Cost Efficient Stock Using Forecasting with Enhanced LightGBM and Optuna

Aditya Saxena[0000-0003-2204-9035]
BITS Pilani
Department of Computer Science & Engineering
f20190089@dubai.bitspilani.ac.in

Dr Tamizharasan Periyasamy[0000-0002-4932-5127]
BITS Pilani
Department of Computer Science & Engineering
tamizharasan@dubai.bitspilani.ac.in

Abstract—The use of machine learning in stock market prediction has garnered significant interest. This paper introduces a novel approach for short-term stock investment utilizing an optimized LightGBM model with a focus on cost awareness. Here, cost awareness refers to an enhanced sensitivity to false-positive errors, or 'fake chances,' which can help reduce investment costs. The primary contribution of this research is the development of an investment strategy for stock price prediction. Following the principles of short-term investment and contemporary quantitative investment research, we selected a series of technical indicators that align with the study's requirements to enhance the reliability of the prediction outcomes. We introduce the concept of cost awareness to boost prediction accuracy. By comparing our results with previous studies, we demonstrated that the performance of the optimized cost-aware model shows significant improvement. We also evaluated and compared the results with XGBoost and Random Forest, concluding that LightGBM excels in prediction accuracy, profitability, and risk management.

I. Introduction

Stock market prediction is widely considered one of the most challenging topics, attracting substantial interest from both investors and academia. Over the past decade, machine learning models such as multilayer neural networks (MLP)[1], recurrent neural networks (RNN)[2], and other approaches have been successfully applied to stock market prediction, yielding relatively good results[3][4]. However, this article introduces the concept of cost awareness from the perspective of financial risk awareness and develops a stock price prediction model for short-term investment using LightGBM (Light Gradient Boosting Machine).

Generally speaking, this study encompasses four main stages: feature engineering, hyper-parameter optimization, cost awareness adjustment, and model effect evaluation. During the feature engineering stage, we utilized data from the main board trading market of the Shanghai Stock Exchange, selecting over 1,500 stocks from 2010 to 2019 as samples. We generated 49 features from time series indicators, technical indicators, and OHLC indicators. In the feature selection process, we applied four methods to eliminate features with missing values, unique values, high correlation, and low importance. In the hyper-parameter optimization stage, we employed the Optuna hyperparameter framework [5] for parameter optimization using the time series split cross-validation method [6]. This article emphasizes preventing overfitting and focuses on limiting the size of the tree model during the hyper-parameter optimization process.

In the cost awareness adjustment stage, we aimed to integrate the cost awareness concept with LightGBM to enhance the model's prediction accuracy. The significance of cost awareness lies in increasing the model's sensitivity to false-positive errors, thereby reducing the occurrence of misjudging 'risks' as 'chances' in predictions. After constructing cost metrics, we used Optuna to optimize the 'scale pos weight' parameter to lower the probability of false positives and reduce investment losses due to misjudgments. Experimental results demonstrated that incorporating cost awareness effectively enhances model performance and profitability.

For model effect evaluation, we assessed model performance, profitability, and risk indicators to gauge the effectiveness of the predictions. Additionally, we compared LightGBM with XGBoost[7], Random Forest[8], and other decision tree-based algorithms.

Ultimately, this article demonstrates that the optimized LightGBM model based on cost awareness achieves higher prediction accuracy and maintains a low-risk index while ensuring a high rate of return.

II. Model and Methodology

A. LightGBM

LightGBM (Light Gradient Boosting Machine) is a framework that implements the GBDT (Gradient Boosting Decision Tree) algorithm. Unlike other GBDT models, LightGBM employs a unique method for calculating gain variation, which accounts for both weak and strong learners (small and large gradients, g(x)). Training instances are ordered in descending sequence based on the absolute value of their gradients.

First, a% of the instances with the largest gradients are selected to form subset A. From the remaining (1 - a)% of instances with smaller gradients, a subset B of size b |Ac| is randomly formed[9]. The instance is then split based on the estimated variance gain Vj(d) from the combined subset A ∪ B.

$$V_j * (d) = \frac{1}{n}\left(\frac{(\sum_{x_i\in A_l} g(x) + \frac{1-a}{b}\sum_{x_i\in B_l} g(x))^2}{n_l^j(d)} + \frac{(\sum_{x_i\in A_r} g(x) + \frac{1-a}{b}\sum_{x_i\in B_r} g(x))^2}{n_r^j(d)}\right)$$

where $A_l = fx_i$ 2 A : x_{ij} dg, $A_r = fx_i$ 2 A : $x_{ij} >$ dg, $B_l = fx_i$ 2 B : x_{ij} dg, $B_r = fx_i$ 2 B : $x_{ij} >$ dg, d is the point in the data where the split is calculated to find the best gain variance, and the coefficient $\frac{1a}{b}$ is used to normalize the gradient sum over B back to the size of A^c.

LightGBM supports efficient parallel training and offers advantages such as accelerated training speed, low memory consumption, higher accuracy, and fast processing of big data.

- Histogram Algorithm [10]

  To speed up the training process and reduce memory usage, this algorithm segments continuous feature values into discrete bins. This significantly reduces the cost of calculating the gain for each split.

- Leaf-wise (Best-first) Tree Growth [11]

  This method automatically selects the leaf node with the largest delta loss to grow. Compared to the traditional level-wise algorithm used in other decision tree-based models, the leaf-wise algorithm can eliminate more losses when growing the same leaf, as illustrated in Fig.1.

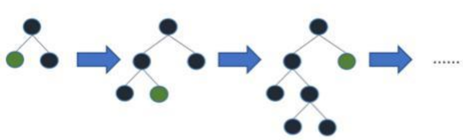

Fig. 1. Leaf-wise Tree Growth

- Gradient-based One-Side Sampling (GOSS) [10]

  Since samples with smaller gradients are often better trained, while those with larger gradients are more likely to be undertrained, GOSS excludes most samples with small gradients and uses the remaining samples to calculate the information gain. This algorithm reduces the data amount while maintaining accuracy, as shown in Algorithm1.

- Exclusive Feature Bundling [10]

  When handling high-dimensional feature data, LightGBM leverages the sparsity of high-dimensional feature values to merge mutually exclusive features into one bundle. This significantly reduces the amount of computation and enhances operational efficiency.

Algorithm 1 Gradient-based One-Side Sampling

```
Input:    : training data, I: iterations
Input: a: sampling ratio of large gradient data
Input: b: sampling ratio of small gradient data
Input: loss: loss function, L: weak learner
  models     fg; f act     1a/b
  topN     a  len( ); randN          b  len( )
  for i = 1 to I do
    preds      models:predict( )
    g   loss( ; preds); w          f1; 1; :::g
    sorted      GetSortedIndices(abs(g))
    topSet      sorted[1 : topN]
    randSet      RandomP ick(sorted[topN : len( )],
    randN)
    usedSet      topSet + randSet
    w[randSet]      = f act == the weights f or the
    small grandient data
    newM odel      L( [usedSet]; g  [usedSet],
    w[usedSet])
    models:append(newM odel)
  end for
```

B. Investment method

Short-term investments [12], also known as temporary investments, are assets that can be easily converted to cash, typically within five years. It is common for short-term investments to be sold or converted to cash in a short period, usually not exceeding 12 months.

The goal of short-term stock investment is to maintain asset liquidity while generating investment income. Based on this principle, our maximum holding period is set to five trading days. This means that all inventory will be sold within five trading days.

Before initiating a transaction, we use the model to calculate the probability that each stock will rise on the next trading day. If this probability exceeds 60%, 20% of the account balance will be used to purchase these stocks. If the probability is lower, no action will be taken.

III. DATA DESCRIPTIONS AND FEATURE ENGINEERING

This section covers sample selection, input variables, and the selection of variables used in this research. Additionally, in line with our investment strategy, we convert this experiment into a binary classification problem. The method for labelling is also provided.

A. Data selection and process

In this study, we concentrated on the main board of the Shanghai Stock Exchange, selecting a total of 1,560 stocks from the period of 2010 to 2019 as samples. We applied Z-score normalization to the model, which represents the deviation of the original data from the mean (μ) in terms of standard deviations (σ).

$$Z = \frac{X - u}{\sigma}$$

B. Input variables and variables selection

In this experiment, we introduced three different types of variables, each contributing time series features. These include:

1. Time Series Features: Variables that describe time series, such as week and month.
2. OHLC Variables: Daily open, high, low, and close prices, which illustrate daily transactions.
3. Technical Indicators [13]: Commonly used for pattern recognition in stock trend forecasts, these indicators help the model capture or identify signals that lead to market trend changes. TABLE I provides a description of some of the technical indicators we selected.

Before inputting these variables into training, we applied four methods to remove the unexpected variables.

- Variables with High Percentage of Missing Values
  In this study, the threshold for missing values is set at 90%. If a variable contains more than 90% missing values, the entire column is deleted.
- Variables with Single Unique Value
  Variables with a single unique value do not contribute to the model and are therefore deleted.
- Highly Correlated Variables
  Using the Pearson correlation coefficient [14], we calculate the correlation between variables. Variables with a correlation greater than 90% are removed. Fig. 2 shows the results after distribution. The deeper the blue, the greater the correlation coefficient between variables.
- Variables Showing Low Importance
  We sort the model variables by importance and remove the less important ones.

C. Labeling methods

In this study, the trading strategy involves only two actions: buy and sell. The action determination is based on the difference between the next closing price and the current closing price. We define the label as follows: if the next closing price minus the current closing price is greater than 0, the label is 1; otherwise, it is 0. Consequently, this research can be classified as a binary classification problem.

TABLE I
DESCRIPTION OF TECHNICAL INDICATORS

Name	Definition
CMO	Chande Momentum Oscillato: calculates the difference between the sum of recent gains and the sum of recent losses and then divides the result by the sum of all price movement over the same period. CMO = 100 ((Su - Sd) (Su + Sd)). where: Su = Sum of the difference between the current close and previous close on up days for the specified period. Up days are days when the current close is greater than the previous close. Sd = Sum of the absolute value of the difference between the current close and the previous close on down days for the specified period. Down days are days when the current close is less than the previous close.
CCI	The Commodity Channel Index (CCI) is a technical indicator that measures the difference between the current price and the historical average price. CCI = (Typical Price - SMATP) (0.015 Mean Deviation). where: SMATP = Simple MA(20) applied to the Typical Price.
SAR	As known as the 'stop and reversal system,' SAR is a indicator that detect the price direction of an asset, decide how to distribute the attention the signal of price direction changing.
KAMA	Kaufman's Adaptive Moving Average (KAMA)is a moving average designed to account for market noise or volatility.
ADX	average directional index(ADX) An indicator representing the strength of a price trend.
MOM	Momentum can be regarded as the ratio of stock price changes over a period of time. MOM = Price - Price of n periods ago
ROC	As know as price rate of change, measures the scale of change in price between the current price and the price a certain historical price. ROC = ((ClosingP $rice_p$ ClosingP $rice_{pn}$) ClosingP $rice_{pn}$) 100: where: ClosingP $rice_p$=Closing price of most recent period. ClosingP $rice_{pn}$ =Closing price n periods before most recent period.

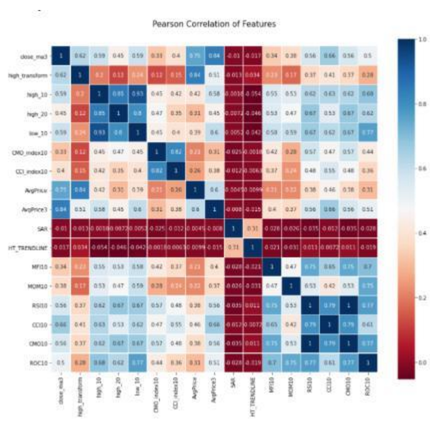

Fig. 2. Pearson Correlation of features

IV. Hyperparameter optimization

The parameters that define the attributes of the model or the training process are called hyperparameters [15]. To enhance the accuracy of the model, this study applied the Optuna optimization framework to determine a set of hyperparameters that deliver the best performance based on time series cross-validation.

In the cross-validation process, the typical approach is to use an ‘n-year sliding window’ to define the training set. However, determining the optimal value of n is challenging for time series problems. Therefore, in this study, we considered three training sets containing data from the periods 2010-2012, 2010-2015, and 2010-2018. In all cases, the test set comprises data from the 'next year' (i.e., 2013, 2016, and 2019), as illustrated in Fig. 3.

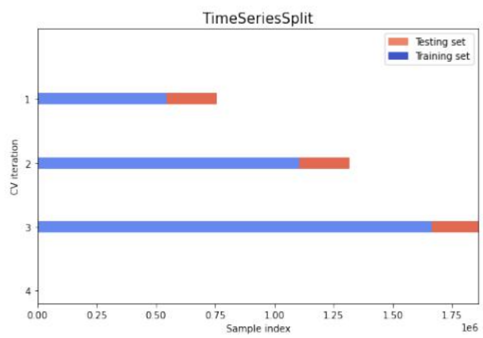

Fig. 3. Time series cross-validation

Optuna is an open-source hyperparameter optimization framework. It can automatically identify the best set of hyperparameters that yield the highest performance using various samplers, such as GridSampler, RandomSampler, and TPESampler.

TABLE II describes the parameters adjusted in this article.

TABLE II
DESCRIPTION OF THE PARAMETERS

Parameter	Value	Implication
lambda _l1	8.52	L1 regularization, prevent over fitting
lambda _l2	1.23e-05	L2 regularization, prevent over fitting
num leaves	143	number of leaves for one tree
feature_fraction	0.9	randomly select a subset of features on each iteration
bagging _fraction	1.0	select 100% data on bagging
bagging _freq	0	perform bagging in every iteration
min child samples	50	The minimum number of features on a leaf

V. COST AWARENESS

Traditionally, a stock trend prediction model is evaluated as a standard binary classification model, using measures such as misclassification error, receiver operating characteristic (ROC), F_1 statistics. Most of these measures are extracted by using a confusion matrix as shown in TABLE III.

From this table several statistics can be extracted. In particular:

Accuracy = (TP + TN) / (TP + TN + FP + FN)
Recall = TP / (TP + FN)
Precision = TP / (TP + FP)
F_1 = 2 P * R / (P + R):

Where P and R are the Precision and Recall, respectively.

TABLE III
CLASSIFICATION CONFUSION MATRIX

	predict negative	predict positive
actual negative	True Negative(TN)	False Positive(FP)
actual positive	False Negative(FN)	True Positive(TP)

However, these evaluation metrics may not be suitable for stock prediction models because they assume that the cost of a false-positive error is equal to that of a false-negative error. This assumption is flawed, as identifying a 'fake chance' (false positive) leads to a direct revenue loss, while missing a profitable opportunity (false negative) result in a lost chance for profit but does not reduce capital. Therefore, we introduced cost awareness in this study to not only improve the model's prediction accuracy but also to make it more cautious about false-positive errors.

The basic idea of cost awareness is to introduce a bias that makes the model more sensitive to false positives, thereby reducing their occurrence. To calculate actual costs during the model testing process, we propose a cost matrix that includes example-dependent financial costs.

A cost matrix, as shown in TABLE IV, has a specific structure for two classes. In this case, we assume the costs of true positives (TP) and true negatives (TN) are zero.The cost of FP is fp_Amt_i while the cost of FN is fp_Amt_i. The cost matrix calculation process is shown in Algorithm 2.

TABLE IV
BINARY CLASSIFICATION COST MATRIX

	predict negative	predict positive
actual negative	C(0,0) = 0	C(1,0) = fp_Amt_i
actual positive	C(0,1) = fn_Amt_i	C(1,1) = 0

Using the cost matrix, the total cost can be calculated as the sum of all individual costs.

This measure evaluates the sum of the cost for m transactions, where fp_Amt_i and fn_Amt_i are the FP and FN cost

Algorithm 2 Function to calculate the cost matrix

Input: test df: array of shape = [n samples]
Input: money init: amount invested in each stock
Output: cost mat: array-like of shape = [n samples, 4] init
buy rate; sell rate; stamp duty
money = money init
cost df = test df
for all (i; row) 2 cost df do
f p rate = f abs(row[buy price] row[sell price])
f n rate = f p rate
tran num = (money=row[buy price])==100 buy money = tran num row[buy price] sell money = tran num row[sell price]
service change = buy money buy rate + sell money sell rate
stamp duty = stamp duty sell money
f p Amt[i] = f p rate tran num+service change+ stamp duty
f n Amt[i] = f n rate tran num service change stamp duty
end for
cost mat[:; 0] = f p Amt
cost mat[:; 1] = f n Amt
cost mat[:; 2] = 0:0
cost mat[:; 3] = 0:0
return cost mat

In the cost matrix, the variables y_i and p_i represent the actual and predicted labels, respectively.

To determine loss, we use the cost matrix from Eq.(1) to calculate the loss for each sample and sum the total loss. To make the model more sensitive to false positives, we employed Optuna and the cost measure to adjust the 'scale pos weight' parameter. By tweaking this weight, the model prioritizes minimizing false positives and, consequently, the overall cost loss.

To verify the efficacy of cost awareness, we compared the model's performance with only cross-validation hyperparameter optimization. Results indicate that cost awareness enhances both predictive performance and profitability, as detailed in TABLE V.

According to the results, cost awareness can improve the predictive per-formance and profitability of the model, as shown in TABLE V.

$$F_{0:5} = (1 + 0:5^2)\ P\ \ R=(0:5^2\ \ P + R):$$

Where P and R are the precision and recall, respectively. Rate of return:

The formula to calculate the rate of return is:

ROR = ((C I)=I) 100

Where C represent current value, I represent initial value.

Annualized rate of return:

An annualized rate of return is calculated as the equivalent annual return an investor receives over a given period.

$$AP = ((P + G)=P)^{\frac{1}{n}} 1$$

Where P represent principal, or initial investment. G represent gains or losses. n represent number of years.

Benchmark return is based on the Shanghai stock market

TABLE V
COMPARISON OF THE EXPERIMENTAL RESULTS

	Optuna CV	Cost awareness
Training set	2010-01-01 to 2019-01-01	
Validation set	2019-01-02 to 2020-01-01	
AUC	0.5964	0.6094
Precision	0.5671	0.5865
$F_{0:5}$	0.5675	0.5794
Rate of return	0.8953	1.5193
Annualized return	0.9062	1.5403
Benchmark return	0.2212	0.2212

VI. PERFORMANCE AND MEASUREMENT

This section examines three key areas: predictive accuracy, profitability, and risk management. We evaluated these metrics across different models (XGBoost, Random Forest, and LightGBM) to determine the best-performing one. The Shanghai Exchange Index served as our benchmark, indicating the minimum acceptable return for any investment.

A. Predictive accuracy

As illustrated in TABLE VI, LightGBM outperforms XGBoost and Random Forest with an AUC of 60.94%, Precision of 58.65%, and F0.5 score of 57.94%.

For assessing prediction accuracy, we focused on the F0.5 score, AUC, and Precision. We chose the F0.5 score over F1 because our cost matrix prioritizes precision over recall.

B. Profitability performance

In this section, we evaluated the model's profitability using the rate of return, annualized return, and benchmark return.

As shown in TABLE VI, LightGBM achieved a rate of return of 151.93% and an annualized return of 154.03%, both surpassing XGBoost and Random Forest, and significantly exceeding the benchmark return of 22.12%.

Figures 4 and 5 illustrate the benchmark application. During the testing period, LightGBM's return curve consistently outperformed the benchmark curve. Figure 5 highlights the cumulative return comparison between the benchmark and the optimized LightGBM, demonstrating that LightGBM's returns and cumulative returns are significantly higher than the benchmark.

TABLE VI
COMPARISON OF THE EXPERIMENTAL RESULTS

Model type	XGBoost	RandomForest	LightGBM
Optuna optimize		Optuna_Costscore	
Training set		2010-01-01 to 2019-01-01	
Validation set		2019-01-02 to 2020-01-01	
AUC	0.5815	0.5799	0.6094
Precision	0.5670	0.5611	0.5865
$F_{0:5}$	0.5519	0.5504	0.5794
Rate of return	0.8497	0.5370	1.5193
Max drawdown	0.0530	0.0586	0.0899
Annualized return	0.8599	0.5370	1.5403
Benchmark return	0.2212	0.2212	0.2212
sharpe ratio	4.79	3.58	6.14
sortino ratio	163.01	146.68	174.28

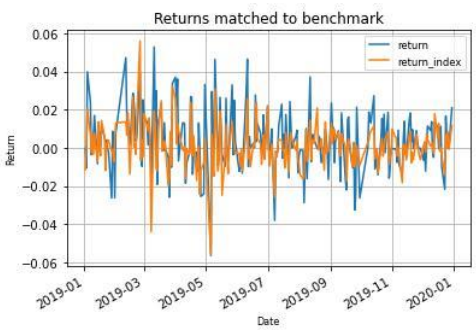

Fig. 4. Returns matched to benchmark

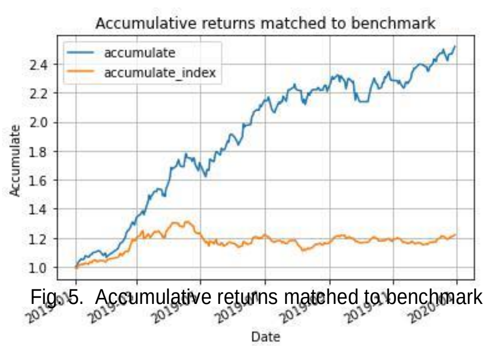

Fig. 5. Accumulative returns matched to benchmark

C. Risk control performance

For risk control, we utilized the Sharpe ratio, Sortino ratio, and maximum drawdown, which are standard indicators for measuring risk-adjusted returns considering both upside and downside volatility.

The Sharpe ratio [16] [17] assesses the excess return per unit of total risk. A positive Sharpe ratio indicates that the return exceeds the volatility risk, while a negative ratio suggests that operational risk outweighs the return. This ratio measures the return on investment relative to the risk taken, with higher values indicating better performance.

The Sortino ratio [18] [19] differentiates between positive and negative volatility by using the downside standard deviation instead of the total standard deviation. A higher Sortino ratio indicates that the model achieves higher excess returns for each unit of downside risk taken.

The maximum drawdown rate [20] quantifies the peak-to-trough decline in value over a specified period. For quantitative investment models, this metric is crucial as it is often more significant than volatility in assessing risk.

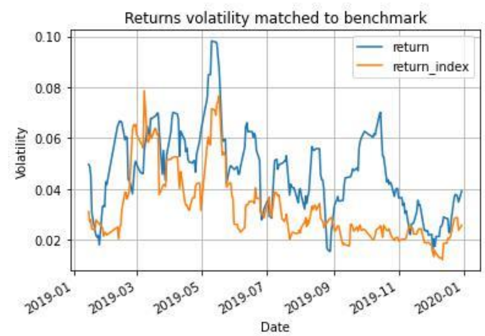

Fig. 6. Returns volatility matched to benchmark

As shown in TABLE VI, LightGBM achieves positive scores for the Sharpe Ratio (6.14) and Sortino Ratio (174.28), the highest among the three models. This indicates that the model maintains profitability even under downside risk. Figure 6 demonstrates that LightGBM outperforms the benchmark even during negative volatility. In terms of maximum drawdown, the three models show similar values. Overall, the optimized LightGBM model excels in risk control and demonstrates strong reliability in quantitative investment.

- Sharpe ratio:

SharpeRatio = pRp - Rf
where:
Rp = return of portfolio

Rf = risk-free rate.

p = standard deviation of the portfolio’s excess return

- Sortino ratio:

  SortinoRatio = dRp - rF

  where:

  Rp = Actual or expected portfolio return

  rF = Risk-free rate.

  d = Standard deviation of the downside
- Maximum drawdown:

  Maximum drawdown = (Trough Value - Peak Value) / Peak Value

VII. CONCLUSION

This paper presents a stock forecasting framework based on the LightGBM model. The framework's steps are: 1) normalize the data; 2) use OHLC, technical, time series, and market indicators as input variables; 3) select features by removing those with unique values, excessive missing values, high correlation, or low importance. The key contribution is optimizing model accuracy by integrating cost awareness into the Optuna hyperparameter optimization framework. This approach enhances the model's ability to accurately identify false-positive errors, thereby improving both prediction reliability and profitability. Consequently, this framework establishes a potentially state-of-the-art stock prediction model.

In evaluating the model, we considered not only prediction accuracy and profitability but also risk mitigation. We used the Sortino ratio, Sharpe ratio, and maximum drawdown rate to assess risk resistance. Moving forward, we plan to explore further optimization of the model with additional technical indicators and improve accuracy by replacing the current method with a block-based time-series validation concept.

REFERENCES

[1] D. S. Palmer, N. M. O’Boyle, R. C. Glen, and J. B. O. Mitchell, “Random forest models to predict aqueous solubility,” J. Chem. Inf. Model., vol. 47, no. 1, pp. 150–158, 2007.

[2] C. Tang, N. Luktarhan, and Y. Zhao, “An efficient intrusion detection method based on lightgbm and autoencoder,” Symmetry, vol. 12, no. 9, p. 1458, 2020.

[3] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” in Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 3146–3154.

[4] H. Zeng, C. Yang, H. Zhang, Z. Wu, J. Zhang, G. Dai, F. Babiloni, and W. Kong, “A lightgbm-based EEG analysis method for driver mental states classification,” Comput. Intell. Neurosci., vol. 2019, pp. 3 761 203:1–3 761 203:11, 2019.

[5] P. Chuhan, G. Perez´-Quiros,´ and H. A. Popper, International Capital Flows: Do Short-Term Investment and Direct Investment Differ?, ser. Policy research working paper. Washington, DC: World Bank, Internat. Economics Dep., Development Data Group [u.a.], 1996, no. 1669.

[6] “Forecasting the equity risk premium,” Hanover, Md, pp. 1772–1791, 2014.

[7] I. Cohen, Y. Huang, J. Chen, and J. Benesty, Pearson Correlation Coefficient: Noise reduction in speech processing. Springer Berlin Heidelberg, 2009.

[8] D. MacKay, “Hyperparameters: Optimize, or integrate out?” 01 1996.

[9] W. Sharpe, “The sharpe ratio,” The Journal of Portfolio Management, vol. 21, pp. 49–58, 10 1994.

[10] C. Israelsen, “A refinement to the sharpe ratio and information ratio,” Journal of Asset Management, vol. 5, pp. 423–427, 04 2005.

[11] L. D. Capitani, “Interval estimation for the sortino ratio and the omega ratio,” vol. 43, pp. 1385–1429, 2014.

[12] V. Mohan, J. G. Singh, and W. Ongsakul, “Sortino ratio based portfolio optimization considering evs and renewable energy in microgrid power market.” Chicago, IL, USA: IEEE, 2017, pp. 1–1.

[13] R. Leal and B. Mendes, “Maximum drawdown,” The Journal of Alter-native Investments, vol. 7, pp. 83–91, 03 2005.