SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I6P107

Volume 11 Issue 6, 74-87, June 2024
© 2024 Seventh Sense Research Group®

Original Article

Stacked Optimized Ensemble Machine Learning Model for Predicting Stock Trends through Candlestick Chart Analysis with Feature Engineering Approach

R. Sumathi[1], S. Ashokkumar[2]

[1,2]Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (Deemed to be University), Chennai, India.

[1]Corresponding Author : r.sumathi.sap@gmail.com

Received: 13 April 2024 Revised: 24 May 2024 Accepted: 09 June 2024 Published: 29 June 2024

Abstract *- The process of predicting stock trends through the analysis of Candlestick Charts (CCs) involves interpreting the patterns formed by these candlesticks to make informed predictions about future price movements. Utilizing Machine Learning (ML) for Stock Trend Prediction (STP) through CC analysis is common in algorithmic trading. CCs provide crucial information about the high, open, closed, and low prices within a specific time rate. However, stacked ensemble methods are employed to enhance reliability and stability, which combine the predictions of multiple models. Motivated by this objective, this work introduces the Stacked Optimized Ensemble ML Techniques with a Feature Engineering Approach for STP, referred to as SOEMLT-FEA. In the training phase, various models, including Random Forests (RF), SVM (Support Vector Machine), XGBoost, Decision Tree (DT), Adaboost, and ANN (Artificial Neural Network), are trained and optimized using the Chiroptera Algorithm (CA) to fine-tune their parameters. The optimized classifiers are then ranked, and the top three models are selected as the base classifiers for a stacking ensemble method. The efficacy of the developed feature engineering approach is confirmed by the experiential outcomes obtained (2000 and 2017) in China's stock market. This approach demonstrates promising economic returns for individual portfolios and stocks, achieving a prediction accuracy exceeding 90% for specific trend patterns.*

Keywords *- Stock trends, Candlestick chart, Future price movements, Stacked ensemble machine learning methods, Feature engineering scheme, Chiroptera algorithm.*

1. Introduction

The efficacy of STP techniques is crucial in attracting more individuals to the market and enhancing overall market confidence. Fundamental examination and methodological analysis are the two principal approaches utilized for predicting stock price activities and informing investment choices [1]. Within the stock market, sellers and buyers convene to execute transactions to maximize their ROI (Return on Investment).

The fundamental economic principles of demand and supply drive fluctuations in stock prices. Predicting trends in the Stock Market (SM) is challenging due to its non-linear characteristics, which are impacted by various factors, including company-specific news, global economic conditions, company profiles, and public sentiment. Over the years, predicting SM trends has enticed many data scientists, given the need to navigate the extensive data produced by the market and undertake both fundamental and technical analyses. The efficacy of Deep Learning (DL) and ML techniques has been demonstrated in this context. Fundamental analysis thoroughly examines a company's financial data, encompassing elements such as the EBITA (Earnings Before Taxes, Amortization) sheet, interest, and quarterly stock results [2]. Conversely, technical analysis is a more frequent process, conducted daily, and involves the scrutiny of Candlestick Patterns (CPs) and the observation of moving averages and other indicators.

Research has revealed a strong correlation between the upcoming trend in the generation of CPs and the SM. Fundamental analysis involves a comprehensive review of an organization's economic data to define its true value and predict future stock values.

Consequently, most investors regard fundamental analysis as particularly well-suited for long-term predictions. Conversely, procedural analysis operates on the premise that historical patterns tend to repeat themselves, seeking to forecast future stock price movements by scrutinizing past trends in stock prices [3].

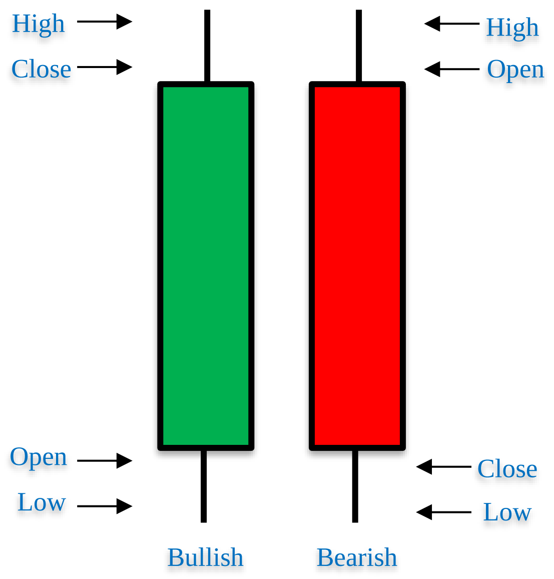

Fig. 1 Candlestick chart (Bullish/Bearish)

An essential technical analysis element involves examining CC patterns [4], commonly known as "candlestick charting." Figure 1 depicts numerous CC patterns recognized as signals for bearish/bullish continuations and reversals. According to historical records, the candlestick charting method is believed to have emerged sometime after 1850 [5]. Despite its popularity and longstanding history, studies employing the candlestick charting technique yield varied results. Research on the predictability of candlesticks yields conflicting findings, with negative conclusions reported [6].

In contrast, positive results emerge for various CC patterns through research on the U.S., Asian SMs, and European markets [7-10]. The revisions [9] employ descriptions that include a series of dissimilarities with diverse constraints to stipulate CPs, contributing to the ongoing controversy surrounding numerical definitions of these patterns. Additionally, chart patterns do not strictly adhere to time series, as stock price fluctuations persist at regular intervals spanning several days, influenced by declarations of significant political news, economic indicators, and other factors [10]. In these intervals, candlesticks are frequently recorded as short-body candlesticks that overlap, forming a sequence of turbulent candlestick patterns.

A preceding study [10] explores a chart retrieval framework incorporating various parameters to characterize CPs. This model utilizes the dynamic programming technique called nLCSm, a numerical adaptation of the LCS (Longest Common Substring) system. Despite the prior investigation successfully predicting future stock trends, it needs more transparency, attributable to the comprehensive management of all necessary processes, including the treatment of noisy candlesticks through the dynamic programming method.

1.1. Research Objectives

This paper presents a systematic methodology that utilizes a blending technique for CCs [11]. The research makes significant contributions in the following ways:

- This study enhances the scope of SM predicting by integrating traditional candlestick charting with cutting-edge ML methods.
- This work devised a straightforward eight-trigram classification system based on the eight-trigram scheme. Furthermore, a comparative analysis of diverse categories of technical indicators concerning their contribution to short-term stock prediction is conducted. This examination aims to elucidate the role played by these indicators when incorporated into ML models.
- The proposition involves the introduction of the SOEMLT-FEA method selection framework, accompanied by a novel feature engineering approach. This framework enables the automatic selection of suitable ML prediction methods corresponding to different candlestick charting patterns.
- A predictive framework has been developed to construct an investment strategy. Empirical results demonstrate the effectiveness of this strategy in generating favourable economic yields for both distinct portfolios and stocks.

The following parts of this document are organized in the following manner: Section 2 elucidates the relevant literature, while Section 3 delineates the design of an ensemble prediction framework utilizing ML techniques. Section 4 showcases empirical findings obtained from stock data, incorporating comprehensive validation procedures. Lastly, Section 5 presents the concluding remarks of this paper.

2. Related Works

This section presents an analysis of current models based on the application of the proposed research. Cagliero et al. [12] introduced an approach suggesting the separation of ML and pattern recognition processes. This configuration enabled the trading system to produce a more detailed set of thoroughly validated trading recommendations. The proposal involved the selective filtration of ML-based trading suggestions, identifying those considered potentially unreliable based on recognized graphical patterns.

The approach explored diverse methods to integrate pattern recognition strategies with a range of ML models, encompassing both deep and shallow supervised models and autoregressive methods. By conducting experiments across various market exchanges and under diverse conditions, this approach's efficacy was illustrated concerning the trading system's return on investment and maximum drawdown. The drawback of ML-based STP models, whether shallow or deep, lies in their challenge to efficiently capture the dynamic and complex nature of economic markets, leading to limited adaptability and accuracy.

Liang et al. [13] introduced a predictive model for forecasting multivariate financial time series related to stocks. The model incorporated sequence similarity and sequential pattern mining to enhance the precision of STPs. The model improved upon traditional sequential pattern mining methods by integrating K-line pattern mining into multivariate time series information. The improvement in the candlestick charting method was based on the morphological features of the K-line, which have been validated by empirical data analysis. The goal was to overcome the limitations associated with outdated forecasting models for individual stocks. The introduced sequence similarity aligned with the financial market's volatility, considering various influencing factors, and effectively-identified analogous volatility patterns. However, it was worth noting that K-line pattern mining in STP may be susceptible to overfitting and may have limited adaptability to changing market conditions.

Ananthi and Vijayakumar [14] forecasted the stock price based on datasets gathered from diverse sources about specific equity, predicting the overall sentiment of the stock. The forecasting of stock prices encompassed regression analysis and the identification of CPs. The system produced signals on the candlestick graph, facilitating the anticipation of market movements with commendable accuracy. This enabled users to evaluate whether to categorize a stock as a 'Buy/Sell' and make well-informed decisions regarding shorting or opting for a long position through delivery. The accuracy of stock exchange predictions has been analyzed and enhanced to 85% by applying ML algorithms. The drawback of candlestick regression in STP was its vulnerability to noise and the challenge of accurately capturing the intricate market dynamics, affecting the model's reliability for trend prediction.

Liu et al. [15] integrated DL and RL (reinforcement learning) to propose a multifaceted data fusion framework referred to as Deep Reinforcement Learning (MSF-DRL). The framework mentioned encompasses technical metrics, stock data, and CCs, with technical pointers used to decrease the influence of noise on stock information. On the MSF basis, trading strategies were developed by extracting temporal features from technical indicators and stock data using an LSTM network. Additionally, a sequence of applying a CNN followed by a bidirectional LSTM was used to excerpt features from the CC. The RL module then utilized these combined features to make trading decisions. Reasonable trials on datasets consisting of Chinese stocks and selected stocks from the S and P 500 index were conducted to demonstrate the efficiency of the MSF-DRL method. This approach demonstrated superior profitability and a higher Sharpe ratio than alternative trading strategies. Mahmoodi et al. [16] designed to improve the precision of predicting SM trading signals by employing an appropriate structural framework. Two models for technical adaptation analysis were utilized for this purpose. In conjunction with PSO, SVM was employed, where PSO served as a rapid and precise classification method to explore the problematic space. Subsequently, the obtained results were combined with the routine of two algorithms, precisely the NN and the CS (Cuckoo Search) system. The findings indicated that all the models exhibited reliability within six days. Notably, SVM-PSO outperformed the baseline research, obtaining a hit rate of 77.5%, while the hit rates for NN and SVM-CS are 71.2% and 71.4%.

The study focused on 2013–2021, suggesting that more optimal results might be achievable with extended periods. The SVM-PSO method demonstrated better performance than SVM-CS and even outperformed the commonly recognized feed-forward static NN algorithm, considered the field benchmark. The study introduced two separate methods for generating input data for the model: signal-based and raw-based approaches. It is important to note that unforeseen events were not considered despite using historical data. The accuracy of the predictions was assessed by calculating the hit rate, i.e., the percentage of correct predictions over 16 days. However, the combination of SVM and PSO for STP has a drawback regarding potential overfitting and limited robustness, particularly when adapting to financial markets' dynamic and non-linear nature.

Shah et al. [17] proposed a framework leveraging an integration of LSTM and CNN for predicting the concluding price of the Nifty 50 SM index. The framework was structured to extract features from various data types, encompassing raw cost data from foreign indexes, commodities, and exchange rates of currency, technical indicators, and the target index. These features were selected based on similarities and recognized trade setups within the industry. For time series modelling, the framework utilized a look-back period of 20 trading days to predict the movement of the following day.

Impressively, the model effectively captured information from these features and achieved a MAPE (Mean Absolute Percentage Error) of 2.54% when predicting the target variable, the closing price, over a 10-year data span. The framework exhibited a significant return enhancement when contrasted with the conventional buy-and-hold method. Jearanaitanakij and Passaya [18] proposed architecture for predicting short-term stock trends by combining a CNN with CPs. They conducted experiments using a dataset of CP images collected from different stocks in the SET (Stock Exchange of Thailand). Each image in the dataset consists of six to twelve consecutive candlesticks. By leveraging CNN and CPs, the architecture aimed to provide accurate predictions of short-term stock trends based on the patterns observed in the CCs. The investigational results demonstrated that the method accurately predicted short-term trends for most stocks. Furthermore, the architecture surpassed the renowned ResNet-18 in terms of both training time and accuracy. The limitation of using CNNs for STP lies in their restricted capability to capture the temporal needs and sequential patterns inherent in financial time series data. Lin et al. [20] employed DL methods for analyzing stock information precisely to forecast the track of ultimate prices.

The presented basis provided a suitable ML prediction technique for each pattern based on the accomplished outcomes. The speculation approach was formulated employing ensemble ML methods. Empirical results covering the period (from 2000 to 2017) in China's SM validated that their feature engineering exhibited active predictive capability.

The model attained a prediction accuracy of over 60% for STP. The drawback of DL methods for STP was their susceptibility to limited interpretability and overfitting, hindering the model's ability to generalize effectively in dynamic and complex financial markets.

2.1. Research Gap

Examining STP through ML uncovers notable shortcomings, encompassing challenges in comprehending the intricate and dynamic nature of financial markets, vulnerability to overfitting, and struggles in accommodating evolving conditions. Concerns such as data quality, interpretability, and reliance on feature engineering further constrain the efficacy of individual models. Nevertheless, the survey underscores the merits of stacked ensemble ML methods in this domain.

Through the amalgamation of diverse models, stacked ensembles offer enhanced predictive performance, improved generalization to new data, and greater adaptability to intricate patterns. This approach mitigates risks associated with depending solely on single algorithms, providing a more dependable and robust framework for STP. Thus, it addresses some limitations observed in standalone ML models.

3. Proposed Methodology

This study expands the horizons of SM prediction by combining traditional candlestick charting with advanced ML approaches. The research introduces a straightforward eight-trigram classification system based on an eight-trigram scheme, providing a systematic approach to categorizing CPs. Furthermore, the study conducts a comparative analysis of various technical indicators to understand their roles in short-term stock prediction when integrated with ML models. The novel introduction of the SOEMLT-FEA method selection framework and an innovative feature engineering scheme allows for the automatic selection of appropriate ML prediction methods corresponding to different candlestick charting patterns. The developed predictive framework for constructing an investment strategy demonstrates empirical effectiveness, showcasing favourable economic revenues for individual portfolios and stocks. This research advances SM prediction methodologies by leveraging the synergy between traditional charting and state-of-the-art ML approaches, offering a comprehensive and efficient prediction system, as shown in Figure 2.

3.1. Dataset Description and FEA

Initially, 13 types of one-day patterns are formulated and categorized from a dataset comprising 3,455 stocks in this study. Subsequently, eight-trigram information and technical indicators corresponding to each pattern are computed. Subsequently, these feature data are fed into the ensemble ML model, which evaluates the prediction accurateness for each pattern. The SOEMLT method with the maximum prediction accuracy for each pattern is duly acknowledged.

Ultimately, the adaptive recommendation plan prescribes specific SP activities based on the assessed outcomes [19]. The foundation of the prediction framework relies on a simple 8-trigram structure to illustrate daily stock price actions depending on 2-day CPs. The CC, known as the K-line, is constructed using high, close, low, and open values, with the segment between close and open designated as the real body. In the Chinese Stock Market (SM), the frame will burst with red if the strength exceeds its opening.

Conversely, in European and American SMs, it is signified in green or white. On the other hand, if the closing price is lower than the opening's worth, the frame is green in the Chinese SM and filled with red or black in European and American SMs. The CPs are categorized into 13 classes based on fundamental elements: opening, closing, low, and high prices. Compared to the closing value of the preceding day, the day's opening position indicates sentiment accumulation during non-trading periods. The inter-day charge measure is then categorized into eight segments based on the comparative location of yesterday's K-line patterns and today's price range. Additionally, trading volume, a parameter often overlooked in academia, is considered a crucial inconstant autonomous of price [19].

3.2. Prediction Model using SOEMLT

Motivated by the objective of enhancing STP, this study introduces the Stacked Optimized Ensemble ML Techniques with Feature Engineering Scheme, abbreviated as SOEMLT-FEA. In the training phase, diverse models such as RF, SVM, XGBoost, ANN, Adaboost, and DT are employed and optimized using the CA for parameter fine-tuning.

Stacking these optimized models in an ensemble framework aims to capitalize on their strengths and collectively improve predictive performance. This approach showcases a comprehensive strategy, combining various ML techniques with an FEA to enhance the robustness and accuracy of STP. The hyperparameters of these classifiers are designed using CA.

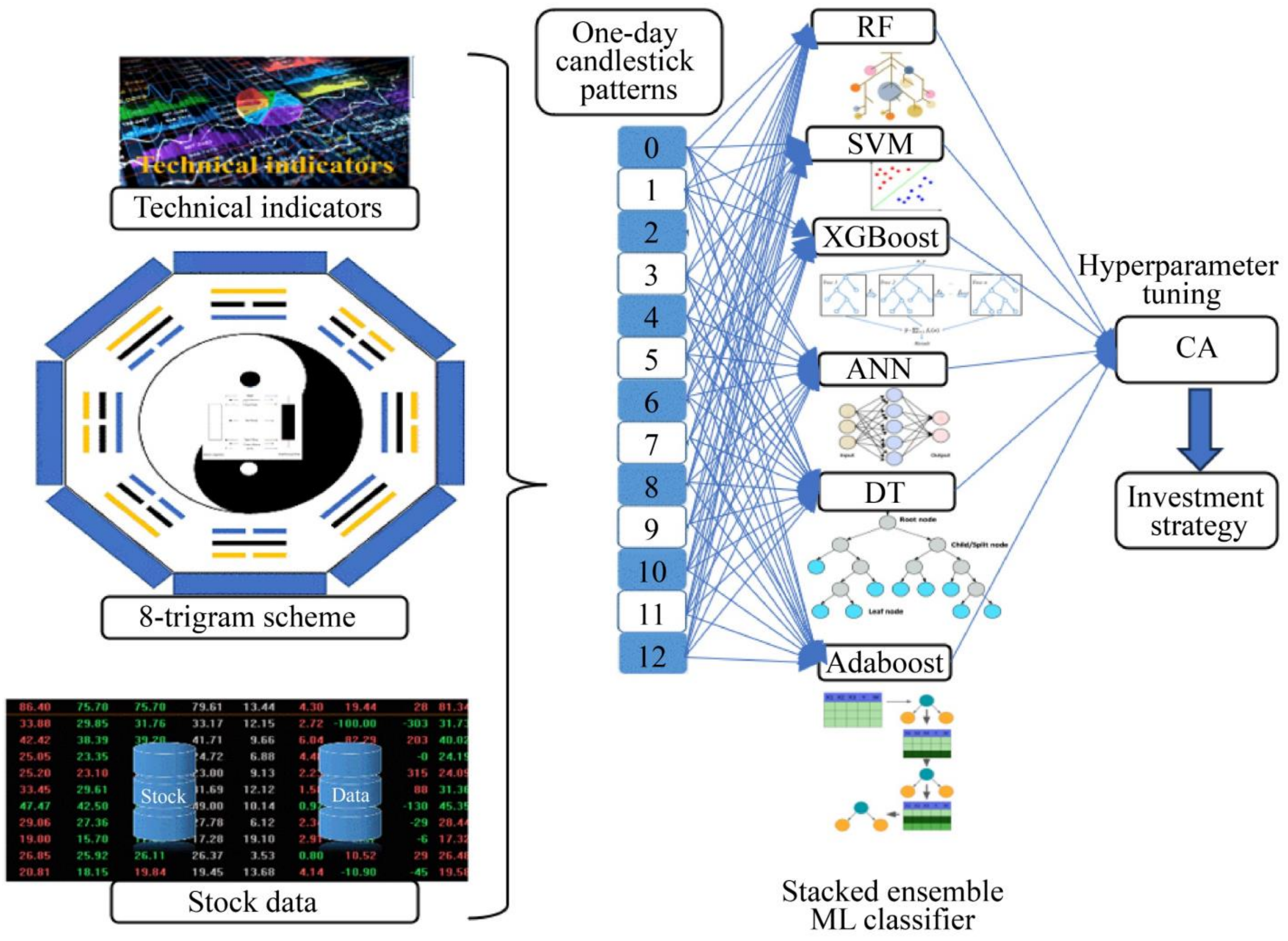

Fig. 2 Structure of proposed SOEMLT-FEA-based stocks trend prediction model

Random Forest

RF is a supervised ML algorithm that utilizes DTs as the foundational building blocks in its ensemble structure [20]. The algorithm operates on the principle of majority voting, wherein each DT independently forecasts the class for an opinion, and the period with the maximum number of ballots is assigned as the final confidential label.

To mitigate the data sensitivity and instability inherent in individual DTs, RF addresses these challenges by training every tree on the random samples with replacements, a procedure identified as bagging. One notable distinction between DT and RF lies in feature randomness. While a decision tree treats all the features while building its hierarchical architecture, each tree within the RF was trained on the subsets of randomly selected attributes.

This approach adds an element of diversity to the ensemble, contributing to the algorithm's robustness. The workflow of the RF procedure is outlined in Table 1.

Table 1. Random forest construction algorithm

1.	Employ a random selection process to extract stock models of size k from the unique data set D.
2.	Iteratively construct a specific tree for each extracted stock data.
3.	Produce an output utilizing each independently formed tree.
4.	Tally the frequency of every class label and designate the one with the maximum count as the ultimate classification.

Hyperparameter Tuning

Refine hyperparameters, including but not limited to the minimum samples per leaf, tree depth, and the number of trees, to enhance the model's overall performance.

Support Vector Machine

Originally familiarized as a binary classifier, the SVM adeptly segregates information into two modules by discerning an optimum hyperplane that exploits the margin between Support Vectors (SV) [21]. The hyperplane serves as the decision boundary, distinguishing between different classes. The determination of the optimal hyperplane involves solving a quadratic optimization delinquent, expressed as follows:

$$f(x) = bias + w^T x \quad (1)$$

In this Equation, w signifies the weight vector, b denotes the $bias$, and x signifies the SVs. SVs play a pivotal role as essential data points near the hyperplane, contributing significantly to its definition.

Hyperparameter Tuning

Optimize the performance of the SVM model by finely adjusting hyperparameters, including kernel-specific parameters and the regularization parameter (C).

ANN

The operation of the human nervous system serves as a model for constructing an ANN or an MLP (Multilayer Perceptron). An ANN is characterized by layers of consistent neurons, which neurons could embody various measured functions for information collection and analysis [22]. The neural network comprises three distinct layers: input, hidden, and output. The input layer describes the input design, while the output layer maps the input data to predefined classifications. Hidden layers in a neural network are assigned weights to enhance network optimization and minimize errors. In the context of the planned study, an ANN is configured with 14 nodes in the input layers and two nodes in the output layers to evaluate the SP. The NN has three hidden layers, employing a ReLU (Rectified Linear Unit) with an Activation Function (AF).

Hyperparameter Tuning

Fine-tune hyperparameters, including the count of layers, nodes in all layers, and activation functions, to optimize the ANN model's performance.

XGBoost

XGBoost, an abbreviation for EML Learning (eXtreme Gradient Boosting, is a scalable Ensemble Machine) technique specifically designed for speed and high performance. It is proficient in solving ranking, classification, and regression problems across various datasets. XGBoost achieves this by decomposing the objective function $OF(\Theta)$ into two main components: a shared differentiable training loss function, denoted as $Loss(\Theta)$, applicable to all RTs (regression trees), and a distinct regularization term for each RT, represented as $\Omega(\Theta)$:

$$OF(\Theta) = Loss(\Theta) + \Omega(\Theta) \quad (2)$$

The selection of an appropriate loss function depends on the dataset's characteristics. However, the regularization term in the $OF(\Theta)$ is determined by the number of leaves, two Loss Functions (LFs), and supplementary constants, as illustrated in Equation (3).

$$\Omega(\Theta) = \alpha\,[Loss_1(\Theta)] + \frac{1}{2}\lambda[Loss_2(\Theta)] + \gamma Q \quad (3)$$

Where $L_1(\Theta)$ and $L_2(\Theta)$ are LFs, Q is the sum of leaves, and ($\lambda, \alpha,$ andγ) are the XGBoost constants that make the model more conventional. Trees are constructed through a greedy function, which calculates the gain to facilitate the determination of the optimum split conclusion. DTs are concurrently built in a multi-level manner in XGBoost, ensuring tree attributes are sorted once at each stage.

Hyperparameter Tuning

Fine-tune hyperparameters, such as maximum depth, the number of boosting rounds, and learning rate, to optimize the XGBoost model's performance.

Decision Tree

DT is classified under supervised learning and is suitable for regression and classification tasks. Conceptually, a DT is depicted as a directed edge and hierarchical structure comprising nodes [23]. Within this structure, the classification process initiates from the root nodes, where all levels present a set of queries. As a dataset record responds to a query, it proceeds to the subsequent level for further inquiries. Each terminal node in the tree structure represents a specific class tag. Despite its simplicity and effectiveness in handling high-dimensional data, decision trees suffer a significant drawback—their susceptibility to instability. A minor change in the data can result in a substantial alteration of the whole structure. The alternative limitation is the prolonged training time required [23]. The construction of a decision tree relies on Information Gain (IG) principles and entropy as measures for attribute collection. At all levels, the trait with the lowermost entropy was nominated for data partitioning. If a branch attains zero, it becomes a leaf node; otherwise, the branching process continues. The mathematical calculation of entropy for dissimilar attributes is detailed in Table 2.

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i \quad (4)$$

Table 2. Decision tree construction algorithm

1.	Initiate the tree construction with the R (root nodes) and the D data set.
2.	In each iteration, calculate entropy to identify the optimal idle attribute A.
3.	Utilize the designated A to partition the D into subsets of data.
4.	Iterate the process on each subcategory till a point is reached with the highest information gain.

AdaBoost

AdaBoost is an ensemble technique based on gradient boosting that does not necessitate prior information of weak learner accuracies. Functioning as a DT algorithm, it generates a forest of stumps characterized as simple trees with a single node and two leaves. These bases are identified as weak learners with limited regression capabilities. Each base provides an individual vote, assigned according to the error of diverse vote w. In each step of the AdaBoost algorithm, the dataset's total vote for a feature is designed based on the total error, err_t, representing the sum of votes w for imprecise samples. The cumulative vote for each stump can be considered as follows:

$$err_t = W_1 + W_2 + \cdots + W_i \tag{5}$$

Where (i) indicates the count of records in the D. Consequently, the vote is determined according to the subsequent Equation:

$$\upsilon = \frac{1}{2}\log\frac{1-err_t}{err_t} \tag{6}$$

In the context of AdaBoost, when the total err_t (0 to 1), higher votes are assigned to lower err_t and vice versa. When err_t = 0.5, the present stump displays no enhancement from the previous phase, as illustrated in Figure 6. The adapted weight, W_{t+1}, is considered as shown in Equation (7):

$$W_{t+1} = W_t \times err^{\upsilon} \tag{7}$$

Adjustments to other features are made based on this modification by applying the following relation (8) to the remaining features in the dataset.

$$W_{t+1} = W_t \times err^{-\upsilon} \tag{8}$$

The overall sample W_t undergoes normalization, and the adapted W_t are allocated to each feature for the subsequent stump in the ensemble. The weighted Gini index is employed to identify the feature that should be split in the next stump, considering the largestW_t. This process accentuates a feature based on a random number$(0,1)$, where the generated number determines the dataset feature with the most substantial influence on the next stump. The iterative execution of this process continues until the predefined stopping measure is met.

Stacking Ensemble Method

Ensemble learning (EL) is a hybrid ML method that leverages predictions from numerous base models to enhance overall predictive demonstration [24]. The base model can be built using various ML algorithms. An ensemble comprising a similar assortment of base learners is referred to as a similar EL model; otherwise, it is termed heterogeneous. EL encompasses three main measures: stacking, boosting, and bagging. Bagging contains the independent training of weak learners and produces the forecast average from diverse ML models. Boosting, on the other hand, sequentially adds base learners and yields the prediction weighted average made by these base models. Stacking is an EL where base classifiers are trained on the same data set, and an extra classifier, the meta-learner, is employed to enhance the overall model routine. In this work, Adaboost is utilized by incorporating predictions from individual classification models to determine the STP, as shown in Table 3. Figure 2 illustrates the stacking ensemble based on ML techniques such as RF, SVM, XGBoost, ANN, DT, and Adaboost.

Table 3. Algorithm steps of stacked ensemble ML classifier

```
Input: Training dataset $D = \{X_i, Y_i\}_{i=1}^{m}$
where $X_i \in$ set of features and $Y_i \in$ class labels
Output: STP.
1: Split the $D$ into m equal parts such that $D = \{D_1, D_2, D_3 \ldots D_m\}$
2: for b = 1 to B, do
    3: Develop base classifier using dataset D, ensuing steps 4 to 7.
    4: Compute the weighted sum and include the bias for all nodes in the hidden layers using $Inf\ o = \sum_i^n x_i \times W_i + bias$.
    5: Compute the values of $\Delta W = W - \eta \frac{\partial E}{\partial W}$ and $E = \frac{1}{2}\sum_{p=1}^{n}\sum_{o=1}^{m}(T_{io} - A_{io})^2$.
    6: Fine-tune the learning parameters and adjust weights iteratively till the lowest $err$ rate is reached.
    7: At all base classifiers, apply a ReLU: $f\,(Info\,) = max\,(0, Info\,)$.
8: end for
9: Create a training set tailored for XGBoost.
10: for i = 1 to m do
    11: $D_E = \{x'_i, y_i\}$, where $x_i = \{h_1(x_i), \ldots, h_B(x_i)\}$
12: end for
13: Train an XGBoost using $D_E$.
14: Return predictions$y_i = \{y_1, y_2, y_3, \ldots, y_n\}$ from the formed ensemble model.
```

3.3. Hyperparameters Tuning of Stacked Ensemble ML Classifiers using CA

Hyperparameters, distinct from model parameters learned during training by model algorithms, are adjustable parameters that can significantly influence the performance of an ML model. Hyperparameter tuning is an external process conducted by the data scientist before training. In optimization, parameter setting serves as a strategy designed to enhance the flexibility and robustness of solvers, but it necessitates meticulous initialization [25]. Algorithm parameters significantly influence the efficiency of solving processes, and determining the optimal parameter setting is not straightforward. The ideal values depend on the specific problem, the instance being addressed, and the desired search time for problem-solving.There exists no universally optimal set of parameter values for a given computational intelligence algorithm [26]. The approach of online parameter control utilizes a bio-inspired problem solver to precisely identify the optimal hyperparameter set for stacked ensemble ML techniques. This methodology functions as a loop declaration that transfers feedback, or accuracy, from the stacked ensemble ML classifiers to the optimizer. The optimizer then works to enhance the results obtained. The Chiroptera algorithm is built upon three basic rules.

1. The underlying assumption is that all Chiropteras utilize echolocation to determine distances and possess the ability to differentiate between prey, background barriers, and food sources.
2. Each Chiroptera, called $chiroptera_i$, actively searches for prey at a specific positionx_i, initially chosen randomly. The Chiropteras adjust their frequency based on the proximity of their target, which subsequently affects their velocity. To transform their position, Chiropteras utilize a frequency $freq_i$ calculated using Equation (9), and their velocity v_i is determined through Equation (10). The new position is then determined using Equation (11). The Chiroptera procedure can be characterized as a frequency-tuning procedure that balances exploitation and exploration. Higher (positive) velocities correspond to increased exploration, whereas lower (positive) velocities correspond to enhanced exploitation.

$$freq_i = freq_{min} + (freq_{max} - freq_{min})\beta \ \left(\beta \sim (0,1)\right) \quad (9)$$

$$v_{(i,t+1)} = v_{(i,t)} + \left(x_{best} - x_{(i,t)}\right)f_i \quad (10)$$

$$x_{(i,t+1)} = x_{(i,t)} + v_{(i,t+1)} \quad (11)$$

3. In the Chiroptera algorithm, the inconsistency of solutions is represented by the loudness $Loud_0$ Moreover, the rate of pulse release r, which falls within the range of (0,1). The $Loud$, determined by Equation (7), can vary in various ways. However, it is implicit that the $Loud$ starts from a great (positive) value $Loud_0$ and gradually decreases to a minimum constant value $Loud_{min}$. The rate of pulse emission, denoted by r and calculated using Equation (8), influences the exploration and exploitation balance of the algorithm.

$$Loud_{(i,t+1)} = \alpha Loud_0 \ 0 < \alpha < 1 \quad (12)$$

$$r_{(i,t+1)} = r_{(i,0)}(1 - \exp^{(-\gamma t)})\gamma > 0 \quad (13)$$

Table 3 displays the pseudocode for CA. Initially, a population of m Chiropteras is set with x_i and v_i. The location of each Chiroptera, represented by $chiroptera_i$, is a vector consisting of a set of weights and biases. This vector is utilized for tuning and training the stacked ensemble classifiers, with each Chiroptera solution serving as the position$chiroptera_i$. The training process involves a single epoch and returns the attained loss and accuracy. Subsequently, the frequency f_i at x_i is established, followed by $Loud$ and pulse charges, which are determined through uniform distribution (0,1). Finally, the OF evaluates the accuracy and loss as fitness. This study employs the OF in the following manner.

$$fit \rightarrow \text{maximize } k \cdot accuracy + (1 - k) \cdot loss \quad (14)$$

In the computational experiments, a constant k is used to weigh the objectives of accuracy and loss. Specifically, k takes on values from the set {0, 10, 20, ..., 100}. This means that the OF is evaluated eleven times for each value of k. The fitness rate is determined as the best (greatest) value obtained and stored in the fitness vector's ith position, fit. The subsequent step entails a while loop encompassing a set of movements to be iteratively executed t times till the T repetitions. In lines 6-15, the $Loud$ of each Chiroptera is associated with a random rate.

Table 4. CA for hyperparameters of stacked ensemble ML classifiers

```
Data: $m, f_{min}, f_{max}, t_{max}, \alpha, \gamma$, and $\varepsilon$.
Result: $x_{best}$.
// Initialize the Chiroptera population as hyperparameters of stacked ensemble ML classifiers:
// velocity $v_i$, $Loud_i$, $x_i$, and $r_i$.
1.  for $bat\ i, (i = \{1, \dots, m\}$ do
2.  $v_i \leftarrow Random[0,1]$,  $x_i \leftarrow Random[0,1]$,  $A_i \leftarrow Random[0,1]$, $r_i \leftarrow Random[0,1]$;
3.  $fit_i \leftarrow objective\_function(x_i)$;
4.  end
5.  global fit ←
6.  hyperparameters of stacked ensemble ML classifiers;
// Produce $t_{max}$-generations of m Chiropteras.
7.  while $t < t_{max}$ do
8.  for $chiroptera_i, (i = \{1, \dots, m\})$ do
//  If the $Loud$ of the $i$th Chiroptera exceeds that of any other Chiroptera.
9.  if $Loud_i > Random[0,1]$ then
10. $Loud_i \leftarrow \alpha Loud_i, r_i \leftarrow r_0[1 - e^{(-\gamma t)}]$;
11. end
12. end
13. $\{best\ f\ it, bestindex\} \leftarrow eval(fit)$;
14. if $the\ best\ fit\ is\ better\ than\ the\ global\ fit$, then
15. $global\ fit \leftarrow best\ fit, x_{best} \leftarrow x_{bestindex}$;
16. end
17. for $bat\ i, (i = \{1, \dots, m\}$ do
18. if $r_i > Random[0,1]$ then
19. $x_i \leftarrow x_i + \varepsilon \overline{Loud}$;
20. end
21. if $Loud_i > Random[0,1]$ and $fit_i < global\ fit$ then
22. $\beta \leftarrow Random[0,1]$;
23. $f_i \leftarrow f_{min} + (f_{max} - f_{min})\beta, v_i \leftarrow v_i + (x_{best} - x_i)f_i, x_i \leftarrow v_i + x_i$ ;
24. $fit_i \leftarrow OF(x_i)$;
25. end
26. end
27. end
28. Return the
```

If the $Loud$ of the ith Chiroptera beats that of any other Chiroptera, it undergoes reduction using Equation (6), and the ratio of pulse emission is increased using Equation (14). This state accommodates the inconsistency of potential resolutions during exploitation and exploration. Smaller values for $Loud$ indicate strengthened results, accompanied by larger values for pulse emission rates.

Afterwards, the best fit and index are determined through the assessment of the OF. If the best fit is superior to the global fit, then the best fit is kept in the global fit variable. Subsequently, the loop declaration between lines 17 and 25 represents the movement of Chiropteras. A result is initially nominated from the current best results, and a new result is produced through random walks (Equation (15).

$$x_{new} = x_{old} + \varepsilon \overline{Loud} \tag{15}$$

In Equation (15), ε is a random variable that takes on values from the set {-1, 0, 1}, and $\overline{Loud}$ represents the average $Loud$ of all Chiropteras. The Chiropteras control the stride and array of their actions, where β introduces variability to the occurrences and is arbitrarily generated from a uniform distribution (0, 1). Equation (10) defines the velocity of the ith Chiroptera at time t, where x_{best} signifies the current global best location from the m Chiropteras. Lastly, Equation (14) regulates the new position of the ith Chiroptera.

Toward the end of the algorithm, the Chiropteras are ranked to find x_{best}, which represents the best configuration (hyperparameters) for the stacked ensemble classifiers and thus produces the superlative classification model. The efficiency of the Chiroptera algorithm has been demonstrated in various instances of combinative and optimization difficulties. The pseudocode is provided in Table 4. This study introduces an adaptive prediction framework that utilizes a stacked ensemble of ML models to estimate the way of the closing price.

3.4. Final Investment Strategy

The investment strategy outlined in this paper involves two scenarios: long-only and long-short. The approach is created based on the DE model mentioned earlier. The steps for constructing the asset approach are as given as follows:

- Check the precise K-line patterns of the up-to-date stocks at time t.
- Select a suitable ML technique from the evaluation model to predict the growth or fall of t + 1.
- If the predicted result aligns with the actual outcome, record the profit at t + 1.
- If the prediction is incorrect, store the negative return at t + 1 as a loss.
- Repeat the above stages to compute the profits and losses at t + 1, t + 2, etc.

This strategy only makes investments when the predicted result indicates an upward movement.

4. Experimental Results and Discussion

This study employs daily data from the CSM, spanning 18 years (2000 to 2017). The dataset encompasses information from all 3,455 stocks sourced from CCER, a limited data earner in China. The initial preprocessing step excludes daily information for a specific stock if the transaction size is 0, indicating a trading halt, possibly due to company redeployment. The dataset comprises 13 patterns, and their distribution is outlined in Table 3, illustrating a consistent historical distribution. Subsequently, feature information is generated for each stock on each day (t), encompassing the intra-day pattern, the date, the closing price for the subsequent day, 21 other indicator values, and the inter-day pattern. To ensure efficiency, three rounds of training are executed. In each round, five thousand rows of everyday stock data for all 13 intra-day patterns are randomly chosen from the database, resulting in 65,000 rows for every iteration. To maintain classification equilibrium during training, an equal number of instances with rising and falling prices are selected for each intra-day pattern.

This section presents the projected SOEMLT-FEA model, evaluated and compared with other commonly used methods in STP, including the Ensemble ML method [19], SVM-PSO [16], and SVM-CS [16]. The experiment results demonstrate that the proposed SOEMLT-FEA model, which utilizes a residual network with multidimensional feature comparison, outperforms traditional models. Performance criteria such as accuracy, sensitivity/recall, f-measure, specificity, and precision are used to assess the performance of the suggested model.

True Positives (TPs) represent accurately classified positive stock trends, while FNs (False Negatives) indicate negatively classified stock trends. True Negatives (TNs) refer to correctly predicted negative stock trends, while False Positives (FPs) represent instances where negative trends are incorrectly classified as positive. The proposed method’s qualitative results show a high accuracy level in identifying STPs. The convergence rate of the model’s contour is improved compared to previous classification approaches. Overall, the experimental results highlight the superior performance of the SOEMLT-FEA model in STP, surpassing traditional methods in terms of effectiveness and accuracy.

Recall

Recall measures the number of positive class calculations made from all positive instances in the dataset as specified below:

$$Recall = \frac{TP}{TP+FN} \tag{16}$$

Precision

Precision measures the number of positive class calculations that belong to the positive class, and it is projected as follows:

$$Precision = \frac{TP}{TP+FP} \quad (17)$$

F-measure

F-Measure carries a single score that balances both the concerns of precision and recall in one number, and it is assessed as follows:

$$F - Measure = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \quad (18)$$

Accuracy

The proportion of successfully segmented data relative to the total number of samples is a popular metric for evaluating classification performance.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (19)$$

Figure 3 shows the degree to which the suggested SOEMLT-FEA agrees with the currently employed models for a given pattern in each database. The SOEMLT-FEA reduces processing times without any loss of accuracy. The SOEMLT-FEA outperforms all other classification models, including the Ensemble ML method, SVM-PSO and SVM-CS, with an accuracy of 98.95% without requiring significant patterns during reduction.

The test findings also show that the SOEMLT-FEA is superior to the individual Ensemble ML method, SVM-PSO and SVM-CS models. The suggested model shows superior feature extraction and outperforms state-of-the-art approaches by a wide margin. The numerical accuracy results for projected and existing methods are shown in Table 5.

For a given subset of database properties, Figure 4 shows how the proposed SOEMLT-FEA compares to other models, such as the Ensemble ML method, SVM-PSO and SVM-CS, in terms of accuracy. The precision improves with time as measured in epochs. For example, the SOEMLT-FEA has a higher accuracy than earlier approaches (97.65%). When SPC-CNN and SOEMLT-FEA are combined, they produce the most outstanding results regarding loss values and overall performance across precision metrics.

Experiments show that the fusion-based hybrid can extract better characteristics for further categorization. The model used the CA method for parameter value selection for MKELM. To boost the model's performance and drastically cut down on false negatives, the IDBSCAN technique is fed into it. The fact that the proposed SOEMLT-FEA technique outperforms the standard model demonstrates its importance.

Table 5. The numerical results of accuracy for projected and existing methods

Number of patterns	SOEMLT-FEA	Ensemble ML method	SVM-PSO	SVM-CS
2	97.5	97	90	89
4	97.92	97.4	91	90
6	98.5	97.8	92.5	91
8	98.76	98	93.5	91.5
10	98.85	98	94	93
12	98.95	98.2	94.58	93.54

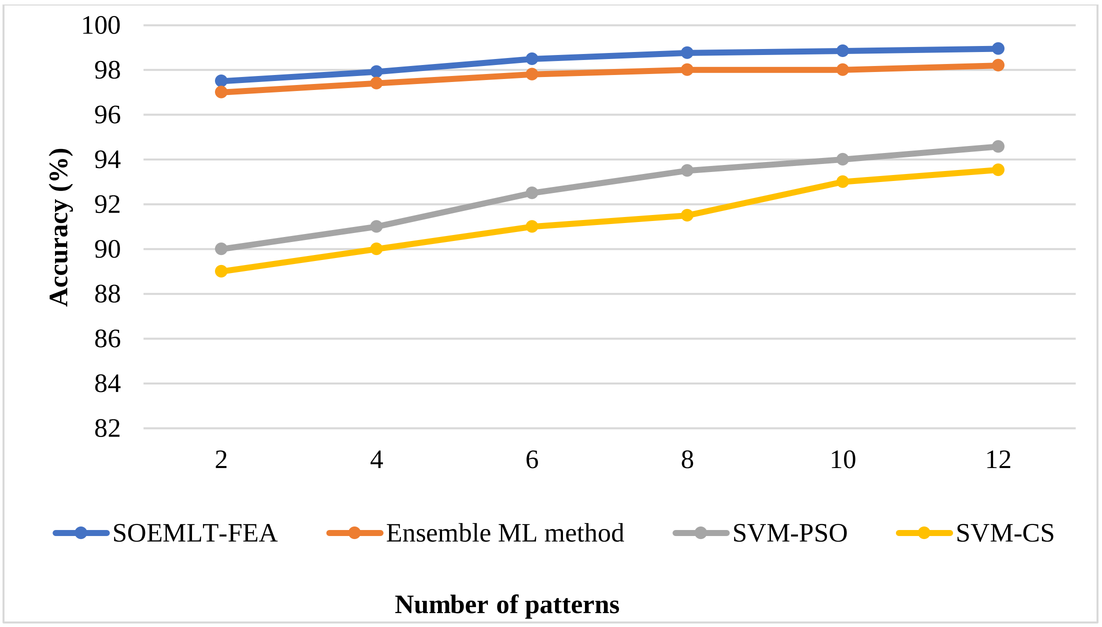

Fig. 3 Accuracy results in comparison

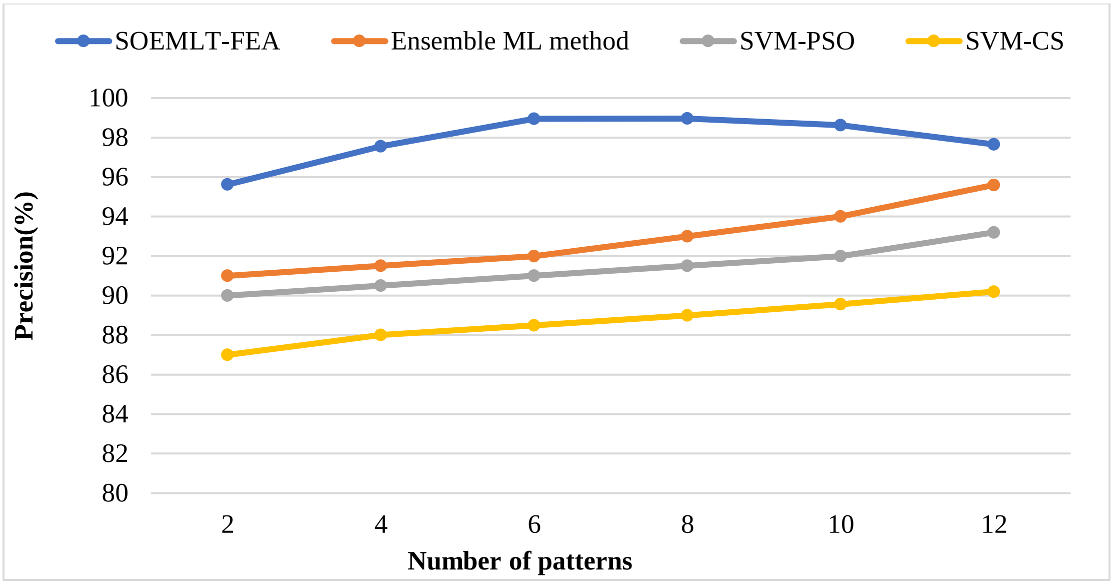

Fig. 4 Precision results comparison

The numerical precision results for projected and existing methods are shown in Table 6. The randomness of the hyperparameters impacts traditional ML models. By providing optimal hyperparameters, the proposed CA optimization algorithm expedites the convergence of the proposed SOEMLT-FEA model. The recall is shown in Figure 5 for the proposed SOEMLT-FEA and other current models like the Ensemble ML method, SVM-PSO and SVM-CS over a range of feature counts.

Table 6. The numerical results of precision for projected and existing methods

Number of patterns	SOEMLT-FEA	Ensemble ML method	SVM-PSO	SVM-CS
2	95.62	91	90	87
4	97.56	91.5	90.5	88
6	98.95	92	91	88.5
8	98.96	93	91.5	89
10	98.63	94	92	89.56
12	97.65	95.6	93.2	90.2

With more patterns, the recall rises For example, the SOEMLT-FEA achieves a recall of 98.56 %, higher than any prior approaches. According to the findings, the suggested SOEMLT-FEA model has superior STP performance compared to all other models. The numerical recall of precision for projected and existing methods is shown in Table 7. By applying the SOEMLT-FEA method independently on the weight and bias vectors, the proposed SPC-CNN and SOEMLT-FEA model is fine-tuned.

The proposed model has a higher F1 score for the number of patterns in the provided databases than the existing SOEMLT-FEA, ensemble ML method, SVM-PSO and SVM-CS models (see Figure 6). The epoch count and the f-measure are both tuned at the same time.Compared to other models, the SOEMLT-FEA achieves an f-measure of 98.23%. On both the accuracy and F1 scales, the SOEMLT-FEA emerged victorious. Indeed, the projected model has the competence to detect patterns and improve prediction accuracy by combining the results of the CA (Cellular Automata) model.

The proposed methodology offers valuable insights and methods to enhance STP by leveraging the CA model's outcomes. The numerical results of the f1-score for projected and existing methods are shown in Table 8.

Table 7. The numerical results of recall for projected and existing methods

Number of patterns	SOEMLT-FEA	Ensemble ML method	SVM-PSO	SVM-CS
2	95.62	93	87	85
4	97.56	94.5	87	85.2
6	98.62	95	88	86
8	98.47	95.5	88	86
10	98.24	96.3	89	86.5
12	98.56	97.3	89.33	87.2

Table 8. The numerical results of the f1-score for projected and existing methods

Number of patterns	SOEMLT-FEA	Ensemble ML method	SVM-PSO	SVM-CS
2	95.74	94.6	88	84
4	96.58	95	88.4	84.5
6	96.12	95.4	88.9	84.9
8	96.85	95.8	90	85
10	97.95	96	90.1	85.5
12	98.23	96.3	90.43	86.7

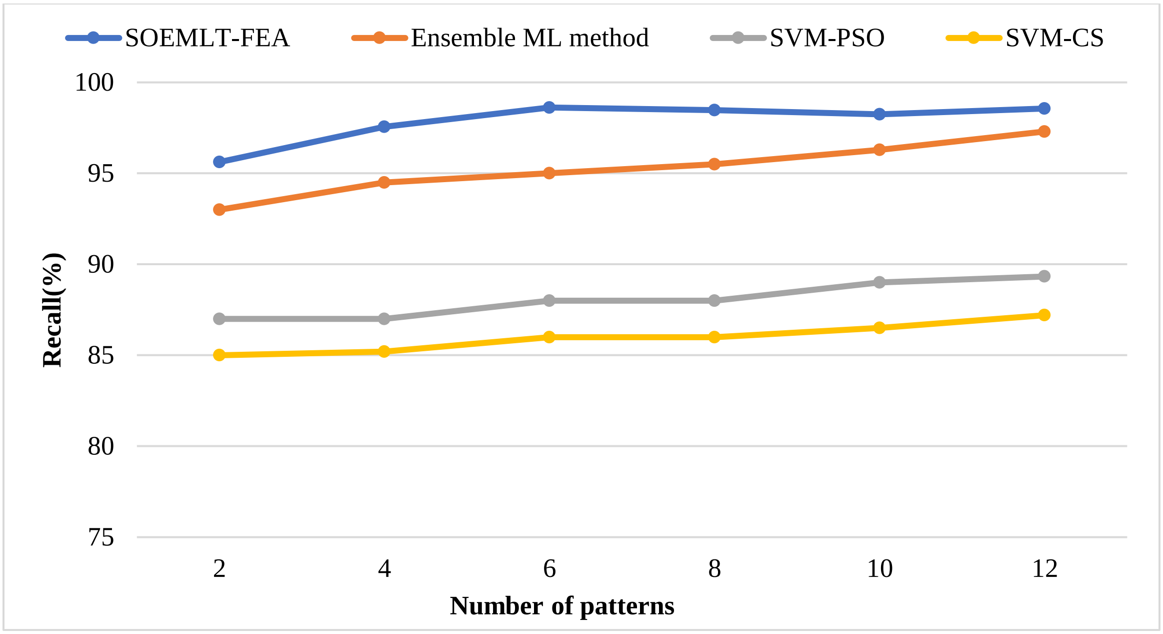

Fig. 5 Recall results comparison

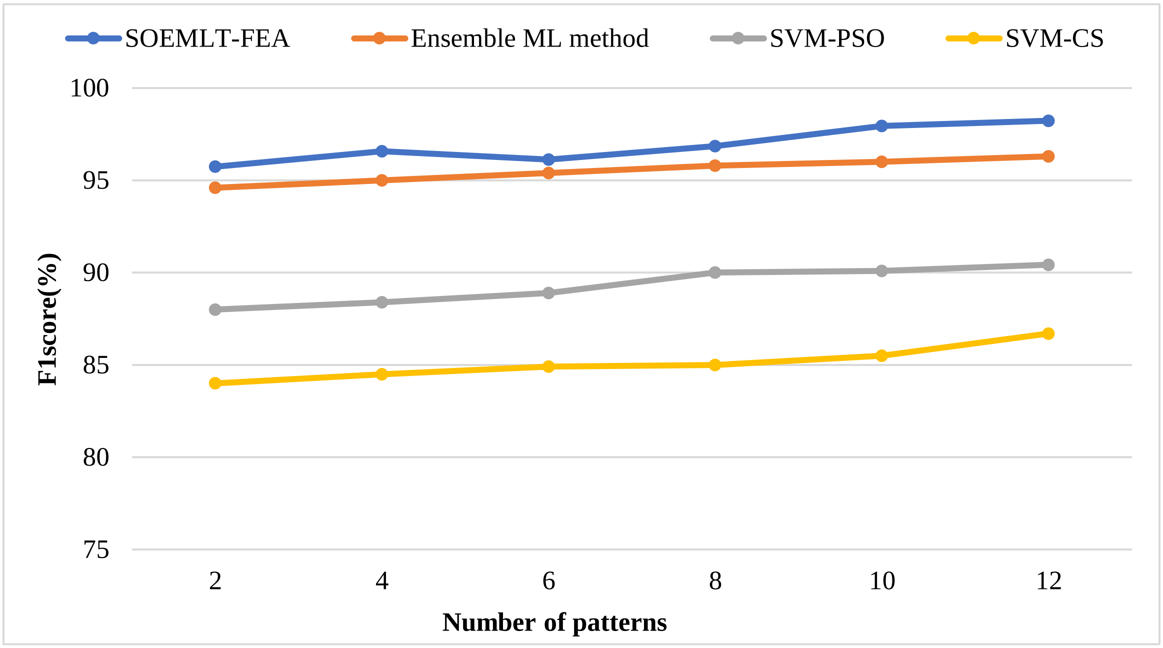

Fig. 6 F1-score results comparison

By evaluating the China stock dataset, the proposed model demonstrates its effectiveness and potential in providing valuable perspectives for STP. By incorporating the CA model's results, the proposed methodology introduces a novel approach that can contribute to improving the accuracy and reliability of STPs. The combination of the CA model and the proposed methodology opens new possibilities for enhancing STP by leveraging the strengths of both approaches. This innovative approach may offer valuable insights and methods that can further advance the SM analysis and prediction field.

5. Conclusion

This work presents an ensemble ML prediction model designed to automatically choose appropriate prediction methods for daily k-line patterns in SM prediction. The practical outcomes validate the effectiveness of this framework, with the investment strategy derived from the model showcasing superior returns. The contribution of this study is multifaceted. Firstly, it merges traditional Candlestick charting with cutting-edge AI techniques, adding depth to SM prediction research. Through scrutinizing the prediction outcomes of every 13 one-day candlestick patterns using diverse ML approaches, the research combines conventional technical analysis with artificial intelligence advancements, identifying candlestick patterns such as patterns 4 and 5 with notable analytical influences on SM. Furthermore, the study introduces an 8-trigram classification of two-day k-line patterns alongside the volume change features and 13 daily k-line patterns. This expanded feature set enhances the model's predictive capabilities. Thirdly, the SOEMLT framework uses six commonly utilized effective prediction approaches (SVM, RF, XGBoost, ANN, DT, Adaboost) and optimizes the parameters of each model. The experimental study reveals that RF and XGBoost consistently exhibit strong predictive abilities for short-term predictions. Finally, based on the prediction outcomes, the study formulates an investment approach. The experiential results illustrate that this model generates favourable economic returns in theory for individual portfolios and stocks. This suggests that the prediction results prove effective by leveraging big data, undergoing multiple training rounds, and implementing feature settings. However, it is crucial to acknowledge that actual transaction costs significantly influence real-world transactions. This paper presents a comprehensive approach combining traditional and AI methods, introduces new features, and utilizes ensemble ML to enhance SM prediction and develop profitable investment strategies. In real-world investment, other factors must be measured to achieve excess yields. Future research in predicting stock trends through DL could delve into various directions to augment the precision and resilience of predictive models. The following are potential directions for future investigations: Temporal Feature Engineering: Examine advanced methodologies for temporal feature engineering, considering the distinctive attributes of financial time series data. This might entail devising innovative features or incorporating external factors influencing stock prices. Hybrid Models: Investigate the amalgamation of diverse DL architectures or hybrid models. Integrating the capabilities of RNNs (Recurrent Neural Networks), CNNs, and transformers can enhance the model's efficacy in capturing both short-term and long-term dependencies within stock data.

Acknowledgements

The authors would like to thank the Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (Deemed to be University), Chennai, India, for their support and motivation throughout this research.

References

[1] Veliota Drakopoulou, "A Review of Fundamental and Technical Stocks Analysis Techniques," *Journal of Stock & Forex Trading*, vol. 5, no. 1, pp. 1-8, 2016. [Google Scholar] [Publisher Link]

[2] Divyanshu Bathla, Ashish Garg, and Sarika, "Stock Trend Prediction Using Candlestick Pattern," *Cybersecurity and Evolutionary Data Engineering*, pp. 235-246, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Peter R. Cox, *Technical Analysis*, Cambridge University, pp. 1-179, 2011. [Publisher Link]

[4] Michael C. Thomsett, *Bloomberg Visual Guide to Candlestick Charting*, John Wiley & Sons, 2012. [Google Scholar] [Publisher Link]

[5] Candlestick Chart Patterns, Incredible Charts, [Online]. Available: https://www.incrediblecharts.com/candlestick_patterns/candlestick-patterns.php

[6] Piyapas Tharavanij, Vasan Siraprapasiri, and Kittichai Rajchamaha, "Profitability of Candlestick Charting Patterns in the Stock Exchange of Thailand (SET)," *Sage Open*, vol. 7, no. 4, pp. 1-18, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[7] Nikitas Goumatianos, Ioannis T. Christou, and Peter Lindgren, "Useful Patterns Mining on Time Series: Application in the Stock Markets," *Proceeding of the 2nd International Conferences on Patterns Recognitions Application and Method*, Barcelona, Spain, vol. 20, pp. 608-612, 2013. [Google Scholar] [Publisher Link]

[8] Min Zhu, Said Atri, and Eyub Yegen, "Are Candlesticks Trading Strategies Effective in Certain Stocks with Distinct Features?," *Pacific-Basin Finance Journal*, vol. 37, pp. 116-127, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[9] Tsung-Hsun Lu, Yi-Chi Chen, and Yu-Chin Hsu, "Trend Definitions or Holding Strategy: What Determine the Profitability of Candlesticks Charting?," *Journal of Banking & Finance*, vol. 61, pp. 172-183, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[10] Yoshihisa Udagawa, "Designs and Implementations of Candlestick Charts Retrieval Algorithms for Predicting Stocks Price Trends," *The Fourth International Conferences on Big Data, Small Data, Linked Data and Open Data (ALLDATA 2018)*, pp. 19-25, 2018. [Google Scholar] [Publisher Link]

[11] Yoshihisa Udagawa, "Predicting Stocks Price Trend using Candlestick Charts Blending Techniques," *IEEE International Conferences on Big Data (Big Data)*, Seattle, WA, USA, pp. 4162-4168, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[12] Luca Cagliero, Jacopo Fior, and Paolo Garza, "Shortlisting Machine Learning-Based Stocks Trading Recommendation using Candlesticks Pattern Recognitions," *Expert System with Application*, vol. 216, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Mengxia Liang et al., "A Stock Time Series Forecasting Approach Incorporating Candlestick Patterns and Sequence Similarity," *Expert Systems with Applications*, vol. 205, pp. 1-26, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] M. Ananthi, and K. Vijayakumar, “Stock Markets Analysis Using Candlesticks Regressions and Market Trends Predictions (CKRM),” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 5, pp. 4819-4826, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Peipei Liu et al., “Multi-Type Data Fusion Frameworks Based on Deep Reinforcement Learning for Algorithmic Trading,” *Applied Intelligence*, vol. 53, no. 2, pp. 1683-1706, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Armin Mahmoodi et al., “A Developed Stocks Price Forecasting Model using Support Vector Machines Combined with Metaheuristic Algorithm,” *OPSEARCH*, vol. 60, no. 1, pp. 59-86, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Atharva Shah et al., “A Stock Market Trading Framework Based on Deep Learning Architecture,” *Multimedia Tools and Application*, vol. 81, no. 10, pp. 14153-14171, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Kietikul Jearanaitanakij, and Bundit Passaya, “Predicting Short Trends of Stock by Using Convolutional Neural Networks and Candlesticks Pattern,” *4th International Conferences on Information Technology (InCIT)*, Bangkok, Thailand, pp. 159-162, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[19] Yaohu Lin et al., “Stock Trends Predictions using Candlesticks Charting and Ensemble Machine Learning Technique with a Novelty Features Engineering Schemes,” *IEEE Access*, vol. 9, pp. 101433-101446, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Lili Yin et al., “Research on Stock Trends Prediction Methods Based on Optimized Random Forests,” *CAAI Transactions on Intelligence Technology*, vol. 8, no. 1, pp. 274-284, 2023. [CrossRef] [Publisher Link]

[21] Yuling Lin, Haixiang Guo, and Jinglu Hu, “An SVM-Based Approach for Stock Markets Trend Predictions,” *The 2013 International Joint Conferences on Neural Network (IJCNN)*, Dallas, TX, USA, pp. 1-7, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[22] Luca Di Persio, and Oleksandr Honchar, “Artificial Neural Network Architecture for Stocks Price Predictions: Comparison and Application,” *International Journal of Circuits, Systems and Signal Processing*, vol. 10, pp. 403-413, 2016. [Google Scholar] [Publisher Link]

[23] Rupesh A. Kamble, “Short and Long-Term Stock Trends Predictions Using Decision Trees,” *International Conferences on Intelligent Computing and Control Systems, (ICICCS)*, Madurai, India, pp. 1371-1375, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[24] Lokesh Kumar et al., “A Hybrid Machine Learning System for Stock Markets Forecasting,” *Journal of International Technology and Information Management*, vol. 20, no. 1, pp. 39-48, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[25] Xin-She Yang, and Xingshi He, “Bat Algorithm: Literature Review and Applications,” *International Journal of Bio-inspired Computation*, vol. 5, no. 3, pp. 141-149, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[26] Adán Godínez-Bautista et al., “Bio-Inspired Metaheuristics for Hyper-Parameters Tuning of Support Vector Machine Classifier,” *Fuzzy Logic Augmentations of Neural and Optimization Algorithms: Theoretical Aspects and Real Application*, pp. 115-130, 2018. [CrossRef] [Google Scholar] [Publisher Link]