inefficient to train on non-linear components of data

America	Dow Jones and S and P 500 indices	To handle volatility of imbalanced data [7]	ZCR and LSTM	RMSE, MAE, POS, TNR and TPR	ZCR is frequently applied for signal processing instead of stock trend prediction
United States	S and P 500	To predict shocks within a day [8]	ARMA-GARCH-NN, mRMR, CMIM	Rand-CV, NK-CV	Proposed technique couldn't control time sensitivity of signals.
United States	Coin base (2018)	To handle big data fusion of crypto-currencies [9]	ARIMA, Prophet, RF, RF Lagged-Auto-Regression, MLP	Random guessing, Momentum-Based Strategy	Performance of prophet model is limited for uni variate modeling
United States	Dell daily stock	To deal with complex nature of data and predict trend [10]	ARIMA, LSTM	F2 score	Computa-tional complexity and prediction accuracy tradeoff needs to be considered.
United States	Crunch base database	To tackle biasdness of input data [11]	LR, SVM	Accuracy, precision, recall, and F1 score	Value of RECALL is reduced due to limited data.
New York,United States	S and P 500	To include non-linear and non-parametric models [12]	SVR, XGBT and avNNet	RMSE, R2 score	Employed neural network are prone to overfitting
America	Yahoo YQL Finance API	To include both linear and non-linear characteristics in evaluations [13]	ANN, RF and LSTM	RMSE, MAPE, MAE, and DA	Employed deep models are time consuming due to hyper-parameters.

Chapter 2

America	Yahoo Finance dataset	To overcome POP [14]	RL and DDPG	RMSE, MAE and MSE	DDPG can perform well only for large number of samples.
America	Yahoo Finance dataset	To explore complex data [15]	LSTM with beta coefficient test and augmented engle-granger two-step co-integration test	MSE, MAPE	This study didn't Perform analysis with transaction cost.
America	Bitfinex exchange	To tackle limited data of cryptocurrencies and perform high frequency analysis [16]	SVM, LR, ANN, MLP and RF	Accuracy	Outcomes of MLP lacks interpretability.
Turkey,Asia	Electronic Data Delivery System (EVDS)	To remove curse of dimensionality and extract informative variables in inflation forecasting [17]	Ridge, lasso, Adalasso, Elastic net, ARIMA and VAR	RMSE, MSE	Adalasso and lasso are slow procedures due to parameter tuning.
Turkey,Asia	XU100 index	To calculate the effect of Russian attack on Ukraine and Covid-19 on Turkish stock market [18]	ElasticNet regression with EMD	Accuracy, recall, precision and recall	Linear ElasticNet model couldn't capture non-linearity of data so well

Chapter 2

Pakistan	Pakistan stock exchange (PSX)	To analyze the sequence of observations in time-series data for forecasting [19]	ARFIMA-LSTM	RMSE, MAE and MSE	Selection of appropriate hyper-parameters is desired for optimal results.
China	Wind database	To overcome the issue of limited data due to centrally controlled system of China [20]	NN1, NN2, NN3, NN4, NN5, ENET, GBRT, PLS, RF, VASA, OLS and OLS-3	Sharpe Ratio, $\mathbb{R}^2$, standard deviation, monthly return rate	GBRT has high computational cost due to training of multiple trees.
Tehran	Tehran stock exchange	To cater instability caused due to correlation within input variables [21]	DT, RF, Adaboost, XGboost, SVC, Naïve Bayes, KNN, LR, ANN, RNN and LSTM	F1 score, Accuracy and ROCAUC	Combination of multiple ML and DL models has increased model's complexity.
China	Chinese A –share market	To construct a non-linear model to improve investor predictability [22]	RF, SVM-RFE and ANN	Winrate, Annual return, Sharpe ratio, Profit loss ratio and AUC	high complexity due to training of muliple SVM models
Taiwan	TAIEX dataset	To tackle volatility of stock signals for long term prediction [23]	SVM, RF and AR	MAE, RMSE	AR assumption that underlying stock patterns remains the same hinders its learning over data.

China	Shenzhen Stock Exchange of China	To handle imperceptibility of signals for intraday shock prediction [24]	RF, SVM, KNN and ANN	Accuracy and F1-Score	Inadequate hyper-parameter tuning
India	NIFTY 50	To handle non-linearity of data [25]	SVM, Perceptron and LR	k-fold cross-validation, Type I and Type II error	Lack of technical indicators in input data reduces its performance.
India	NIFTY 50	To formulate stacked framework for informative feature extraction and time series forecasting models [26]	CNN and LSTM	F1-score, accuracy	High computational cost and model's complexity.
China	China securities 100 index	To involve both direct and indirect factors in analysis [27]	RF, SVR, LSTM and Stochastic gradient descent	MSE, MAE, HR, HR+ and HR-	Irrelevant features of input data reduced the efficacy of model.
Asia	BHEL, Maruti, TCS, Wipro dataset	Temporal dimension of stock signals is ignored [28]	MMGAN-HPA	MAE, MSE	Stock-GAN couldn't perform well due to its inefficient training over data
China	KOSPI data	To tackle irrational behavior [29]	GA-XGBoost	RMSE, MSE, MAE and MAPE	Input data needs to be increased for effective performance.

China	SSE 50 index	To employ hybridization of multiple models [30]	IFA-XGBoost	RMSE, MSE, MAE and MAPE	Employed tree-based XGBoost is inefficient to capture non-linear details of stock data.
Saudi Arabia	Saudi Stock Market (Tadawul) dataset	To address the issue of reduction of effectiveness of the data labeling [31]	Hill climbing, Simulated annealing, and Manual labeling	Objective function	It didn't Address the correlation between labelling and accuracy.
India	National Stock Exchange (NSE)	To predict up and down trend within a day [32]	DNN	Accuracy and F measure	DNN may not consider external factors affecting prediction results.
China	SZSE and SSE dataset	To use stock price charts with multi-layered deep neural network is to attain accurate predictions. [33]	DLNN, ReLU	Average accuracy rate	DLNN may not effective on high dimensional stock data.
Europe	EURO STOXX 50 index	To detect major cause of fluctuations of signals either inside or outside European country [34]	LASSO	LASSO Coefficient	Employed LASSO is ineffective for highly correlated input data.

Chapter 2

Italy	API protocol, Brescia and Varese.	To capture the complexity in the variety of assets of current real estate market. [35]	ElasticNet, XGBoost, ANN, EDA	MAE	ElasticNet performance is limited for specific stock.
Hybrid	S&P 500, NIFTY BANK, NASDAQ, NIFTY 50 and DOW JONES	Tackles high correlation among input parameters while low correlation between input and output parameters [36]	RF-WMGEP-SVM	ROR, PP, MDD	This study lacks in the exploration of Tradeoff between generalization ability and forecasting accuracy.
Hybrid	CAC40, DAX 30, FTSE 100, Bovespa Index, S&P 500 Index and NIKKEI 225	To explore the effects of literature gap as compared to indicators applied in market [37]	LASSO	Accuracy, precision, recall, and F-Score	LASSO proceed by removal of some features completely which may lose important information of input data.
Hybrid	Hybrid	To handle Bitcoin pricing data [38]	DFFNN	RMSE	DFFNN lacks interpretability of its results for effective decision making.
Hybrid	20 Forex currency pairs	To identify security and market trends to maximize trading returns with minimal associated risk [39]	DC based strategies	Accuracy, Precision and Recall	This study didn't explore customized classification algorithm.
Hybrid	IBEX, DAX and DJI.	Tackle non-stationary data with high heteroscedasticity [40]	LM, ANN, SVR and RF	MAE, MAPE, RMSE, sMAPE	Time horizon can be optimized.

2.2 Problem Statement

This section presents the problem statement of the thesis. The major problem in predicting financial stock trends is the complex and chaotic nature of input stock signals. In [6], it is reported that large and unstructured price data is handled by applying hybrid machine learning approaches for stock price forecasting. Stock signals revolve around the cycle of expansion, recession, depression and so on. Majority of the investors and retailors are constantly striving to maximize profit returns with minimal associated risk. Therefore, stock signals having scarce recession periods originate class imbalance problem [1]. Furthermore, previous work in [12, 13] and [17] outlined that the non-linear temporal dimension of signals is ignored for the ease of computations leading to inappropriate results. Limited data or noisy data having outliers generate biased results with high heteroscedasticity. In order to gather data in a specific range, the authors in [7], [15] and [29] performed normalization to reduce computational speed and enhance quality of results. With every passing day, financial market is expanding in its versatility and demand worldwide. The curse of dimensionality propagates from the past data [3] and is resolved through extraction of informative features from correlated patterns of parameters involved in the input stock signals [21].
Moreover, the main problem is divided into two subproblems which are presented below.

2.2.1 Problem Statement 1:

Class imbalance leads to baised results : (volatile data)
From the literature, it is seen that stock data comprises of highly fluctuating data that frequently come across class imbalance problem. It means that for binary classification of trend, instances of both classes are not present in equal number in input training data. Thus, leading to more training of classification technique for one class as compared to another. While testing on unseen data, results will be generated in the favour of dominant class. These wrong predictions lead to huge loss for investors and decision makers. Asmarket trend is quite unpredictable so equally sampled data is indeed needed for making effective decisions.

2.2.2 Problem Statement 2:

Overfitting issue in predicting stock trend causes loss: (Memorizing)
Another common issue is overfitting in machine learning techniques. Its indication occurs when the employed method performs well on training data while it fails to predict accuratelty on unseen testing data. It can be illustrated from an example. Consider a scenario in which a blind person is kept in a closed room for few days. He/she will learn

and memorize the place of things placed inside it. That is he/she gets trained for the dimensions of that particular room. When such a person comes out of the room, he/she miserably fails to walk safely and might have dangerous experience. Now, that blind person is the 'proposed classifier' who learns so well on training data with high training accuracy and fails when tested on unseen data. This leads to less testing accuracy. In short, the mismatch of training and testing accuracies is clear depiction of overfitting.

Chapter 3

System Model and Proposed Methodology

3.1 Proposed System Model:

This section details the proposed solution for the above-highlighted frequent problems extracted from an extensive literature review. The proposed solution is divided into three stages: pre-processing, In the first stage, data is pre-processed and cleaned to get an insightful analysis of signal data by checking missing values and outlier detection. To investigate the volatility issue of input signals, we have taken an extensive dataset with a timeline from 2001 to 2023 approx. This huge and abrupt signal features of input data are fed into Random forest (RF). RF estimates the importance score of each variable in the predictive analysis by ensemble methods. Further, the whole dataset is split into two disjoint subsets randomly with 70% training and 30% testing ratio. In the second stage, the training set is fed to a bunch of base classifiers named Quadratic Discriminant Analysis (QDA), Extra Tree Classifier (EXT), Xtreme Gradient Boosting (XGB), and Naïve Bayes (NB). This combination is designed keeping the diversity and versatility among these classifiers in mind. Our main goal is to generate unbiased and highly generalized results. Each of them has a different working methodology with its own constraints and benefits. In short, none of them is dependent on other. All of these characteristics of the base classifiers are necessary conditions for stimulating ensemble methods for getting effective results. On the chain, We choose three advanced heterogeneous ensemble methods: stacking, voting, and blending. Base classifiers are employed in stacking, voting, and blending ensemble architectures separately with their default hyper-parameters. Each ensemble method gets trained individually and independently with its own working mechanisms. Then, the same meta classifier AdaBoost is set in each trajectory to get trained over the predictions made by base classifiers and generate final predictions under each ensemble method's canvas. Indeed these outcomes must be different under the same

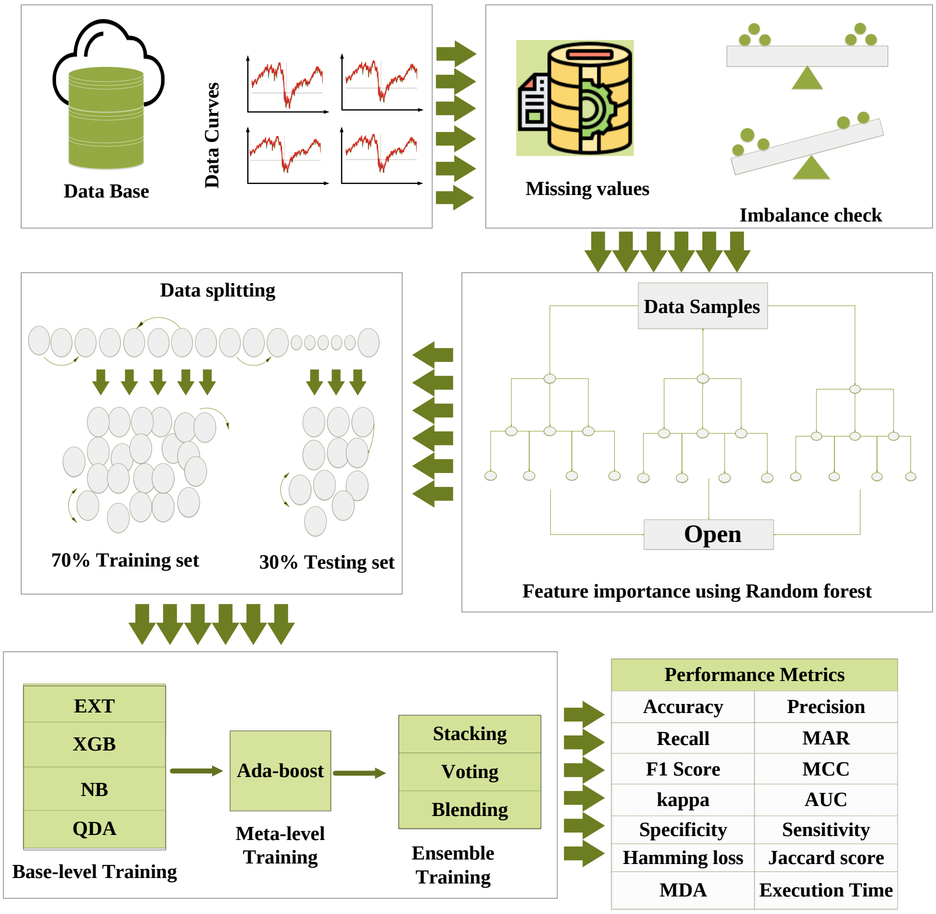

FIGURE 3.1: Proposed system model

dataset, the same base classifiers, and meta classifier. This can be clearly depicted by the formation of three distinct confusion matrices as outcomes of each proposed ensemble method under the same input and ML tools. In the end, all three models are evaluated under a variety of classification metrics for robustness check. This leads to a neutral and clear comparison providing an enriching study analysis of the ML domain under the financial stock market framework.

3.2 Ensemble Methods :

Ensemble methods in machine learning are based upon the principle of "wisdom of crowds" [41]. It states that the collective approach of multiple groups of different people or models is far much better than relying on the instinct of an individual opinion or model's result. It is proposed to ensure diversified training of machine learning techniques. This aggregation is quite profitable in concluding a result for the binary classification at hand, i.e., predicting upward or downward trend of the stock signal. Since stock data is complex, non-linear, and volatile, there is a need to deploy such a method that captures and learns all aspects of stock data. The ensemble method is enriched with all the characteristics required for diversified analysis of input stock data. It finalizes its decision based on the majority outcome of multiple independent techniques where each technique has its own strengths and weaknesses, and is trained on random input data. It is quite unlikely that all base learners make similar mistakes at the same time. The predictions made by each technique are fed to the conjunction ensemble model that generates output opting majority voting or averaging methodology. It is further verified using actual testing data. As the ensemble model is able to learns over each small detail of data, so minimal training error is incurred. Thereby, it can be visualized that such a model performs well on unseen data with few testing errors. This synchronization depicts that the model neither overfits nor underfits. Moreover, it clearly handles the bias-variance tradeoff of machine learning techniques. This provides confidence to investors, financial advisors, retailers, and regulators and make them believe in the generalized results. Thus, encouraging them to make profitable decisions.

Mathematically, let $D = \{(x_1, y_1), \ldots, (x_n, y_m)$ be the input stock data .This model inherently should be a strong learner which combines all predictions of base models into a new feature matrix. That is, for meta learner x_{train} is combined with s_i base predictions. Then, it gets trained over it along with the corresponding target variable for the validation set stored in y_{train}. This diversified ensemble of training input fed to a strong learner enables it to learn which base model predictions are coinciding with the true values of the target variable in y_{train}. This scanning leads it to effectively combine base learners for final prediction. To subsume the contribution of all base models $\hat{M}$ performs a majority voting or averaging strategy. Hence, generalized output $\hat{s}$ is generated by the meta-model on unseen testing data. Then, $\hat{s}$ is validated by the y_{test} of input data. Moreover, the higher accuracy of the ensemble model as compared to individual base models is proof of its efficacy.

Its illustration is shown below in the figure. We selected the following five machine learning classifiers to be used as base level and meta level of following three proposed ensemble methods: Stacking, Voting and Blending.

3.2.1 Classification Techniques

3.2.1.1 Extra Tree Classifier (EXT) :

EXT is an ensemble machine learning algorithm which works on boosting and it is a very famous algorithm for classification and regression problems [42]. Basically, the working nature of EXT classifier is similar to random forest. The Random Forest and Extra Trees classifiers both had comparable results. However, there are some performance disparities. Specifically, EXT exhibit low variance, random forest exhibit medium variance and decision tree exhibit high variance. It creates extremely randomized trees instead of creating one tree for decision-making. The base model is independent of each other to achieve the best performance of tree-based algorithm, and this can be done by using extremely randomized decision trees for every training model. The ability of ensemble approaches to incorporate the predictions of multiple models yields more accurate results than a single model individually, which makes extra tree classifier more robust. It creates extremely randomized number of unpruned decision tree from input data. Growing randomized tree makes greater diversity between the base models, this makes algorithm overcome the correlation between the attributes of dataset. While, dealing with larger dataset increases the computational overhead because it needs to train multiple classifiers for predictions. In extra-tree classifier the whole training dataset is used for growing multiple decision trees and this makes EXT classifier a large number of base models. A decision tree consists of root node (starting point), branch nodes (represent different conditions of attributes), and leaf nodes (represents outcomes) or decision nodes respectively. The Extra-Trees algorithm selects a split rule starting from the root node using a partially random cut point and a randomly selected subset of attributes. The EXT algorithm basically choose a split rule, which is based on random subgroup of attributes for starting the root node of decision tree. There are three main parameters of extra tree algorithm: total number of randomly selected samples (K), total number of decision trees (T), and minimum numbers of records required to split a node (S_{min}). For training of EXT algorithm, the subset of randomly selected uncorrelated samples are selected from the original dataset. The decision tree creates numbers of multiple trees from this randomly selected subset of samples. The gini index is used as a measure because it is more efficient than the entropy function. To purely split the tree in the extra tree algorithm, the trees are allowed to evolve until depth. The whole training samples are used to create each tree instead of bootstrapping. To reach the leaf node, this process is repeated in the branch node. Classification is done by number of majority voting and averaging the prediction is used in case of regression. EXT create a unique data sample for every tree from the dataset. It selects random samples for splitting the tree instead of calculating the entropy of the features. Algorithum of extra tree classifier is shown below in algorithum 1.

3.2.1.2 Naïve Bayes Classifier (NB):

This non-parametric model is employed as a base learner classifier. Naïve Bayes is combination of two concepts; Naïve and Bayes. It is named “Naïve” because the features of data participate independently in making predictions. That is, its covariance matrix

Algorithm 1 Extra Tree Classifier [42]

Input: Load dataset
Output: Result

Input Training samples S $=s_1, s_2, ..., s_q$, where $s_i = f_1, f_2, ..., f_D$ samples are D-dimensional vector, K is a randomly selected features from dataset, n_{min} is a required minimum number of samples for splitting a node.
If $Q_p < n_{min}$ all observations have unique label in the node. select the node as leaf node and stop splitting
Else Select a random subgroup of K features $f_1, f_2, ..., f_k$ from original features randomly.
For each feature k in the subgroup **do:**
Pass the training data G_m (x) to the classifier
Find f_k^{max} and f_k^{min} as the maximum and minimum values in the subset of S_p of the features k.
Obtain a random cut-point, f_k^c, uniformly in the range $[f_k^{min}, f_k^{max}]$.
Set $[f_k < f_k^c]$ as a candidate split.
End for
Select a split $[f_* < f_k^c]$ such that S core $(f_k^c) = min_{k=1,...,k}$
Output: best split $[f_* < f_k^c]$ at the child node p

is diagonal matrix with zero covariance on off-diagonal entries [43]. This characteristic aids it to simplify complex datasets and make the efficient probabilistic classification. For example, the stock market trend in the next day is predicted as 'upward' due to some features. The presence of these features may depend upon each other but all of them contribute independently to the probability that the trend will be "upward" [44]. This also reduces biases in predictions and performs well with categorical attributes. Secondly, the word 'Bayes' represents Bayes theorem which is used to estimate the prior probability $P(y_t)$ and Likelihood $P(x|y_t)$ to calculate posterior probabilities for both 'upward' and 'downward' trend class. Their formulas are given in algorithm 2, final prediction is done on the basis of higher value of posterior probability of particular class. The estimation of a probability distribution is dependent on the selection of the right training set. Moreover, the power of Bayesian networks is more evident when N is large [45]. This probabilistic analysis is clearly shown in algorithm 2.

3.2.1.3 Xtreme Gradient Boosting Classifier (XGB):

XGBoost is a non-metric classifier [47] that extracts features based on their importance using a decision tree at the backend, thereby optimizing residual errors at each step. This tree-making process continues parallelly until the predicted value matches or becomes closer to the actual one. Moreover, these trees are formulated on a random subset of the

Algorithm 2 Naïve Bayes Classifier [46]

Input: Training data $D = \{(x_i, y_i)\}$ $(x_i \in \mathbb{R}^n, y_i \in Y)$
Output: Classification Results

1: Target column has two classes m=2 i.e, 'upward' and 'downward' trend.
2: **for** $i = 1\ to\ T$
3: Fit the classifier $y_i(x)$. Calculate $P(C_k|X)$ using Bayes theorem,
$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)}, 1 \leq k \leq m$ (1)
4: **Maximum Likelihood Estimation**
$Given(x_i, y_i) \in C_i \iff P(C_i|X) > P(C_k|X), i \neq k$ (2)
5: As, P(X) is equal for both classes. We maximize $P(C_k|X)$ by :
6: $P(C_i|X)_{max} = max(P(X|C_k)P(C_k))$ (3)
7: Compute $P(C_k) = \frac{freq(C_k in x_{train})}{size(x_{train})}$ (4)
8: Using naïve assumption (conditional independence), Compute:
9: $P(X|C_k) = \prod_{i=1}^{n} P(x_i|C_k)$ (5)
10: Compute $P(x_i|C_k) = g(x_i, \mu_{c_k}, \sigma_{c_k})$ (6), where μ_{c_k} and σ_{c_k} are mean and standard deviation of gaussian distribution.
11: **end for**

feature set which leads to the automatic selection of the most relevant and informative features. XGB selects one-third of k features out of n total attributes. This makes it a 10 times faster toolkit than common boosted trees. Then, multiple classification and regression trees (CARTs) are ensembled sequentially to train new decision trees by removing the residual errors of previous trees. A pictorial view of the formation of K trees for K iterations using data features at nodes with depth Z shows the predictions of the resultant trend of the stock signal at leaf nodes [48]. Initially, the random learning objective function is fitted to the data and residual errors are obtained. Then, XGBoost comes to play its part by continuously updating these functions with regularized objectives which makes the classifier converge toward accurate load predictions.

Mathematically, for stock datasets with n dimensions and m features represented under $D = (x_i, y_i)$ with $|D| = n$, $x_i \in \mathbb{R}^m$, $y_i \in \mathbb{R}$. Model aggregates K trees to predict the output label y_i given as $\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \psi$ where $\psi = \{f(x) = w_{q(x)}\}$ $(q : \mathbb{R}^m \rightarrow S, w_{q(x)} \in \mathbb{R}^s)$ represents function space of CART. In order to reduce regularization function for maximum learning of K trees 'L' is defined as sum of loss function 'l' and penalty term 'Γ', given as

$L = \sum_i l(y, \hat{y}_i) + \sum_i \Gamma(f)k$. In order to avoid overfitting L, is minimized, where $l(y, \hat{y}_i) = (y - \hat{y}_i)$ and $\Gamma(f_k) = \gamma T + \frac{1}{2}\lambda\|w\|^2$ (γ controls the depth of tree and λ handles the weight of leaf nodes). In XGBoost, parameters are calculated using gradient descent method. This process is initialized by adding trees iteratively which also reduces objective function,

$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i) + \Gamma(f)k)$.

Further, to optimize the above objective function, taylor series expansion is used to estimate final values of parameters. Algorithm of XGBoost is depicted below in Algorithm 3.

Algorithm 3 Xtreme Gradient Boosting Classifier [48]

Input: Training data $D = \{(x_i, y_i)\}$ $(x_i \in \mathbb{R}^n, y_i \in Y)$
Output: Classification Results

1: Fit XGB(D)
2: Initialize model with a constant value $F_o(x) = argmin_\gamma \sum_{i=1}^{n} L(y_i, \gamma)$.
3: **for** m = 0 to M
4: Compute the pseudo-residuals
5: Fit base learner to pseudo residuals
6: T_i =updated DecisionTree()
7: $featurs_i$ = RandomFeatureSelection (D_i)
8: $T_i.train(D_i, features_i)$
9: Compute multiplier γ_m
10: Update the model
11: **end for**
12: **output** $F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$

3.2.1.4 Quadratic Discriminant Analysis (QDA):

Unlike Bayes model, QDA didn't impose the condition of independence among data features. That's the reason QDA could capture hidden patterns of stock data so well. Moreover, it create quadratic curves using the quadratic equation $\{x : \delta_k(x) = \delta_l(x)\}$ for the segregation of data points among classes [49]. Here,
$\delta_k(x) = \frac{-1}{2} log|\sum_k| - \frac{1}{2}(x - \mu_k)^T \sum_k^{-1}(x - \mu_k) + log\pi_k$.
Also, $\pi_k = \frac{N_k}{N}$, N_k shows number of class k instances and $\hat{\mu}_k = \sum_{y_i=k} \frac{x_i}{N_k}$. These curves provides much flexibility to bend in a direction through which maximum error could be minimized. It performs extremely well on non-linear data having correlated features for binary classification. Like in our case, the classification of stock signal into 'upward' and 'downward' trend classes. Further, unlike NB, QDA doesn't have zero covariance matrix and it estimates outputs same as LDA, with a bit difference that it must estimates the covariance matrix separately for each class without considering them same for each class i.e, $\sum_i \neq \sum_k$(if equal than QDA reduces to LDA having linear decision boundary). Algorithm of QDA is shown below in Algorithm 4.

3.2.1.5 AdaBoost Classifier (ADA):

Adaptive boosting, also named AdaBoost, is a boosting ensemble learner founded in 1996 by Freund et al. AdaBoost works by creating a 'Forest of stumps'. A stump is a tree with a parent node and two leaves. These stumps are technically known as 'weak learners' s_i.

Algorithm 4 Quadratic Discriminant Analysis Algorithm

Input: Training data $D = \{(x_i, y_i)\}$ $(x_i \in \mathbb{R}^n, y_i \in Y)$
Output: Classification Boundaries

1: Train input data D on output classes Y
2: Compute class-conditional densities using Gaussian assumption.
3: $P(x|t = c, \mu_c, \sum_c) = N(x|\mu_c, \sum_c)$ (1)
where $c \in (0$ or $1)$ for binary classes and μ shows mean and $\sum_c$ shows covariance matrix for class c.
4: Compute posterior probability using Bayes theorem,

$$P(t = c|x, \mu_c, \sum_c) = \frac{P(x|t = c, \mu_c, \sum_c) \cdot P(t = c)}{\sum_{k=1}^{K} P(x|t = k, \mu_k, \sum_k P(t = k))} \qquad (2)$$

5: Separate class by x h(x) = argmax $P(t = c|x, \mu_c, \sum_c)$ (3)
6: $log(P(C = c|X = x))log(P(C = c)) - \frac{1}{2}log \quad det \sum_c - \frac{1}{2}(x - \mu_c)^T \sum_c^{-1}(x - \mu_c)$

Unlike random forests, stumps are created independently by taking only one feature at a time. As the name indicates, it boosts these weak learners by updating their weights $w_i = \frac{1}{no.of features}$for sequentially making strong predictions. This job is done by tweaking the weights w_i of misclassified samples in their favor by updating $w_i^+ = w_i * exp^{+\alpha}$, reducing the weights of correctly classified samples by $w_i^- = w_i * exp^{-\alpha}$. This increases the chance of the selection of misclassified one ocurr at multiple times and be correctly classified in the preceding stumps. The average weighted weak learners are ensembled with maximum 'amount of say' α_{max} and minimal 'total error' ϵ_i for final prediction of each class given by [50].
Adaboost is known as the "granddaddy" of all boosting techniques [41]. Random selection of features with updated weights at each stump highly reduces the overfitting issue, making it suitable for most prediction problems. Rather, it is highly affected by noise and outliers of data [51]. We use this out-of-box classifier as a meta-learner in all three proposed ensemble techniques. It makes ensemble stumps over base classifier predictions and vice versa. Algorithum of AdaBoost Meta classifier is depicted by algorithum 5.

We consider the following ensemble methods to tackle problem statements 1 and 2 for their solutions and comparative analysis among them.

3.2.2 Stacking Ensemble Architecture (SEA) :

Stack generalization named as stacking was first founded by David Wolpert. As the name indicates, this method formulates a stack of base classifiers and ensemble them to

Algorithm 5 AdaBoost Classifier [50]

Input: Training data $D = \{(x_i, y_i)\}$ $(x_i \in \mathbb{R}^n, y_i \in Y)$

Output: $Y_T(x)$

1: Initialize sample weights $w_i = \frac{1}{N}$ for $i = 1, 2, ..., n$

2: **for** $i = 1$ *to* T

3: Fit the classifier $y_i(x)$ Minimize weighted error function J_n:

$J_n = \frac{\sum_{i=1}^{T} w_i^n . 1_{y_i(x_n) \neq t_n}}{\sum_{i=1}^{T} w_i^n}$ (1)

4: Compute amount of say α_i:

$\alpha_i = log(\frac{1-\epsilon_i}{\epsilon_i})$ (2)

5: Update the weights of the data: $w_i^{(n+1)} = w_I^{(n)} \exp^{\alpha_i . 1_{y_i(x_n) \neq t_n}}$ (3)

until s_i with α_{max} and ϵ_{min} is obtained.

6: **end for**

7: Make predictions using final model:

8: $Y_T(x) = sign(\sum_{i=1}^{T} \alpha_i y_i(x))$ (4)

get highly generalized predictions at meta end. This is done by employing "k-cross validation" strategy on D_{train} that integrates base learners using winner-takes-all approach. As, stated above, QDA, XGB, EXT, NB are used at base level and AdaBoost is used as fusion classifier. This combination of four base models increases the predictive capability of classifier. However, integrated stacking structure takes place in following three steps:

3.2.2.1 k-fold Training of Data :

Firstly, the input data is splitted into distinct k-sets of equal size such that the input data, $D_{train}(x_{train}, y_{train}) = \{f_1 \cup f_2 \cup ... \cup f_k\}$ $s|t$ $\forall f_i \cap f_j = \phi, i \neq j$ condition holds. These subsets are given input to base classifiers to get them train using "leave-one-out" approach. This strategy states that each base classifier gets trained over $k-1$ subsets leaving $1/k$ portion of D_{train} [48]. This multi-dimensional fold training increases the generalizability of model and undo misfitting.

3.2.2.2 Level 0 Predictions :

In this step, trained base models $L_1, L_2, ..., L_q$ are used to make predictions over remaining $1/k$ portion of D_{train}. This iterative process continues recursively until each subset f_i served itself for testing to base learner once(shows that folds selection is done without replacement). That is, each time base model recieves different validation set and no duplicates are formed leading towards unbiased predictions. This loop runs k times, forming k predictions for each f_i. All base predictions are spliced to form meta training

set $\zeta_{meta} = (\zeta_1, \zeta_2, ..., \zeta_q)$, where $\zeta q = (z_{q1}, z_{q2}, ..., z_{qk})$. Notice, ζ_{meta} is a 2-D tensor forming 2nd order square meta feature matrix represented below,
$\zeta_{meta} = \sum_{t=1}^{k} \sum_{i=1}^{q} z_{it}$
$= \sum_{t=1}^{k} (z_{1t}, z_{2t}, ..., z_{qt})$

$$\zeta_{meta} = \begin{bmatrix} z_{11} & z_{21} & \dots & z_{q1} \\ z_{12} & z_{22} & \dots & z_{q2} \\ \vdots & \vdots & & \vdots \\ z_{1k} & z_{2k} & \dots & z_{qk} \end{bmatrix}$$

In order to include the impact of each kth prediction, averaging operation is performed and meta testing set $T_{meta} = T_1, T_2, ..., T_q$, where $T_q = T_{q1}, T_{q2}, ..., T_{q_p}$ is obtained. As before, 2-D tensor T_{meta} can be expressed as,

$$T_{meta} = \begin{bmatrix} T_{11} & T_{21} & \dots & T_{q1} \\ T_{12} & T_{22} & \dots & T_{q2} \\ \vdots & \vdots & & \vdots \\ T_{1k} & T_{2k} & \dots & T_{qk} \end{bmatrix} \tag{3.1}$$

3.2.2.3 Level 1 Final Prediction :

In the final step, meta classifier is trained over ζ_{meta} and tested on T_{meta} to predict the final output of trend. Moreover, in general Level 0 combines diverse ML tools (QDA, XGB, EXT, NB) to effectively reduce over-fitting and biasedness, a simple meta level 1 classifier L (AdaBoost) is employed to get train on the super-features of level 0 training set. This leads to generate final predictions ζ_{up} or ζ_{down} which matches the ground truth. Moreover, the mathematical expression shown below [50] represents the γ_i weighted sum of Level 0 predictions to get the final prediction $\hat{y}(x)$. These weights are assigned based on accurate base predictions. In stacking, these Level 0 learners must have a low correlation i.e. possessing different working methodologies. This provides liberty to Level 1 learner to extract from the diverse analysis of each learner to improve accuracy score.
$\hat{y}(x) = \sum_{i=1}^{m} \gamma_i h_i(x)$
Algorithm of stacking model is given below in algorithm 6 and the process of stacking model is depicted in figure 3.2,

3.2.3 Voting Ensemble Architecture (VEA):

This ensemble method works on the fact that the “majority is authority” for classification tasks. This method generates its final prediction by ensembling the predictions of

Algorithm 6 Stack integrated classifier Algorithm [52]

Input: Training data $D = (x_i, y_i)$ $(x_i \in \mathbb{R}^n, y_i \in Y)$
Level 0 classifiers $L_1, L_2, ..., L_P$
Level 1 classifier L
Output: Trend Prediction

1: **Data processing**
2: **Step 1** Base-learning
3: **for** p$\leftarrow$ 1 to P **do**
4: Learn base learner h_p on D_{train}, $h_p = L_p(D_{train})$
5: **end for**
6: **Step 2** Construct new set, $\zeta_{meta} = \phi$
7: **for** i$\leftarrow$ 1 to n **do**
8: **for** p$\leftarrow$ 1 to P **do**
9: $\zeta_p = L_P^{(k)}(x_n)$
10: **end for**
11: $\zeta_{meta} = \zeta_{meta} \cup (\zeta_1, \zeta_2, ..., \zeta_p)$
12: **end for**
13: $\hat{h} = L(\zeta_{meta})$
14: **Output :** $\zeta = \hat{h}(T_{meta})$

conceptually variant base learners based on "majority votes" on each data point [53].

This model is simple to implement and works faster. It simply generates output in favor of the max-classified class. Empirically, this method outperforms when significantly diverse base learners are grouped in a linear combination. Moreover, its prediction accuracy is superior to individual classifier performance. However, its computational cost is more expensive since it takes all of the hyper-parameters of each model into account. For generalized predictions, random and independent selection of base learners is preferred. So that each base learner gets trained over random subsets of data with its working mechanism to get generalized votes.

Symbolically, the Input training set containing both dependent and independent data $D_{train} = \{x_{train}, y_{train}\}$ is fitted to these four base learners as $C_1.fit(x_{train}, y_{train})$, $C_2.fit(x_{train}, y_{train}), C_3.fit(x_{train}, y_{train}), C_4.fit(x_{train}, y_{train})$ respectively. Their outcomes $\hat{y}_{m_1}, \hat{y}_{m_2}, \hat{y}_{m_3}, \hat{y}_{m_4}$ are obtained over x_{test}, y_{test} (set containing independent (x) and target (y) variables).

Then, the final output is obtained based on the maximum votes. In the case of equal votes for both classes, a tie-breaker rule strategy is implemented to select the final prediction. It states that in such cases choose the final answer between a high number of votes or a high probability estimate for both classes. Moreover, this decision should be taken carefully considering multiple facts.

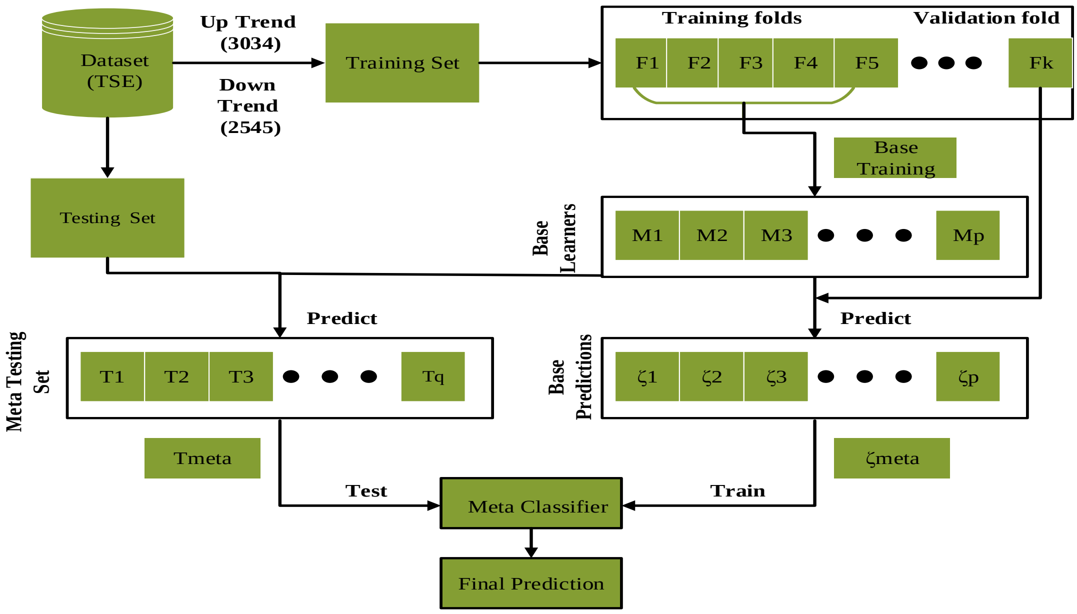

FIGURE 3.2: The architecture of stack model

Algorithm 7 Voting integrated classifier Algorithm [54]

Input: Training data $D = (x_i, y_i)$ $(x_i \in \mathbb{R}^n, y_i \in Y), Y \in (0, 1)$
Classification models $C_1, C_2, ..., C_P$
Output: Trend Prediction

Data Splitting
$D = \{D_{train}, D_{test}\}$
for p← 1 to P **do**
Train classifiers on D_{train} to get h_p, $h_p = C_p(D_{train})$
end for
for p← 1 to P **do**
Take predictions of h_p over D_{test}, $V_p = h_p(D_{test})$
end for
Sum of votes for each class $V_f = \sum_{i=1}^{p} V_i$
Final result

The algorithm 7 shows the working methodology of voting architecture and is pictorially depicted in figure 3.3.

Further in this research proposal, two kinds of voting are performed, hard and soft voting, described below as;

3.2.3.1 Hard Voting Classifier (HV):

HV classifier is a wrapper of set of predictions of base learners and selects the final one by taking mode of all [54]. That is, final decision is acknowledged in the favor of frequently

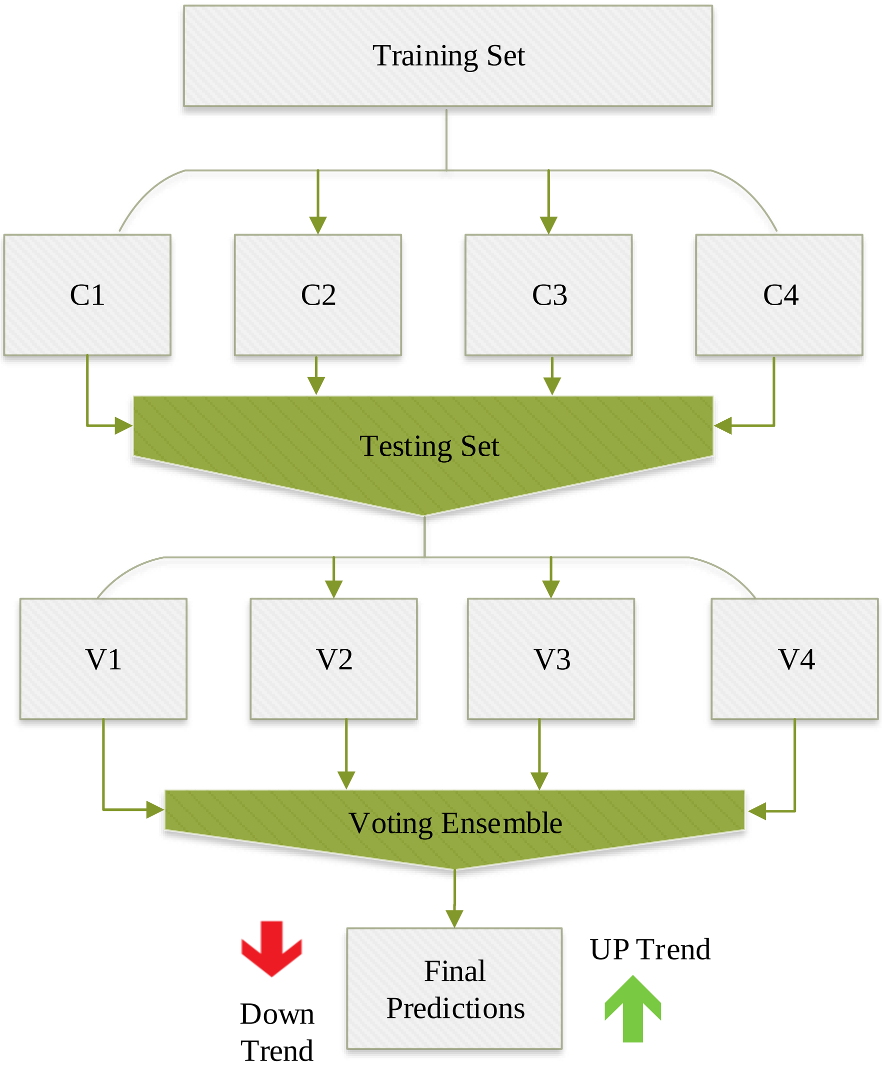

FIGURE 3.3: Voting Methodology

occurring class. $Eq.1$ shows the final class label $\hat{y}$ based upon plurality of each learner M_1, M_2, M_3, M_4 is given as,
$\hat{y} = \text{mode}\ \{M_1(x), M_2(x), M_3(x), M_4(x)\}$ (1)
where, $M_i(x)$ shows upward or downward trend for binary classification of stock signal. Moreover, HV is applicable for usage when even (not odd) number of classifiers are included [55]. Let's take an example in our scenario, we need to solve binary classification stock trend prediction problem using these five classifiers (QDA, EXT, NB, XGB, ADA). If three of them vote for class 1 label while remaining two vote for class 0 label, then ensemble classifier decision is class 1 (upward trend) with $\frac{3}{5}$ of the votes.

3.2.3.2 Soft Voting Classifier (SV) :

In this branch of voting, final prediction is done on the basis of highest total prediction probability of a class. It entails combining every classifier's prediction probability. This difference makes it different from hard voting. Equation below shows how probabilities of each class label can be calculated;

$\hat{y} = \sum_{i=1}^{n} P_i(M_i(x))$ (2)

where, $M_i(x)$ shows the outcome and $P_i(M_i(x))$ shows the probability of each class. Further, in our case, n=2 (Class 0 or 1). Again, lets consider an example in the same scenario, classifier 1 predicts 80% for class 1, classifier 2 predicts 50% chance, classifier 3 predicts 65%, classifier 4 gives 49% and classifier 5 gives 30% chances. Then, average of these probabilities comes out to be 54.8% for class 1 which is higher than 45.2% chances for class 0. Hence, soft classifier labelled as class 1 (upward trend) respectively.

3.2.4 Blending Ensemble Architecture (BEA) :

Blending architecture is closely allied with stacking architecture. The only difference between both ensemble methods is that stacking outcome is combination of predictions of k-fold trained base learners whilst blending perform predictions over holdout validation set rather on out-of-fold set [56]. In blending, dataset is splitted into three sets; training, hold-out and testing set. Training set is utilized by base learners for training and then predictions are obtained over hold-out set. Then, blend model gets trained over holdout predictions to generate one set of blend predictions. Also, another set of base predictions are obtained on testing data using training set. Then, both predictions set are evaluated to calculate the accuracy. Its working mechanism is shown in algorithm 8 shown below for further clarification [57].

Algorithm 8 Blend integrated classifier Algorithm [57]

Input: Training data $D = (x_i, y_i)$ $(x_i \in \mathbb{R}^n, y_i \in Y, Y \in (0,1))$

Level 0 classifiers $L_1, L_2, ..., L_P$

Level 1 classifier L

Output: Trend Prediction

Data Splitting

$D = \{D_{train}, D_{holdout}, D_{test}\}$

Step 1 Base-learning

for p← 1 to P **do**

Learn base learner h_p on D_{train}, $h_p = L_p(D_{train})$

end for

Step 2 Hold-out predictions

$S_p = h_p.(D_{holdout})$

Construct 2-D array using stored S_p

Step 3 Construct new blend model 'B' and train on S_p

for p← 1 to P **do**

$B_t = B_p.(S_p)$

end for

Make predictions of blend classifier on D_{test}

$\hat{B} = B_t.(D_{test})$

end for

Step 5 Make predictions of base classifiers on D_{test}

for p← 1 to P **do**

$\hat{L} = L_p.(D_{test})$

end for

Step 6 Construct 2-D array using stored $\hat{L}$

Evaluate predictions, $\epsilon = diff(\hat{L}, \hat{B})$

where ϵ is the error.

Moreover, the process of blending classifier is depicted in figure 3.4,

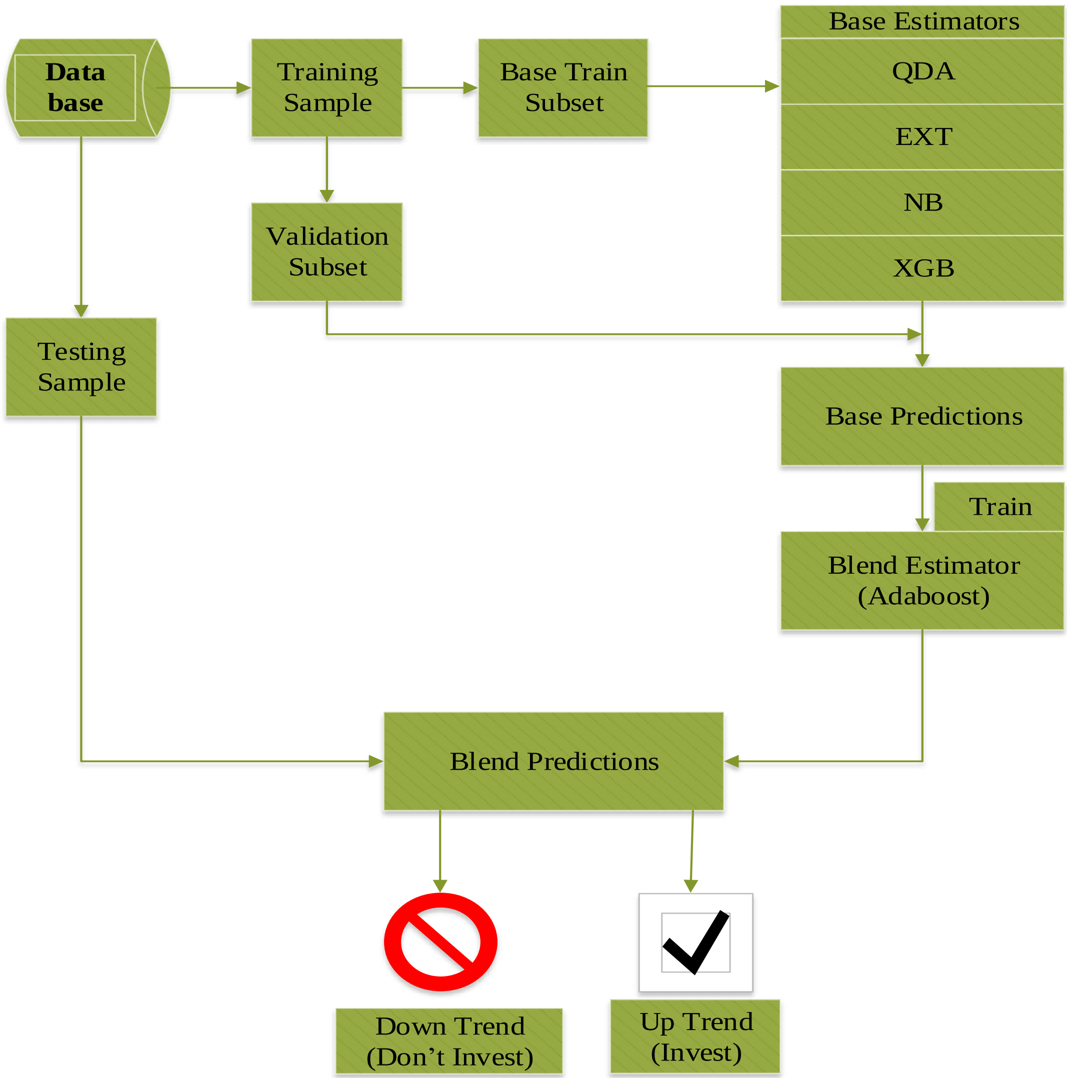

FIGURE 3.4: Blending Process

3.2.5 Comparison of ensemble methods

Table 3.1 represents the comparative analysis of working strategies of proposed ensemble methods. It is divided into three portions based upon base and meta level training procedure and outcomes of meta classifier. Further, this table draws clear segregation between these overlapping concepts.

TABLE 3.1: Comparision of Ensemble Architectures

Features	**Stacking**	**Voting**	**Blending**
A) Base classifiers training	It uses whole dataset 'D' for training in k-folds. Each of k^{th} fold is used for taking predictions once. Then 'k' group of predictions from M_i base learners are averaged together.	It randomly selects subsets from data for training and predictions take place on the basis of max votes or max probabilities.	It splits data into three portions, train, holdout and test. Base learners learn/train only at training portion of data. Then, it makes predictions on holdout set.
B) Meta classifier training	Meta learner gets train on averaged k-fold base predictions.	Meta learner gets train on individual base predictions.	Meta learner gets train on holdout validation set predictions.
C) Meta classifier final prediction	Meta classifier performs predictions on test set for final output.	Meta classifier performs predictions on test set and generate final output using max-voting or max-likely occurring technique.	Meta classifier performs predictions taking base predictions (on test set) and test set as input.

3.3 Exploratory Stock Data Analysis :

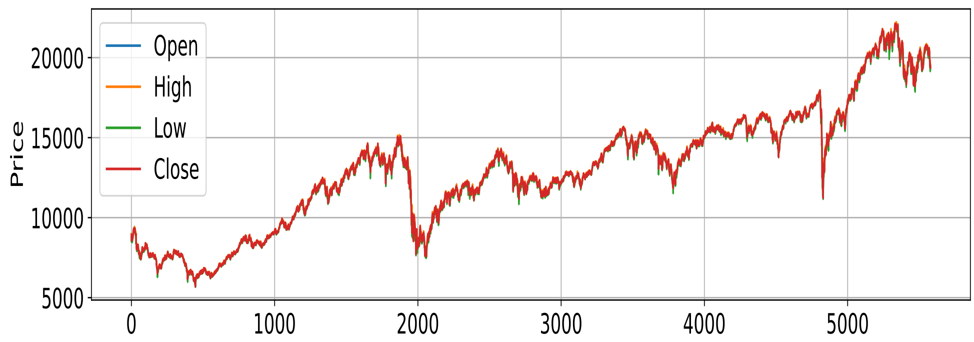

FIGURE 3.5: 2001.1.2 - 2023.3.17 TSE Index

This research proposal explores Toronto, which is the 2nd largest economic hub of North America and competitive business center of Canada. Figure 3.5 shows the dataset taken from the Toronto stock exchange (TSE) from January 2001 to March 2023. It consisted of 5579 days stock data and 7 features. These are Date, Open, Close, Adjusted Close, High, Low, and Volume. The dataset is prepared for the binary classification of the stock signal's trend into an Upward (class=1) trend and a Downward (class=0) trend. Based on these predictions, Investors/ market stakeholders decide whether to buy or sell the stock share. The target column is created using the 'Close' attribute by comparing the two consecutive values iteratively. If the closing price of (n+1)th entry is higher than (n)th one target column is filled with 1 else 0 is assigned. Table 3.2 describes the features of data:

Table 3.2: Features of Dataset

Features	**Values**
Date	It shows specific dates on which data is taken.
Open	It shows starting price of the stock in a day.
Close	It shows the closing price of the stock in a day.
AdjClose	It shows the price after subtracting the dividends per stock from it.
High	It shows the highest value of stock price in a day.
Low	It shows the lowest value of stock in a day.
Volume	It shows the measure of liquidity of the market.

Table 3.3 shows a detailed summary of the dataset.

Table 3.3: Statistics of TSE

Size	5579
No. of Classes	2
No. Percentage of Majority class	54.3%
Percentage of Minority class	45.6%
No. of instances in Training data	3905
No. of instances in Testing data	1674

3.3.1 Feature Importance using Random Forest Classifier (RF):

Data is pre-processed by employing a feature selection technique to find out the dataset's most and least important features. It varies from data to data and is highly dependent upon the problem at hand. It provides insight into data along with improving the classification by the model. We apply RF for selecting optimal features for the binary classification of stock data using the sklearn library in Python. RF does its job by aggregating multiple decision trees. It calculates the Gini index of each variable and ranks their importance score accordingly. Its value is between 0 (perfectly pure node) and 1 (highly impure node). RF calculates the Gini index using the following formula [58]:

$$Giniindex(j) = 1 - \sum_{k} [p(k|j)^2] \tag{3.2}$$

where k shows classes in binary classification and $j \in 1, 2, \ldots, N$ shows the no. of nodes of the tree. $p(k|j)$ shows relative frequency of k^{th} class at j^{th} node. Values show that Open-Close and High-Low are more important features than Volume, see Table 3.4. Table

TABLE 3.4: Importance measure of features

Features	**Importance Score**
Open	0.2288
Close	0.1749
AdjClose	0.1791
High	0.1564
Low	0.1376
Volume	0.1231

3.4 is pictorially depicted in figure 3.6.

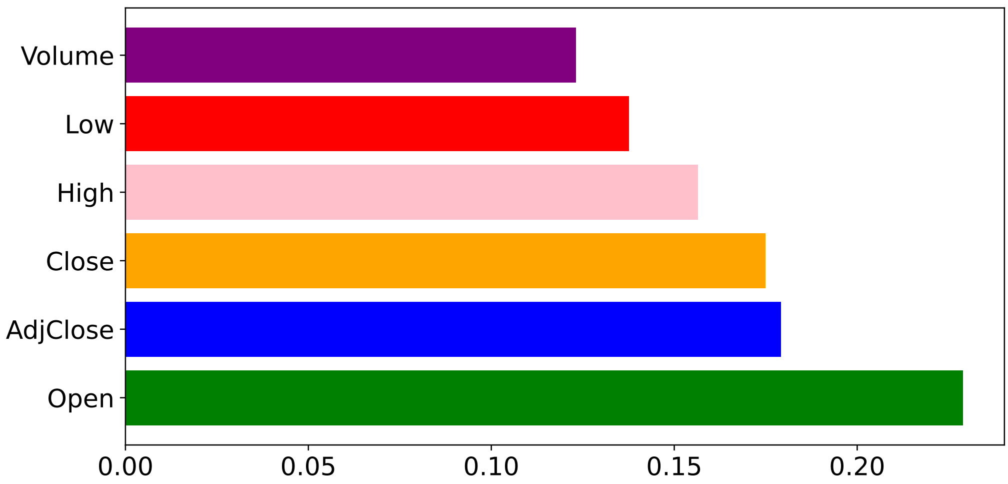

FIGURE 3.6: Features Importance of TSE Data

3.3.2 Class Distribution :

Input data is prepared and divided into two classes (class 0 and class 1) for binary classification of trend (see table 3.5). That is, whole data instances are splitted into two columns based upon target column.These class counts are depicted in the table and figure below;

Class imbalance is clearly evident in the figure 3.7. Class imbalance is root cause of biased and imbalance class predictions. That is, employed classifier gets train more over

Timeline	**No. of uptrend classes**	**No. of downtrend classes**
2001/2/1-2023/17/3	3034	2545

TABLE 3.5: Class distribution

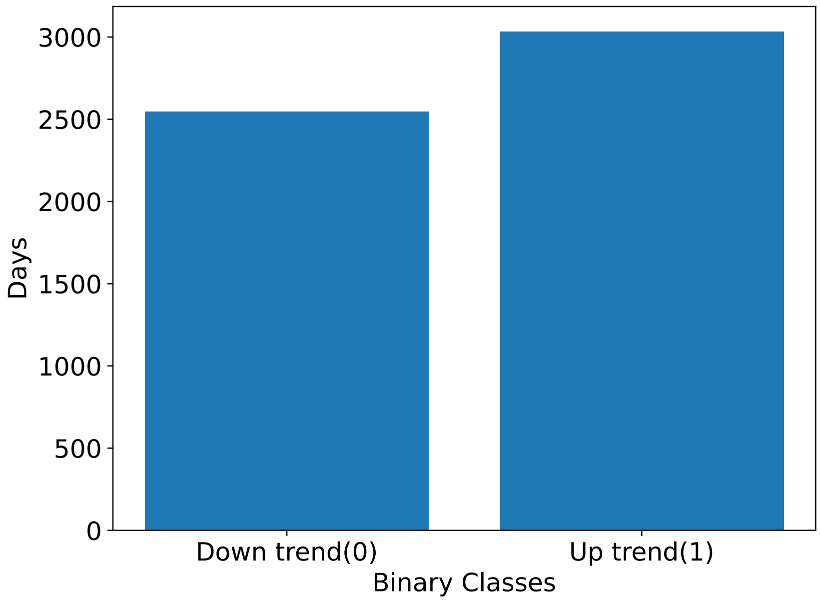

FIGURE 3.7: Class Distribution

frequently occurring class of input data (i.e upward class). Then, on testing stage it will generate predictions in the favor of majority (i.e upward) class.

Chapter 4

Proposed Evaluation Metrics

4.1 Evaluation Metrics for Trend Classification

Most of classification metrics are derived from confusion matrix. Its brief description is given below.

4.2 Confusion Matrix (CM)

Confusion matrix, also named as error matrix, is a combination of true or false positive and negative classes. It is a square matrix composed of true positive and true negative entries in leading diagonal while false positive and false negative entries in counter diagonal. These four quadrants are defined as follows:

True Positive (TP): is an outcome which shows how much model has truly or accurately predicted positive class. If the actual trend of stock signal is 1 and model classification outcome is also 1, then number of such entities are recorded as 'TP'.

True Negative (TN): is an outcome which shows how much model has truly or accurately predicted negative class. If the actual trend of stock signal is 0 and model classification outcome is also 0, then number of such entities are recorded as 'TN'.

False Positive (FP): is an outcome which shows how much model has falsely or incorrectly predicted positive class. If the actual trend of stock signal is 0 and model classification outcome is also 1, then number of such entities are recorded as 'FP'.

False Negative (FN): is an outcome which shows how much model has falsely or incorrectly predicted negative class. If the actual trend of stock signal is 1 and model classification outcome is also 0, then number of such entities are recorded as 'FN'.

Mathematically,

$$\text{Confusion Matrix} = \begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix}$$

These four counts are elaborated in table 4.1 for binary trend by considering positive class (Class 1) and negative class (Class 0) and pictorially represented in matrix form in figure 4.1. This three dimensional ensemble study is evaluated by the following classification indicators described below in terms of combinations of different entries of confusion matrix, some with probability and similarity concept for their versatile performance and robustness check.

TABLE 4.1: Equivalent Array Representation of Confusion Matrix for Binary Classification

Binary Classes	**Actual Positive Class(1)**	**Actual Negative Class(0)**
Predicted Positive Class(1)	TP	FP
Predicted Negative Class(0)	FN	TN

4.2.1 Accuracy

It is the ratio of sum of true positive and true negative to the total number of combinations of confusion matrix. It is preferable to use when classes are equally balanced. Mathematically, from [59],
$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
It quantifies misclassified errors in percentage.

4.2.2 Precision

It reflects how the model performs when predicting the true positives of all the positive class samples [59].
$Precision = \frac{TP}{TP+FP}$

4.2.3 Recall

It measures how much positive samples are correctly classified by the model among all possible outcomes [59].
$Recall = \frac{TP}{TP+FN}$

4.2.4 Macro-Average Recall

It is the arithmetic mean of recall of each class. This variant of recall is suitable for use in case of class imbalance problems. It assigns equal weights to all classes. Further, in case of binary classification, it is important to give equal importance while detecting the right class. Mathematically,
Macro-average recall $= \sum_{i=1}^{n} \frac{TP_i}{TP_i+FN_i}$

4.2.5 F1 score

It is harmonic mean of precision and recall [59]. It is classification metric used to check performance accuracy of model for binary classes. It is more suitable than other metrics for the case of imbalanced datasets to measure incorrectly classified samples. Its value ranges from 1 (best) to 0 (worst). Further, it can be extended for multi-class classification.
$F1score = 2 * \frac{Precision*Recall}{Precision+Recall}$, it can also be written as [60],
$F1score = \frac{(Precision^{-1}+Recall^{-1})^{-1}}{2}$

4.2.6 Matthews's Correlation Coefficient

It is the ratio of combination of all four entities of confusion matrix. Usually, it doesn't performs well in case of sparse and unstructured data. From [61],
$MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

4.2.7 Sensitivity

It tells us how much the model has truly predicted positive class (Uptrend). It is the ratio of true positives with the total number of positively predicted classes.
Mathematically, it is given as,
$sensitivity = \frac{TP}{TP+FN}$

4.2.8 Specificity

It tells us how much the model has truly predicted negative classes (Downtrend). It is the ratio of true negatives with the total number of negatively predicted classes. Mathematically,
$specificity = \frac{TN}{TN+FP}$

4.2.9 Cohen's kappa

It is a statistical measure used to quantify the mutual agreement between observed and actual probabilities of classes of categorical data. Its values lie between -1 to 1. The positive extreme of this range represents perfect agreement while the negative shows complete disagreement. Mathematically,
$kappa = \frac{p_o - p_e}{1 - p_e}$
where, $p_o = \frac{TP+TN}{N}$
and $p_e = \frac{\sum_{p=1}^{C}([\sum_{q=1}^{C}\sum_{s=1}^{N} f(s,k)C(s,q)].[\sum_{q=1}^{C}\sum_{s=1}^{N} f(s,q)C(s,k)])}{N^2}$ [62]
where, N shows number of instances $f(s,q)$ shows actual and $C(s,q)$ shows predicted probability.

4.2.10 Hamming Loss

It tells how many irrelevant labels are predicted by the classifier. It is frequently used for the performance check of machine learning classifiers. As the name indicates, the higher the hamming loss value, the higher will be prediction error. This loss function is mathematically expressed as [63],
$L_H(x, h(y)) = \frac{1}{n}\sum_{i=1}^{n}[x_i \neq h_i(y)]$
Hamming loss is the sum of the conventional 0/1 Loss function (frequently used in binary classification). Zero/One loss function is defined as,
$L_s(x, h(y)) = [x \neq h(y)]$ It is discrete, discontinuous and non-differentiable function whose values fluctuates from 0 to 1 and viceversa. If any single label is missclassified its value is assigned as 1 otherwise 0. Moreover, it cannot be used for optimization (loss minimization purposes).

4.2.11 Jaccard Score

It measures the disimilarity between actual and predicted class labels by the employed ensemble techniques. Firstly, it puts 'test target' and 'predicted' columns into sets named as 'T' and 'U'. Then, check for each element of T that whether it matches with the corresponding element of U or not. These number of matches are calculated in $(T \cap U)$ and combined as $(T \cup U)$. Mathematically [64],
$J(T, U) = \frac{|T \cap U|}{|T \cup U|}$
Its values ranges from 0 to 1. Closer the value to 1, higher will be the similarity and lesser will be the prediction error. Here, are some metric properties of jaccard score:
$J(T, U) \geq 0$
$J(T, U) = 1 \iff T = U$
$J(T, U) = J(U, T)$

$J(T, S) \geq J(T, U) + J(U, S)$
Note, Jaccard score is a similarity metric not a distance measure thatswhy '$\geq$' is used.

4.2.12 Movement direction accuracy (MDA)

This metric is used to verify the actual stock signal direction with the predicted signal's direction (up or down) throughout the input instances. Mathematically [60],
$MDA = \frac{\text{no. of accurately predicted movements}}{\text{total no of predicted movements}}$

4.2.13 Area Under Receiver Operating Characteristic Curve

It is a classification metric which calculates the area under the ROC curve formed between the true positive rate and false positive rate at the threshold. It distinguishes between classes and noise.

4.2.14 Precision recall Area Under the Curve

It provides summary of behavior of proposed model in case of rare events. Its value is one in case of perfect classification. Moreover, it is mostly applicable for unbalanced binary responses.

4.2.15 Execution Time

The time taken by a model to perform classification or regression tasks over large datasets is considered as execution time. It is affected by model complexity and data impurity.

Chapter 5

Simulation Results and Performance Analysis of Ensemble Methods

5.1 Case Study Setup

To evaluate the performance of our proposed framework, Google Colaboratory, an online platform is used with packages of sklearn library, according to system model illustrated in chapter 3.

5.2 Simulation Results

Table 5.1 shows the values of proposed performance metrics for all three ensemble architectures. It is evident that stacking ensemble has highest accuracy score, outperforming in 11 performance scores. Classification metrics (Accuracy, Precision, Recall, Macro-average Recall, F1 Score, mcc), validation metrics (sensitivity, specificity, Movement direction accuracy), loss metric (hamming loss), area metric (auc), probability metric (cohen's kappa) and similarity metric (jaccard score) are analyzed and calculated in this study.

Moreover, for clear comparative visualization of results of table 5.1, classification performance metrics are depicted in bar chart shown below.

TABLE 5.1: Performance Metrics of Ensemble Classifiers

Performance Metrics	**Stacking**	**Voting Soft**	**Voting Hard**	**Blending**
Accuracy	**0.85**	0.81	0.80	0.82
Precision	**0.86**	0.80	0.81	0.78
Recall	0.86	0.87	0.81	0.92
Macro-average Recall	0.84	0.80	0.79	0.92
F1 Score	**0.86**	0.84	0.81	0.84
mcc	**0.69**	0.62	0.59	0.64
kappa	**0.69**	0.62	0.59	0.63
auc	**0.84**	0.80	0.79	0.81
Specificity	**0.82**	0.73	0.77	0.70
Sensitivity	0.86	0.87	0.81	0.91
Hamming loss	**0.15**	0.18	0.20	0.18
Jaccard score	**0.75**	0.71	0.68	0.27
MDA (%)	**0.73**	0.67	0.65	0.68
Execution time (sec)	**13.41**	42.72	45.39	30.17

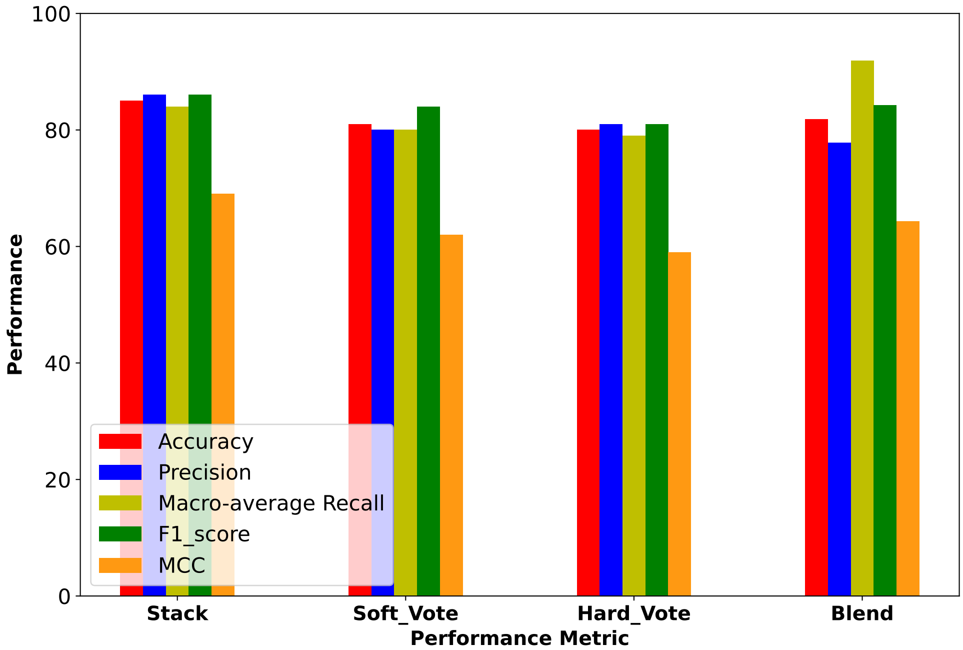

FIGURE 5.1: Classification metrics of ensemble methods

5.3 Reasoning

5.3.1 Which count of a confusion matrix is dangerous and which is beneficial?

TP (Gain in profit):
It tells us how accurately the model has predicted the stock signal's positive class/ up trend. A higher value of 'TP' prevents loss. Its higher value will increase the probability of buying the stock. Rather, its lower value will increase the probability of selling stock. Moreover, an increase in 'TP' is directly proportional to direct profit to the market.
TN (Saves from loss):
It tells how much the model has truly predicted a downtrend. A higher value will benefit investors in an alternative way by selling the decreasing stocks. This decision indirectly saves the market from facing loss.
FN (Red zone):
It tells us how accurately the model has falsely predicted the stock signal's negative class/ down trend. A higher value of 'FN' will lead to disastrous predictions which cause 'huge losses'. Lowering the value of 'FN' as much as possible interprets good generalizability of the classifier with minimal loss.
FP (Wrong selection of stocks):
It tells us how much the model has falsely predicted the upward trend of the stock signal. The trend was about to go down with decreasing stock prices. In such a case, an investor will buy the stock keeping an increasing trend in view. This will lead to the wrong choice of stocks. Therefore, it should also be minimized.

5.3.2 Confusion Matrix Comparative Analysis

The confusion matrix is a clear depiction of the performance of ensemble techniques in classifying target variables. Out of four proposed ensemble techniques, stack ensemble has the highest 'TN'= '627' (truly predicted downward trend) and lowest 'FP'='130' (falsely predicted upward trend) value. While soft voting achieves the highest 'TP'= '795' (truly predicted upward trend) value with the highest 'FP' = '200' (falsely predicted upward trend) error count. Therefore, we cannot select such a method having both extremes of accuracy and error clearly showing its inadequacy. Further, it is evident from Figure 5.2, that hard voting has higher 'FN' = '165' value (red zone). That's why, it has underperformed in most of the performance metrics showing its poor performance overall. The blend ensemble model is at the lowest extreme of the 'FN' = '49' value among all proposed ensembles. This is the main reason behind its highest values of 'recall', 'macro-average recall', and 'sensitivity' metrics among all of the four architectures. Moreover, confusion matrices of all proposed methods can be depicted below in Figures 5.2, 5.3, 5.4, and 5.5.

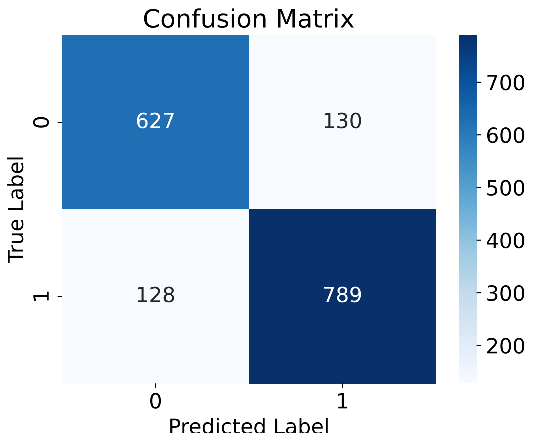

FIGURE 5.2: Stack Error Matrix

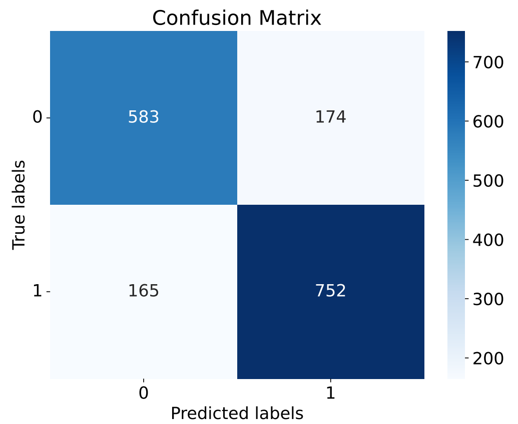

FIGURE 5.3: Vote-Hard Error Matrix

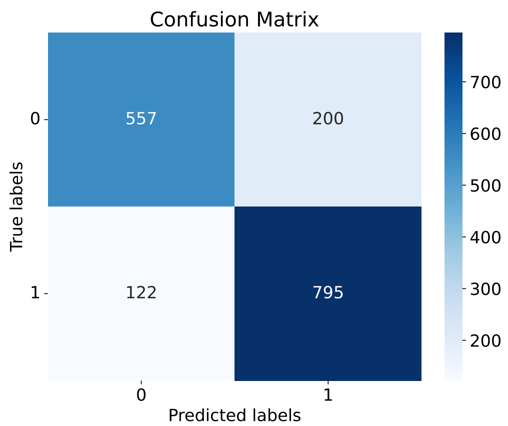

FIGURE 5.4: Vote-Soft Error Matrix

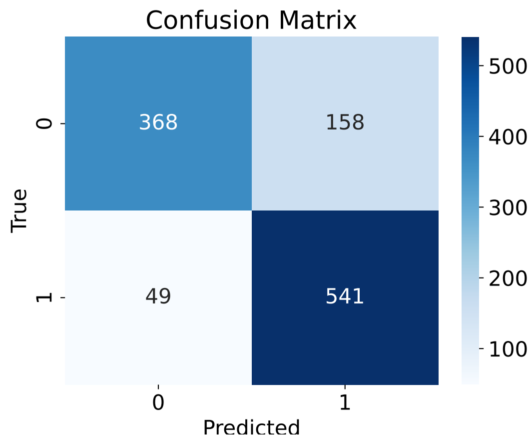

FIGURE 5.5: Blend Error Matrix

5.3.3 Sensitivity, Specificity and Accuracy Comparative Analysis:

This 3-D combination of performance analysis is selected to capture the varying behavior of proposed techniques undercover of error matrix scores. Firstly, lets built an analogy between accuracy, specificity and sensitivity performance metrics with confusion matrix. As, explained in chapter 4 we can see that specificity is directly proportional with 'TN' rate while inversely proportional to 'FP' rate. Further, sensitivity increases with the increase in 'TP' value and decreases by lowering the value of 'FN' as much as possible.

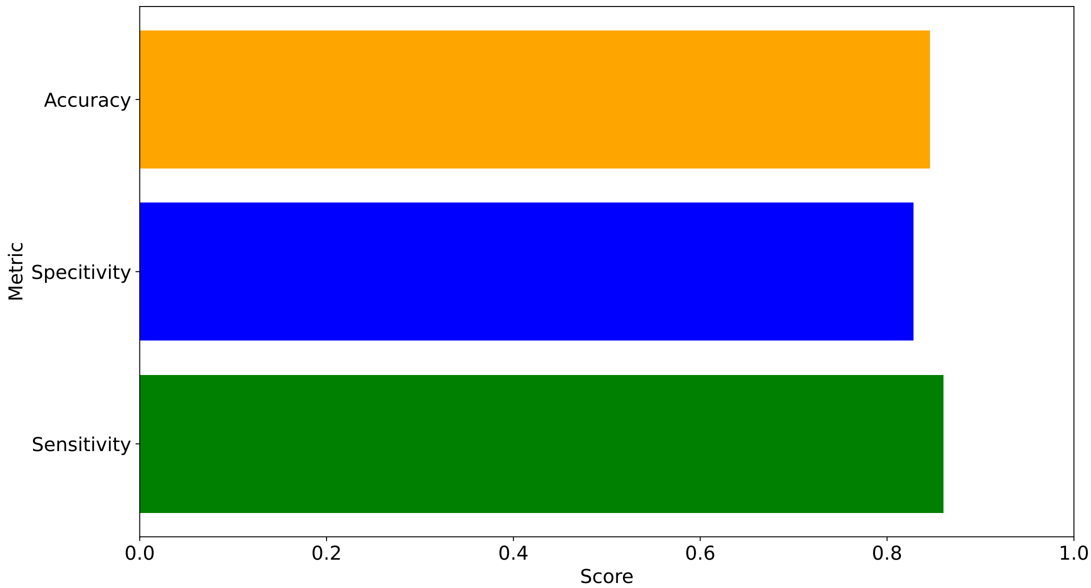

FIGURE 5.6: Stack performance

Accuracy is computational combination of all counts of confusion matrix. Moreover, it is balancing curve between specificity and sensitivity curve. As, sensitivity and specificity both are inversely proportional to each other, their curves cut each other in a cross section. Accuracy curve occurs in between these two passing from their cut points [65]. That's the reason their comparative analysis chosen to be part of this extensive performance analysis.

Stack ensemble has highest value of 'TN' with lowest value of 'FP' attains highest value of specificity. While blend ensemble has lowest value of 'FN' highest bar of sensitivity. In stock market context, higher specificity will lead investors to buy more profitable stocks while higher value of sensitivity will indicate loss prediction strengthening countries economy. Here, comes accuracy to play it part to overcome this balance by providing clear cut idea overall. Because accuracy measure shows both truly predicted upward and downward trend predictions collectively. It combines the impact of both sensitivity and specificity in this comparison. Evidently, stacking ensemble has highest accuracy surpassing all of proposed combinations. Its reason lies in its 'two-way cross validation' working methodology.

While in hard and soft voting, both fails to beat in any of metric of this chosen combination. In stock market, it is quite necessary for a classifier to learn and get train over whole input data in order to capture its volatile non-linear patterns. However, this ensemble technique didn't allows classifiers to do so. It is because in voting both training

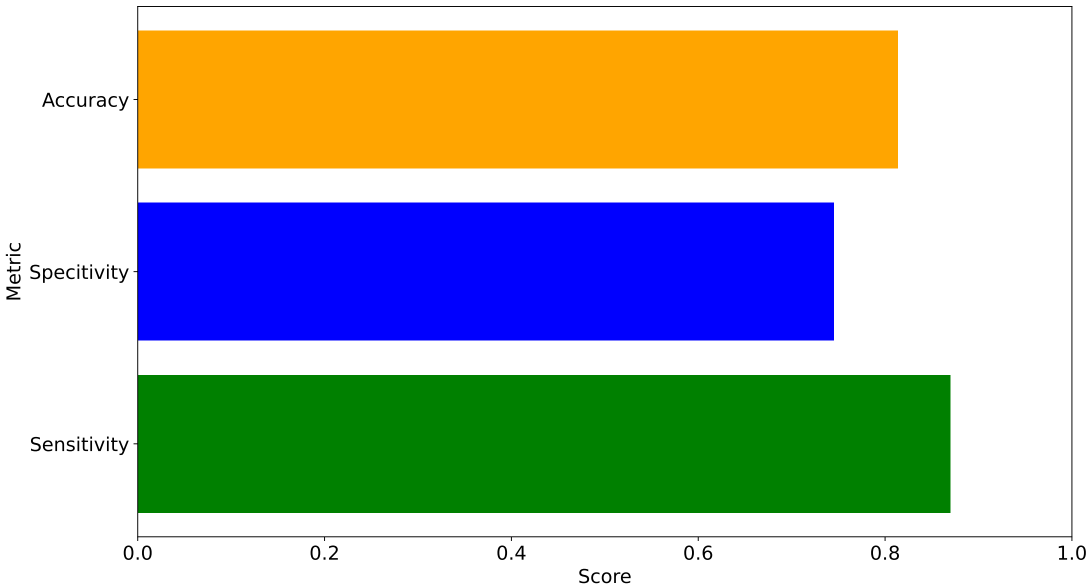

FIGURE 5.7: Soft voting performance

and testing set is selected on random basis once and remain fixed till end of procedure. Due to which, whole data couldnot be used for training and decision is taken by meta classifier on the basis of votes of partially learned base classifiers.

Blending ensemble, the third dimension of this ensemble study has transend stacking in 3 coherent metrics (sensitivity or recall and macro-average recall) out of 13 selected check metrics. On closer view, empirical values shows that blend method attains '12' units lesser specificity count while '5' units higher sensitivity than stacking method. That is, it performs slightly well in predicting 'up' trend while performs poorly in case of 'down' trend. That'swhy, their combined effect has lowered overall accuracy value with '3' units from stacking. This variation of blend results happen due to its out-of-fold validation by base learners. It provides meta learner to learn over more variant and relevant training data before making predictions.

Lastly, we will prefer an ensemble model which has less 'FN' and 'FP' value as much as possible.

5.3.4 Precision-Recall Comparative Analysis:

It represents the area under the precision-recall curve by taking precision on the vertical axis and recall on the horizontal axis. It is pertinent to mention that range of the PR curve lies from 0 to 1. Recall is the measure of relevancy of predicted outcomes while precision is the measure of relevant outcomes present in the set of predictions. From Table

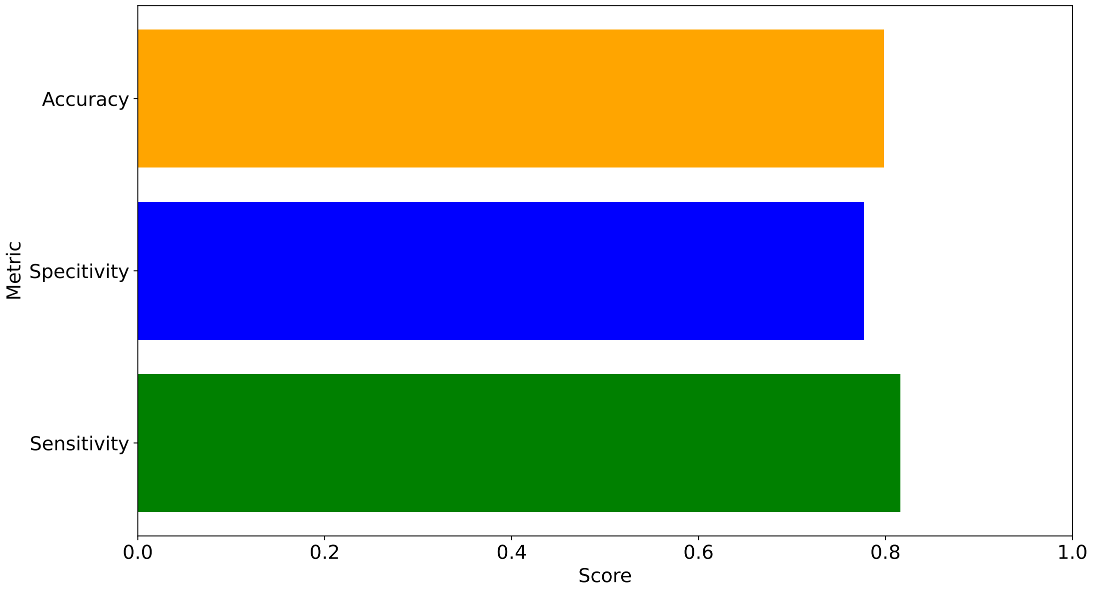

FIGURE 5.8: Hard voting performance

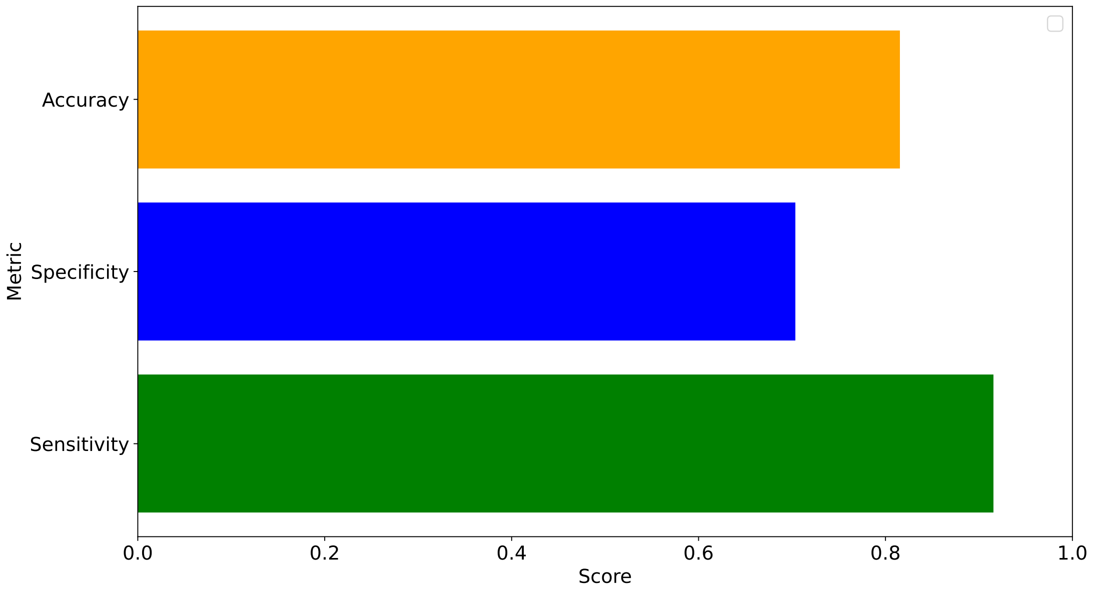

FIGURE 5.9: Blend performance

4.1, the Blending ensemble model has the highest recall count among all other ensemble methods due to its lowest 'FN' value. This shows the ratio of uptrend signals (1) that are misclassified as downtrend (0). Relying on this count will make investors sell their profitable stocks unknowingly. Another hidden fact behind this steep rise of recall lies in its architecture. That is, it trains its meta-classifier on validation predictions, which increases the relevancy of retrieved sampled data. Thereby, decreasing 'FN' and increase in recall.

Now, discussing another extreme 'Precision' of this 2-D graph. It is evident from Table 4.1, that the stack model has gained the highest precision with 0.86 units due to its lowest 'FP' value among all ensembles. While the blend ensemble has gained the lowest precision with 77.76 units which decreases the area under its PR curve. Further from Figure 5.9, the stack model PR curve is at the leading front showing its high generalization with the highest 85% accuracy score.

In our case study, accurate predictions of downtrends have more importance than uptrends. This is because beforehand 'loss' prediction will save investors from buying or retaining such stocks. Hence, 'TN' value analysis is the turning point of convergence toward choosing a highly robust ensemble method. In the particular stock scenario, the model with high specificity would be preferred over high sensitivity as a robust trend prediction model to be used by decision-makers.

Lastly, both variants of the voting method have almost similar trajectories of their PR curves, lying under their competitor's PR curves.

5.3.5 AUC ROC Comparative Analysis:

The area under the receiver operator curve is used to show the separability power of the classifier between classes when data is unevenly distributed. That is higher the value of AUC, the more relevant predictions could be made by the proposed classifier. Moreover, its value ranges from 0 to 1. The black diagonal line represents 0.5 or 50% AUC value, which is as good as selecting labels randomly portraying biased results. So, an AUC value of 0.5 or below represents underfitting, the model is unable to learn from the training set. Figure 5.11 depicts that all ensemble AUC curves are higher than this lower bound. Moreover, the AUC curve of the blend ensemble has a higher peak with a ROC area of 90.89. AUC-ROC is a graphical depiction of TPR (sensitivity) and FPR (specificity) on the vertical and horizontal axis, respectively. Mathematically, it is the integration operation of TPR (FPR) with respect to FPR taking 0 to 1 lower and upper limits.
Let's have AUC interpretation in terms of sample space and probability shades. Consider a sample space D given as,

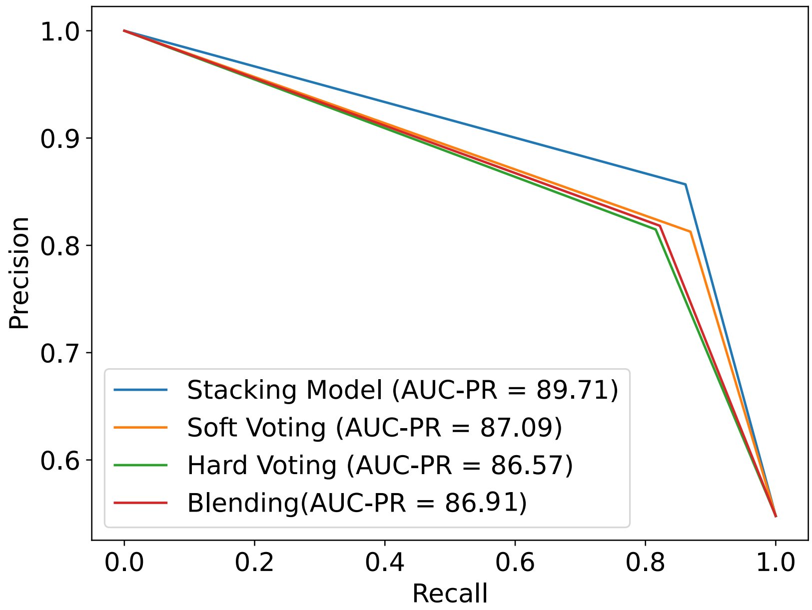

FIGURE 5.10: Precision-recall curves of ensemble methods

$D = X \times \{-1, +1\}$, where X represents the space of instances and P(x, c) represents the probability that the input example x belongs to class c ((+1) or (-1)). Let f is the employed classifier as a mapping function between input space X and range [0, 1]. Specifically, $f(x)$ represents the number of chances that the input instance belongs to (+1) or (-1). The AUC value of classifier 'f' taking randomly chosen sample 'S' as input can be calculated as [66],

$AUC(f, S) = \frac{1}{|p|.|n|} \sum_{x_i \in p} \sum_{x_j \in n} w(f(x_i) - f(x_j))$

where $S = \{p \cup n\}$ is a set of positive 'p' and negative 'n' classes. Fir the case of study, we have used the notation 0 for downtrend and 1 for uptrend. The modifier function $w(.)$ maps the values from $[-1, +1]$ to $[0, 1]$. It assign weights to the probability differences of positive (x_i) and negative (x_j) samples. Its piece-wise illustration is given as,

$$w(t) = \begin{cases} 1 & \text{if } t > 0 \\ 0.5 & \text{if } t = 0 \\ 0 & \text{if } t < 0 \end{cases}$$

The AUC value of classifier f shows that how much the probability of $f(x_i)$ is higher than $f(x_j)$, for randomly chosen x_i and x_j from sample space D. In an ideal case, value of AUC is 1 with 0 'FPR' and then along x the curve moves straight till positive limit of vertical axis. Further, if AUC values falls above the diagnol with minimal distance from

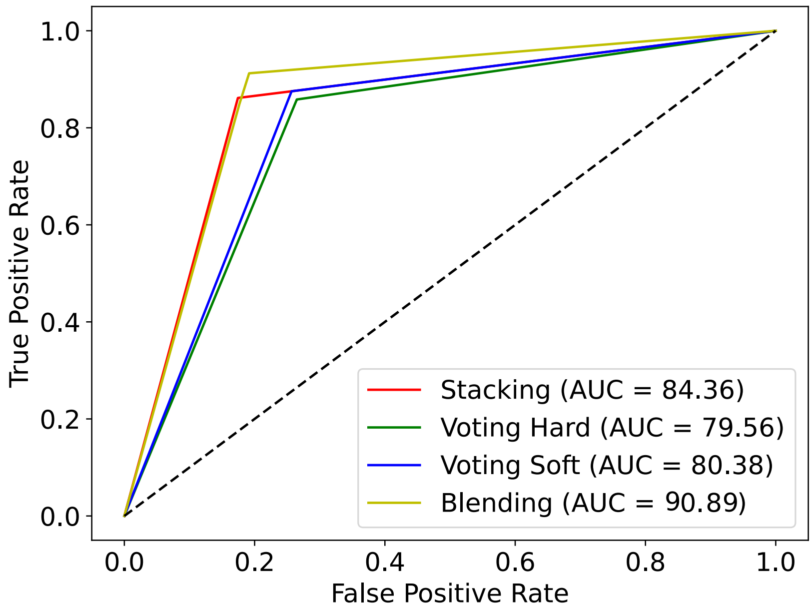

FIGURE 5.11: Auc-Roc curves of ensemble methods

1 (vertical extreme), depicts good discriminating power of classifier.

As shown in the figure 5.11, all of ensemble methods have performed well above the diagnol line. While, blend has highest discriminating power between the classes due to its higher relevancy recall score. In stock trend prediction, investors are highly concerned with their increase in profit, therefore they prefer the model which gives overall accurate results.

5.4 Individual Base and Meta Classifier Comparative Analysis:

From chapter 3, we recall that Ext, Qda, Naive bayes, Extreme gradient boosting are applied as base classifier and Adaboost as meta classifier. Table 5.2 shows the values of performance metrics in case of base and meta classifiers.

From figure 5.12, it is clearly evident that base classifier QDA outperforms than all of its individual competitors. EXT processes data to generate randomized trees of features to get optimum trend predictions. This randomization generates highly unbiased and generalized outcomes. However, this algorithm performs well in high dimensional data with uncorrelated features. While in this case study data, we have only 7 correlated

TABLE 5.2: Performance Metrics of Individual ML Classifiers

Performance Metrics	**QDA**	**EXT**	**NB**	**XGB**	**ADA**
Accuracy	**0.82**	0.69	0.53	0.75	0.56
Recall	**0.93**	0.70	0.77	0.79	0.87
Precision	**0.79**	0.72	0.55	0.77	0.57
auc	**0.81**	0.69	0.51	0.75	0.54
Macro-average Recall	**0.81**	0.69	0.51	0.75	0.54
F1 Score	**0.85**	0.71	0.64	0.78	0.69
mcc	**0.65**	0.37	0.01	0.50	0.10
kappa	**0.64**	0.37	0.01	0.50	0.08
Specificity	**0.69**	0.74	0.24	0.70	0.20
Sensitivity	**0.93**	0.72	0.77	0.79	0.87
Execution time (sec)	2.01	1.45	**1.06**	4.21	12.5

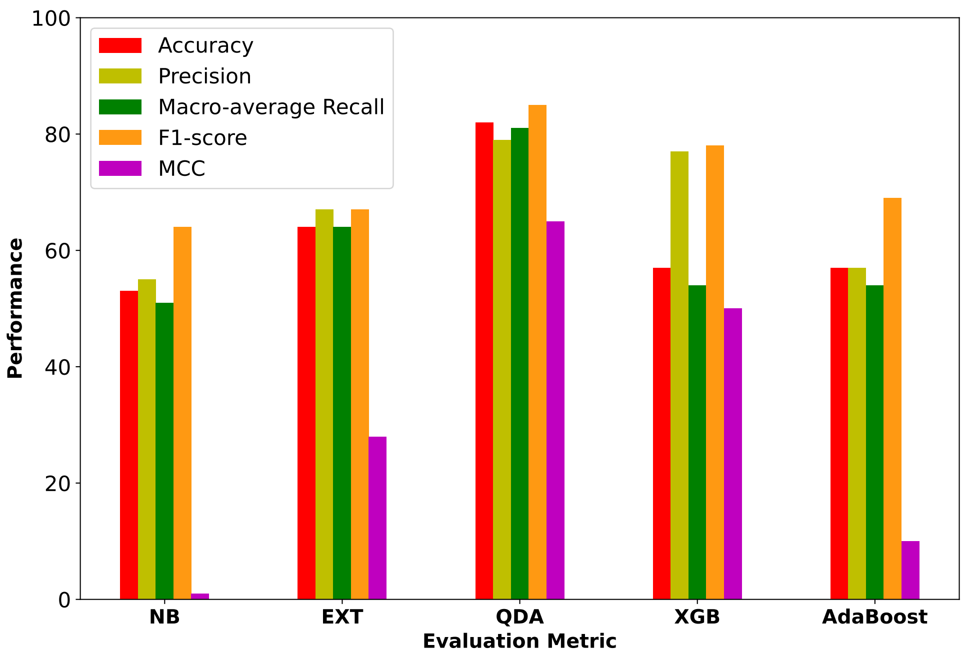

FIGURE 5.12: ML Classifiers individual performance

features that's why it couldn't standout. Moreover, its computational overhead rises with increase in dimensions.

Naive Bayes classifier calculates the posterior probability of each class and selects the one which has its higher value as final prediction. It performance increases when 'N' is very large. Thats the reason it couldn't performed well for 5579 instances. While, XGB optimizes its objective function by adding trees, which reduces the loss function. It performed much better than others because it controls the depth of trees and weights of leaf nodes itself.

AdaBoost is used as meta algorithm because its random feature selection at each iteration highly reduces overfitting problem. At this conjunction, it makes trees upon base predictions and combines the effect of each classifier equally for ensembling. However, it is highly affected by noisy and scattered data. Individually, it couldn't performed well due to less availability of input features.

QDA is non-linear base classifier. It generates 2-D decision boundary curves to separate the on-linear stock signal's data. That's the reason it has captured the non-linear dependencies of data so well. Moreover, this algorithm runs over bayes theorem to find posterior probability but it is exempted from independent features condition. This collectively played a major role for its overall robust performance. Moreover, performance metrics shows that it is efficient in predicting uptrend of stock signal's.

Lastly, to undo overfiiting and baisedness issues of individual classifiers, heterogeneous ensemble methods are applied for generalized and accurate results.

Chapter 6

Conclusion and future work

6.1 Conclusion

A comparative study is conducted to solve trend prediction problem of financial stock markets. The ensemble models used in the study are processed on TSE market data. This study highlighted pros and cons of these proposed methods under the shade of stock data. Empirical results have shown that the proposed stacking model has the highest accuracy of 85% with higher values for most performance metrics taken from literature. It also emphasized on the fact that stacking model learns the entire training data. That is why it performed so well on stock data. Moreover, its comparison with other two methods exhibit that the blending ensemble model has higher AUC value due to its lowest 'FN' value. Overall, the confusion matrix played a crucial role in determining the behavior of ensemble methods using multiple metrics. Further adding to the research contribution, behavior of employed base and meta classifier is analyzed individually, which figured out the efficacy of non-linear QDA classifier. In our study, accurate prediction of down trend (higher TN value) is found to be more crucial than up trend. Therefore, stacking model is preferred for further use in studies and by financial advisors to perform efficient trend prediction.

6.2 Future Work

- In future, proposed ensemble methods performance can be increased by adding technical features into the dimensions of dataset.
- An ensemble architecture can be built by taking ML classifiers on base level with a DL model as meta model.

- Comparative analysis could be done by implementing same proposed architecture on multiple stock datasets having same features and number of instances.
- Hybrid ML classifier or DL models could also be compared with these proposed ensemble techniques to figure out the effective one.
- Performance of these methods could be improved by employing different data pre-processing techniques.

Chapter 7

References

Bibliography

[1] Nevasalmi, L. (2022). Recession forecasting with high-dimensional data. Journal of Forecasting, 41(4), 752-764.

[2] Chacon, H. D., Kesici, E., & Najafirad, P. (2020). Improving financial ´ time series prediction accuracy using ensemble empirical mode decomposition and recurrent neural networks. IEEE Access, 8, 117133-117145.

[3] Zhong, X., & Enke, D. (2017). Forecasting daily stock market return using dimensionality reduction. Expert Systems with Applications, 67, 126-139.

[4] Zhao, Y., & Chen, Z. (2021). Forecasting stock price movement: New evidence from a novel hybrid deep learning model. Journal of Asian Business and Economic Studies.

[5] Haq, A. U., Zeb, A., Lei, Z., & Zhang, D. (2021). Forecasting daily stock trend using multi-filter feature selection and deep learning. Expert Systems with Applications, 168, 114444.

[6] Chandola, D., Mehta, A., Singh, S., Tikkiwal, V. A., & Agrawal, H. (2022). Forecasting Directional Movement of Stock Prices using Deep Learning. Annals of Data Science, 1-18.

[7] Lin, Y. F., Huang, T. M., Chung, W. H., & Ueng, Y. L. (2020). Forecasting fluctuations in the financial index using a recurrent neural network based on price features. IEEE Transactions on Emerging Topics in Computational Intelligence, 5(5), 780-791.

[8] Sun, J., Xiao, K., Liu, C., Zhou, W., & Xiong, H. (2019). Exploiting intra-day patterns for market shock prediction: A machine learning approach. Expert Systems with Applications, 127, 272-281.

[9] Ibrahim, A., Kashef, R., & Corrigan, L. (2021). Predicting market movement direction for bitcoin: A comparison of time series modeling methods. Computers & Electrical Engineering, 89, 106905.

[10] Srivastava, A. K., Srivastava, A., Singh, S., Sugandha, S., & Gupta, S. (2022). Design of Machine-Learning Classifier for Stock Market Prediction. SN Computer Science, 3(1), 1-11.

[11] Żbikowski, K., & Antosiuk, P. (2021). A machine learning, bias-free approach for predicting business success using Crunchbase data. Information Processing & Management, 58(4), 102555.

[12] Calainho, F. D., van de Minne, A. M., & Francke, M. K. (2022). A Machine Learning Approach to Price Indices: Applications in Commercial Real Estate. The Journal of Real Estate Finance and Economics, 1-30.

[13] Chen, W., Xu, H., Jia, L., & Gao, Y. (2021). Machine learning model for Bitcoin exchange rate prediction using economic and technology determinants. International Journal of Forecasting, 37(1), 28-43.

[14] Kabbani, T., & Duman, E. (2022). Deep reinforcement learning approach for trading automation in the stock market.

[15] Chang, V., Man, X., Xu, Q., & Hsu, C. H. (2021). Pairs trading on different portfolios based on machine learning. Expert Systems, 38(3), e12649.

[16] Akyildirim, E., Goncu, A., & Sensoy, A. (2021). Prediction of cryptocurrency returns using machine learning. Annals of Operations Research, 297(1), 3-36.

[17] Özgür, Ö., & Akkoç, U. (2021). Inflation forecasting in an emerging economy: selecting variables with machine learning algorithms. International Journal of Emerging Markets.

[18] Alsayed, A. R. (2022). Turkish Stock Market from Pandemic to Russian Invasion, Evidence from Developed Machine Learning Algorithm. Computational Economics, 1-17.

[19] Bukhari, A. H., Raja, M. A. Z., Sulaiman, M., Islam, S., Shoaib, M., & Kumam, P. (2020). Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting. Ieee Access, 8, 71326-71338.

[20] Leippold, M., Wang, Q., & Zhou, W. (2022). Machine learning in the Chinese stock market. Journal of Financial Economics, 145(2), 64-82.

[21] Nabipour, M., Nayyeri, P., Jabani, H., Shahab, S., & Mosavi, A. (2020). Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis. IEEE Access, 8, 150199-150212.

[22] Yuan, X., Yuan, J., Jiang, T., & Ain, Q. U. (2020). Integrated long-term stock selection models based on feature selection and machine learning algorithms for China stock market. IEEE Access, 8, 22672-22685.

[23] Nguyen, H. T., Tran, T. B., & Bui, P. H. (2021). An effective way for Taiwanese stock price prediction: Boosting the performance with machine learning techniques. Concurrency and Computation: Practice and Experience, e6437.

[24] Kong, A., Zhu, H., & Azencott, R. (2021). Predicting intraday jumps in stock prices using liquidity measures and technical indicators. Journal of Forecasting, 40(3), 416-438.

[25] Parray, I. R., Khurana, S. S., Kumar, M., & Altalbe, A. A. (2020). Time series data analysis of stock price movement using machine learning techniques. Soft Computing, 24(21), 16509-16517.

[26] Shah, A., Gor, M., Sagar, M., & Shah, M. (2022). A stock market trading framework based on deep learning architectures. Multimedia Tools and Applications, 81(10), 14153-14171.

[27] Ma, Y., Han, R., & Wang, W. (2021). Portfolio optimization with return prediction using deep learning and machine learning. Expert Systems with Applications, 165, 113973.

[28] Polamuri, S. R., Srinivas, K., & Mohan, A. K. (2022). Multi-model generative adversarial network hybrid prediction algorithm (MMGAN-HPA) for stock market prices prediction. Journal of King Saud University-Computer and Information Sciences, 34(9), 7433-7444.

[29] Yun, K. K., Yoon, S. W., & Won, D. (2021). Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process. Expert Systems with Applications, 186, 115716.

[30] Chen, W., Zhang, H., Mehlawat, M. K., & Jia, L. (2021). Mean–variance portfolio optimization using machine learning-based stock price prediction. Applied Soft Computing, 100, 106943.

[31] Alsulmi, M. (2021). Reducing Manual Effort to Label Stock Market Data by Applying a Metaheuristic Search: A Case Study From the Saudi Stock Market. IEEE Access, 9, 110493-110504.

[32] Naik, N., & Mohan, B. R. (2020). Intraday stock prediction based on deep neural network. National Academy Science Letters, 43(3), 241-246.

[33] Liu, Q., Tao, Z., Tse, Y., & Wang, C. (2022). Stock market prediction with deep learning: The case of China. Finance Research Letters, 46, 102209.

[34] Khattak, M. A., Ali, M., & Rizvi, S. A. R. (2021). Predicting the European stock market during COVID-19: A machine learning approach. MethodsX, 8, 101198.

[35] Rampini, L., & Cecconi, F. R. (2021). Artificial intelligence algorithms to predict Italian real estate market prices. Journal of Property Investment & Finance.

[36] Thakur, M., & Kumar, D. (2018). A hybrid financial trading support system using multi-category classifiers and random forest. Applied Soft Computing, 67, 337-349.

[37] Peng, Y., Albuquerque, P. H. M., Kimura, H., & Saavedra, C. A. P. B. (2021). Feature selection and deep neural networks for stock price direction forecasting using technical analysis indicators. Machine Learning with Applications, 5, 100060.

[38] Lahmiri, S., & Bekiros, S. (2021). Deep learning forecasting in cryptocurrency high-frequency trading. Cognitive Computation, 13(2), 485-487.

[39] Adegboye, A., & Kampouridis, M. (2021). Machine learning classification and regression models for predicting directional changes trend reversal in FX markets. Expert Systems with Applications, 173, 114645.

[40] Ayala, J., García-Torres, M., Noguera, J. L. V., Gómez-Vela, F., & Divina, F. (2021). Technical analysis strategy optimization using a machine learning approach in stock market indices. Knowledge-Based Systems, 225, 107119.

[41] Kunapuli, G. (2023). Ensemble Methods for Machine Learning. Simon and Schuster.

[42] Jindal, A., Dua, A., Kaur, K., Singh, M., Kumar, N., & Mishra, S. (2016). Decision tree and SVM-based data analytics for theft detection in smart grid. IEEE Transactions on Industrial Informatics, 12(3), 1005-1016.

[43] Differences between LDA, QDA and Gaussian Naive Bayes classifiers by Francesca Argenziano, Online Available: https://towardsdatascience.com/differences-of-lda-qda-and-gaussian-naive-bayes-classifiers-eaa4d1e999f6 (Last Access on: 7th May, 2023)

[44] Mahajan Shubhrata, D., Deshmukh Kaveri, V., Thite Pranit, R., Samel Bhavana, Y., & Chate, P. J. (2016). Stock market prediction and analysis using Naïve Bayes. International Journal on Recent and Innovation Trends in Computing and Communication, 4(11), 121-124.

[45] Gopal, M. (2019). Applied machine learning. McGraw-Hill Education.

[46] Udomsak, N. (2015). How do the naive Bayes classifier and the Support Vector Machine compare in their ability to forecast the Stock Exchange of Thailand?. arXiv preprint arXiv:1511.08987.

[47] Dey, S., Kumar, Y., Saha, S., & Basak, S. (2016). Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting. PESIT South Campus, 1-10.

[48] Xue, J. (2021, November). Prediction of NASDAQ Index Based on Stacking Method. In 2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE) (pp. 505-510). IEEE.

[49] Ou, P., & Wang, H. (2009). Prediction of stock market index movement by ten data mining techniques. Modern Applied Science, 3(12), 28-42.

[50] Gyamerah, S. A., Ngare, P., & Ikpe, D. (2019, May). On stock market movement prediction via stacking ensemble learning method. In 2019 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr) (pp. 1-8). IEEE.

[51] Ampomah, E. K., Qin, Z., Nyame, G., & Botchey, F. E. (2021). Stock market decision support modeling with tree-based AdaBoost ensemble machine learning models.

[52] Xia, R., Gao, Y., Zhu, Y., Gu, D., & Wang, J. (2022). An Efficient Method Combined Data-Driven for Detecting Electricity Theft with Stacking Structure Based on Grey Relation Analysis. Energies, 15(19), 7423.

[53] Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A comprehensive evaluation of ensemble learning for stock-market prediction. Journal of Big Data, 7(1), 1-40.

[54] Habib, A. Z. S. B., & Tasnim, T. (2020, December). An ensemble hard voting model for cardiovascular disease prediction. In 2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI) (pp. 1-6). IEEE.

[55] Peppes, N., Daskalakis, E., Alexakis, T., Adamopoulou, E., & Demestichas, K. (2021). Performance of machine learning-based multi-model voting ensemble methods for network threat detection in agriculture 4.0. Sensors, 21(22), 7475.

[56] Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A comprehensive evaluation of ensemble learning for stock-market prediction. Journal of Big Data, 7(1), 1-40.

[57] Hansrajh, A., Adeliyi, T. T., & Wing, J. (2021). Detection of online fake news using blending ensemble learning. Scientific Programming, 2021, 1-10.

[58] Liu, H., Zhou, M., Lu, X. S., & Yao, C. (2018, March). Weighted Gini index feature selection method for imbalanced data. In 2018 IEEE 15th international conference on networking, sensing and control (ICNSC) (pp. 1-6). IEEE.

[59] Understanding and Implementing Classification Model Evaluation Method (Accuracy, Precision, Recall and F-Score) in Python from scratch by Get unlimited access Gilbert Putra, Online Available: https://medium.com/@gilbertputra/understanding-and-implementing-classification-model-evaluation-method-accuracy-precision-recall-d5b611e714ba (Last Access on: 4th January, 2023)

[60] Li, Y., & Pan, Y. (2022). A novel ensemble deep learning model for stock prediction based on stock prices and news. International Journal of Data Science and Analytics, 1-11.

[61] Matthews Correlation Coefficient is The Best Classification Metric You've Never Heard Of by Boaz Shmueli, Online Available: https://towardsdatascience.com/the-best-classification-metric-youve-never-heard-of-the- matthews-correlation-coefficient-3bf50a2f3e9a (Last Access on: 4th January, 2023)

[62] Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. Pattern recognition letters, 30(1), 27-38.

[63] Dembczyński, K., Waegeman, W., Cheng, W., & Hüllermeier, E. (2010). Regret analysis for performance metrics in multi-label classification: the case of hamming and subset zero-one loss. In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part I 21 (pp. 280-295). Springer Berlin Heidelberg.

[64] Fletcher, S., & Islam, M. Z. (2018). Comparing sets of patterns with the Jaccard index. Australasian Journal of Information Systems, 22.

[65] Classification Problem:Relation between Sensitivity,Specificity and Accuracy by Mohit Khanna, Online Available:https://www.analyticsvidhya.com/blog/2021/06/classification-problem-relation-between-sensitivity-specificity-and-accuracy/(Last Access on: 1st June, 2023)

[66] Vanderlooy, S., & Hüllermeier, E. (2008). A critical analysis of variants of the AUC. Machine Learning, 72, 247-262.