Proceedings of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 761–766 DOI: 10.15439/2024F2428 ISSN 2300-5963 ACSIS, Vol. 39

Exploring Stability and Performance of hybrid Gradient Boosting Classification and Regression Models in Sectors Stock Trend Prediction: A Tale of Preliminary Success and Final Challenge

Ming Liu, Ling Cen, Dymitr Ruta
EBTIC, Khalifa University, UAE
{liu.ming,cen.ling,dymitr.ruta}@ku.ac.ae

Quang Hieu Vu
GREENFEED, Vietnam
hieu.vq@greenfeed.com.vn

***Abstract*—In the dynamic field of financial analytics, the ability to predict stock market trends is crucial for effective trading strategies, which is the task for FedCSIS 2024 Data Science Challenge: Predicting Stock Trends. This paper presents a comprehensive study on the use of hybrid gradient boosting models, incorporating both classification and regression approaches, to forecast stock trends across different sectors of the S&P 500. Utilizing a rich dataset comprising key financial indicators for 300 companies over a decade, our research aims to unravel the complexities of sector-specific trend predictions. The model leverages 58 financial indicators per company, along with their annual change metrics, to predict the future stock movements. In the preliminary phase of the competition, our hybrid model demonstrated promising results, achieving the lowest weighted error of 0.5941 among competitors. However, despite the initial success, the final phase of the model evaluation revealed a significant performance decline with the error rising above 0.84. This discrepancy highlights potential issues in model stability and preliminary performance when transitioning from a controlled to a truly unseen testing environment. This work not only underscores the complexities of predictive modeling in finance but also sets the stage for future research into creating more resilient AI-driven trading systems.**

***Index Terms*—Stock prediction, Machine Learning, Gradient Boosting Trees, Classification, Regression, Ensemble Learning.**

I. Introduction

Predicting stock market trends has been a critical challenge and a focal point of interest for investors, financial analysts, and researchers alike. The complexity and dynamic nature of financial markets make this task both intriguing and difficult. Over the decades, various traditional and computational methods[1][2] have been employed to forecast stock prices and trends, ranging from fundamental analysis of financial statements to technical analysis involving chart patterns and indicators. However, the advent of artificial intelligence (AI) and machine learning (ML) has transformed the landscape of financial forecasting, offering new insights and capabilities that were previously unattainable.

Machine learning models, unlike their traditional counterparts, can handle large volumes of unstructured data and quickly uncover complex internal patterns within them. Techniques such as regression trees [3], support vector machines [4], neural networks [5] and long short-term memory (LSTM) networks [6] have been widely adopted to predict stock prices and trends with varying degrees of success.

Hybrid models [7]-[11] that combine multiple AI techniques or integrate machine learning with traditional financial analysis have emerged as a powerful approach to improve prediction accuracy and robustness beyond the performance of any individual even the best model.

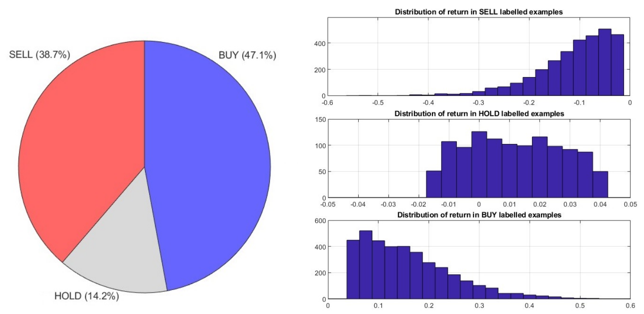
This paper aims to explore the application of hybrid gradient boosting models, which utilize both classification and regression techniques, to predict stock trends across different sectors of the S&P 500 index to address the task given in the FedCSIS 2024 Challenge[1]. We examine the preliminary success of these models in capturing the nuances of sector-specific trends and address the critical challenge of maintaining model stability and performance in diverse market conditions. With this research, we seek to contribute to the ongoing dialogue on improving the reliability and efficacy of AI-driven stock market predictions, providing valuable insights for both academic research and practical trading applications.

The structure of the remainder of this paper is outlined as follows. A concise description of the FedCSIS 2024 Challenge is provided in Section II. The analysis of data distribution and the methodologies used for feature engineering are discussed in Section III. This is followed by an explanation of the gradient boosting classification, regression and the ensemble hybrid models in Sections IV and V, respectively. The experimental results are detailed in Section VI followed by the lessons learnt from the preliminary success and the final challenge discussed in Section VII. Finally, the paper concludes with some closing thoughts and observations in Section VIII.

II. FedCSIS 2024 Challenge

The 2024 FedCSIS Data Science Challenge [12], focusing on Predicting Stock Trends, marks the 10th such event hosted by the FedCSIS Conference on Computer Science and Intelligence Systems[2]. This special anniversary edition centers

[1] https://knowledgepit.ml/fedcsis-2024-challenge/
[2] https://fedcsis.org/



(a) Class distribution (b) Return distribution in classes

Figure 1. Distribution of trading action labels in the training dataset (8000 examples) along with the distribution of return within corresponding classes

on financial data, with participants challenged to forecast the performance of selected stocks across various industry sectors. The competition enjoys sponsorship from Yettel.Bank (former Mobi Banka)[3], alongside the FedCSIS Conference itself.

III. Data analysis and feature engineering

The challenge requires participants to devise a reliable method for predicting trading actions (buy, sell, or hold), yet with offered also a continuous return in the training set it gives a freedom of deploying classification and regression paradigm for ML model construction as long as the final outputs are crisp trading actions.

A. Training dataset

The training dataset comprises 8,000 instances, presented in a tabular CSV file format. Each data instance corresponds to a specific event—namely, the announcement of a financial statement from one of the selected 300 companies. This dataset includes information about the company's sector, the values for 58 financial indicators and a 1-year (absolute) change for each indicator. The last two columns include the trading action ('Class' column), and the return performance following the announcement period (labelled as 'Perform' column). The distribution of classes within the training set along with the distribution of return within classes are illustrated in Figure 1.

B. Test dataset

The test dataset, which includes 2,000 instances, is formatted in the same tabular CSV file format and follows the same structure and naming conventions as the training data, however, it lacks the 'Class' and 'Perform' columns. It is important to note that not all testing set examples are necessarily in the future of all examples from the training set, which would have significantly limit the size of the testing set. However, as the organizers assured, the best care has been made to avoid temporal data leakage.

[3]https://www.yettelbank.rs/en/

C. Features

Both the training and test datasets include in total 117 features, which are divided into three categories: the Group feature that identifies one of the 11 company sectors, values for 58 critical financial indicators, and a 1-year (absolute) change for each of these indicators, as detailed in Table I.

Table I
Overview of the original features

Category	**Details**
Group	Financial, Industrial, Energy
	Information Technology, Consumer Staples
	Health Care, Utilities, Materials, Real Estate
	Consumer Discretionary, Communication Services
Indicators	I1, I2, ..., I58
1-Year Change	dI1, dI2, ..., dI58

D. Features aggregation and statistical features

In a search for additional features, while having limited expertise in the financial domain, we organized several raw features in groups based on their names' similarity, as outlined in Table II, and then attempted to aggregate correlated features to achieve more stable derived features. To complete this approach we have considered several aggregation operators that have been applied to all listed groups for each instance and thereby engineered many new candidate features with potential to enhance predictability of the targets.

Table II
Features aggregation

Category	**Aggregation**
Indicators	I1-I2, I3-I4, I7-I8-I9, I29-I30-I31
	I38-I39-I40, I41-I42, I45-I46
1-Year Change	dI1-dI2, dI3-dI4, dI7-dI8-dI9, dI29-dI30-dI31
	dI38-dI39-dI40, dI41-dI42, dI45-dI46

The following list summarizes all statistical aggregators applied to group-based engineering of numerical features:

- *minimum, maximum, mean*,
- *median, sum, standard deviation*

E. Return feature

Although the return *(Perform)* column, which represents the future return of stock price movement, is only present in the training dataset, and therefore cannot be directly used for the testing set, it plays important role in designing model options and making final financial decisions. We have analyzed the return distribution per class as illustrated in Figure 1, and jointly, as shown in Figure 2 and concluded that since the return is monotonic to the ordinal encoded trading class labels (-1:sell, 0:hold, 1:buy), building a classifier on (-1,0,1) classes is equivalent to building a regression model whose outputs can mapped back to discrete sell/hold/buy classes by simple threshold.

What is more due to the same monotonic alignment of the return and training labels as well as the fact that the model evaluation criterion uses symmetrical mis-classification cost

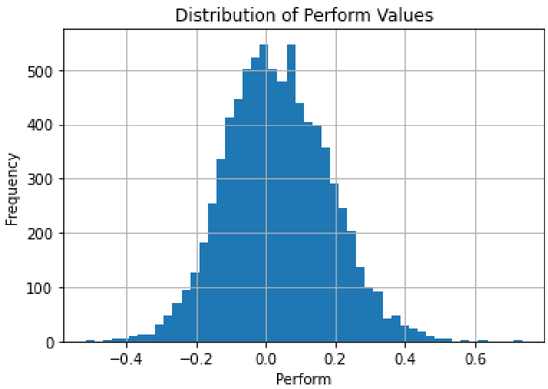


Figure 2. Distribution of Perform values

matrix as shown in Table III, it can be easily derived that cost-weighted mis-classification used as evaluation criterion in the competition is equivalent to the mean absolute error (MAE) of the regression model trained against discrete (-1,0,1) instead of continuous targets. This discovered property gives yet another design flexibility which could be useful when building hybrid supervised ML models.

Table III
MIS-CLASSIFICATION COST MATRIX

actual \predicted	**sell(-1)**	**hold(0)**	**buy(1)**
sell(-1)	0	1	2
hold(0)	1	0	1
buy(1)	2	1	0

IV. GRADIENT BOOSTING MODELS

Gradient Boosting Decision Trees (GBDT) algorithms have become a formidable and popular method in machine learning and data mining. By merging the advantages of decision trees with the technique of boosting, GBDT forms a predictive model that is both precise and robust. This methodology has been effectively utilized across several fields such as finance, healthcare, and online advertising [13][14].

For this challenge, we utilized two well-known GBDT algorithms, XGBoost and LightGBM, to develop an ensemble learning model aimed at predicting stock trends. Additionally, our team has a longstanding history of participating in data science competitions hosted by the KnowledgePit platform[4], employing GBDT-based algorithms for tasks in classification, regression, and other areas [15] - [32], achieving remarkable success. The versatility of Gradient Boosting Decision Trees in handling various data types, along with their capabilities in feature engineering and model hyper-parameter optimization, has consistently demonstrated their effectiveness in predictive modeling across multiple fields.

To manage the task of adjusting a multitude of parameters for each specific model, we employed a rapid and efficient rotational grid search technique, an extension of the conventional grid search method for hyper-parameter tuning [33]. Selecting the right values for hyper-parameters, including learning rate, tree depth, and regularization parameters, can markedly enhance the model's predictive accuracy and generalization capacity. And to improve the dependability of the optimal parameter configurations identified, we employed a Repeated Stratified 10-Fold cross-validation technique. This approach reduces the risk of inadvertently choosing configurations that perform exceptionally well by chance.

A. GBDT inspired Target Guided Binning (TGB)

In exploring the alternative and possibly diverse ways to build a reliable predictor we have also tested our target guided binning algorithm which has recently showed impressive predictive performance compared even to the leading gradient boosting models [16]. The simple model can be considered a combination of the Area Under the Curve (AUC) [34]-optimised 1-level singleton trees greedily merged to maximize any specific evaluation function with AUC set as a default. To attempt this model in a slightly more diverse setup we have trained it in the classification mode in two variants one using return sign as binary (buy/sell) class and another using the original buy and sell class examples only, i.e. completely excluding the hold class examples based on the rationale that the hold class examples may simply be confusing the binary classification with unstable border conditions and should therefore be trained on the strong positive (buy) and negative (sell) return examples only. We have trained both variants on all original 117 features only and achieved transformed monotonic risk features taking values from 1 (least risky - sell) to 10 (most risky - buy). We have then proceeded with constructing the model output using greedy selection of the binned features in turns maximizing the AUC at each next addition. With such selected binned features we have achieved the model returning the ordinal output of the sum of risk votes from each selected feature and the last task was to convert such output into sell, hold, buy (-1,0,1) discrete labels. Since the model output is monotonic with the return we had to simply identify optimal pair of thresholds that separate the continuous domain into three bounded regions corresponding to the sell, hold and buy class. This has been achieved exhaustively yet with the fast iterative algorithm of crawling thresholds from both ends towards the middle of output range until the cost-weighted mis-classification error is minimized. Noting down these thresholds completed the TGB model build and the same thresholds have been applied to classify the testing examples. Repetitive experiments and fine-tuning of this model resulted in rather consistent rule of the best model achieved when trained without hold-class examples and allocating bottom 20% of predictions to the sell-class, 40% of top predictions to the buy-class and and the remaining 40% in the middle to the hold-class. This result may somewhat come counter-intuitive to the original distribution of classes in which the hold class represents only 14% of the data, yet on balance simply reflects the cost function that penalizes double for for making opposite trade mistakes and hence placing the hold

[4]https://knowledgepit.ai/

class as a safer bet given big uncertainty while also reflecting a buy-class bias both of which are genuinely reflected in the investment environment.

V. Ensemble model

In constructing the final ensemble, we utilized two core gradient boosting models: XGBoost (XGB) and LightGBM (LGBM) trained in regression mode as well as an alternative regression or classification model (initially TGB model) that we have switched on or off throughout the competition depending on the evolving leader board performance feedback. To improve the models' ability to generalize, we implemented filtering techniques. These techniques aim to diversify the classifiers by creating multiple variants, which are trained on either the full training set or specific subsets of it. These variants are then deployed on the testing set, and their outputs are collectively aggregated to form the final prediction.

To further enhance diversity and seek improved predictive performance, we trained all baseline models on different feature subsets generated by our feature engineering engine. The primary distinction between these feature subsets was that the second set included a greater number of sparse columns obtained from an extensive application of one-hot-encoding to categorical features. This approach aimed to introduce more varied and complementary information for prediction.

Additionally, to explore further opportunities for enhancing performance, we implemented an extra stacked layer of simple linear regression. This layer was trained on the outputs from the baseline models. To seamlessly incorporate this stacking layer, we split the training data into two separate segments. The initial segment was used to develop the baseline models, and the latter segment was specifically dedicated to training the parameters of the linear regression model within the stacked layer.

Ultimately, we combined the outputs from each individual model with those from the linear regression-based stacking layer by averaging them. The architecture of this final ensemble, depicted as a flow chart that illustrates the structure, is presented in Figure 3.

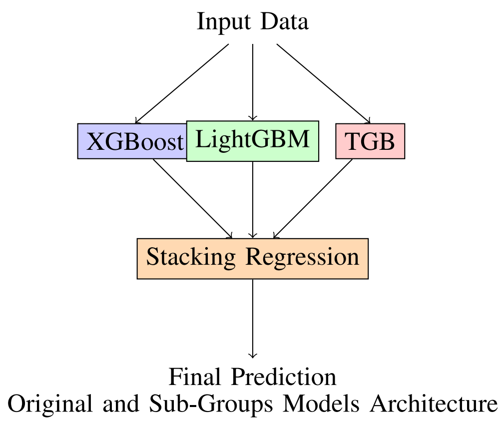


Figure 3. Stacking model architecture

VI. Experimental results

During the competition, we utilized sklearn packages, xgboost, lightgbm, and Python3 Jupyter Notebook[5] operating on a Windows Server Virtual Machine equipped with 128G RAM and an Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz with 2 processors for running simulations. Our approach included extensive feature aggregation and various combinations of features, where we selectively removed or filtered out specific columns as detailed in the Table IV below.

Table IV
Version of feature sets

Version	Number of Features	Remarks
V1	117	Original features
V2	127	Add combined features as in Table II
V3	187	Add statistical features
V4	237	Add Top 50 importance features

The various feature sets and their respective effects on the performance of individual models, particularly within the constrained and sparse training and testing datasets, are summarized in the Table V below.

Table V
Features and model performance

Version	LGBM	XGB	TGB
V1	0.8267	0.8218	0.8291
V2	0.7673	0.7970	0.8262
V3	0.7475	0.7624	0.8231
V4	0.7376	0.7178	0.8241

Interestingly TGB model did not seem to gain from additionally engineered features and hence was dropped from the ensemble in the subsequent hybrid model versions.

Throughout the competition, many parameter variations showed robust performance. To optimize these parameters, techniques like Grid Search was employed, which iterative explore the hyper-parameter space to identify the best combination based on cross-validated performance metrics. Our highest individual model scores were achieved using particular model settings, as detailed in the Table VI below.

Table VI
Optimized individual model parameters

Model	Iterations	Learning Rate	Tree Depth
LGBM	1000	0.08	3
XGB	2000	0.3	6

To accommodate the variety of company sectors indicated by the Group feature, we experimented with dividing the datasets into 11 subgroup models, applying distinct value sets to these subgroups. This approach enhanced the evaluation performance, as outlined in the Table VII below.

The final predictions were derived by averaging the results from both the stacking models and the varied individual baseline models through an ensemble technique. This method

[5] https://jupyter.org/

Table VII
FEATURES AND 11 SUB-GROUP MODELS PERFORMANCE

Version	LGBM	XGB
V1	0.7475	0.7624
V2	0.7079	0.7327
V3	0.6980	0.7178
V4	0.6683	0.6832

achieved a preliminary score of 0.5941, ranking as the top preliminary score among the 10% preliminary test datasets.

VII. PRELIMINARY SUCCESS AND FINAL CHALLENGE

Despite the initial success, the final evaluation phase of the model showed a decrease in performance, recording a score of only 0.841500. This decline underscores possible challenges in model stability and generalization as it moves from a controlled testing environment to a more diverse one.

After the competition organizers released the complete test datasets, we conducted a thorough evaluation of all the models. Upon analysis, we realized that the narrative of initial success followed by a subsequent drop in performance during the final phase could be attributed to two primary factors. First, there might have been an issue of overfitting, where models tuned to excel on preliminary data failed to generalize effectively to the broader dataset. Second, the variation in the test dataset's characteristics compared to the training set could have exposed weaknesses in the models' adaptability. These issues highlight the importance of robust model validation strategies and underscore the need for models that can maintain consistency across different data subsets.

To provide a more detailed explanation, the major factor concerns the sub-group models, which seemed logical given that different financial sectors may exhibit distinct financial patterns. However, a significant challenge arose due to the limited size of the preliminary test dataset, which constituted only 10% of the total data. When this data was further subdivided by Group for the subgroup models, each individual subgroup ended up with an even smaller portion of data for training. This scant amount of data likely resulted in models that were under-fitted and unstable. Such models struggle to capture the complexity and variability of their respective sectors, leading to performance issues when faced with a broader and potentially more diverse set of test data. This situation underscores the critical importance of having a sufficiently large and representative training dataset to ensure robust model training and stability.

The next concern is related to the error cost matrix[6] as already presented in Table III, which is used as follows to compute the final error used as evaluation in the competition:

$$\text{err} = \frac{\text{confusion\_matrix(preds, gt)} \cdot \text{cost\_matrix}}{\text{length(gt)}}$$

The other decline factor relates to the adjustments we made through post-processing techniques, which were guided by the error cost matrix. In an attempt to refine the model's performance, we utilized the cost matrix to prioritize certain types of errors over others, aligning the model's output with specific financial implications associated with different types of prediction errors. This strategy involved adjusting the model's predictions to minimize the financial risk as quantified by the cost matrix. While this approach can effectively optimize the model for scenarios represented within the training data, it may inadvertently lead to a lack of generalization on new data sets if the error characteristics differ. This reliance on post-processing based on the cost matrix can potentially skew the model's ability to predict accurately in diverse real-world scenarios, as the adjustments might not align well with the actual error distribution in unseen data.

Accordingly, we undertook some adjustments to the distribution of the labels in our dataset. Although for the TGB model we have observed the optimal predicted class distribution to be (20%,40%,40%), we have observed that for gradient boosting models, and the whole hybrid design, squashing the prevalence of hold-0 while elevating the remaining buy and sell classes seems to elevate the performance. This maneuver originating from the model fine-tuning seemed somehow to fit the preliminary set really well as we have achieved unprecedented gains in the preliminary set evaluations clearly capturing the buy and sell class examples well from that limited set. In the competitive conditions when the thorough cross-validation testing can by costly and time consuming to vet such quick new post-processing discoveries, such last minute leader board score following could lead to a classic overfitting trap that could quickly compromise generalization ability of otherwise really good ML model. Trusting the representative nature of the preliminary set we have followed through with the model adjustments that have placed our score well ahead of competition and clearly must have optimized for specific characteristics of the preliminary data, at the cost of the model's ability to perform consistently across a more comprehensive dataset. One could wonder, regardless, how is it possible to achieve such a high score for the preliminary set that was still hidden for the participants? A possible explanation could be that our complex hybrid model with several layers of grouping may have discovered data snooping leads that connected data points of the same stock from different moments in time. Whatever the reason, consequently these alterations contributed to the preliminary success yet ultimate demise of our model performance in the final evaluation phase. This experience clearly highlights the critical importance of the proper model cross-validation however costly and complex it may be in the case of hybrid, ensemble and stacked models.

VIII. CONCLUSIONS

In this competition, we aimed to improve the predictive capabilities of the already effective models in the gradient boosting family, specifically XGBoost and LightGBM. To meet this challenge, we utilized various GBDT techniques with diverse ensemble strategies, achieving enhanced performance by aggregating a broader array of model variants. Further-

[6]https://knowledgepit.ml/fedcsis-2024-challenge/

more, we implemented regression-based stacking and carefully selected the top-performing ensemble models, prioritizing a balance between performance and diversity to optimize results. This approach allowed us to refine and advance the effectiveness of our predictive models within the ensemble framework.

During the initial phase of the competition, our hybrid model showed promising results, securing a leading score of 0.5941, which placed us at the forefront among all competitors. However, this early success was not sustained in the final evaluation phase, where the model's performance fell to a score of 0.8415. This significant drop in performance underscores potential issues with the model's stability and its ability to generalize effectively across different testing environments, transitioning from a controlled setting to one that is more diverse and unpredictable. This experience highlights the inherent challenges in financial predictive modeling and paves the way for future research aimed at developing more robust and resilient AI-driven trading systems.

REFERENCES

[1] S. Yadav and K. P. Sharma, "Statistical Analysis and Forecasting Models for Stock Market," *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, Jalandhar, India, 2018, pp. 117-121, doi: 10.1109/ICSCCC.2018.8703324.

[2] J. Bagul, P. Warkhade, T. Gangwal and N. Mangaonkar, "ARIMA vs LSTM Algorithm – A Comparative Study Based on Stock Market Prediction," *2022 5th International Conference on Advances in Science and Technology (ICAST)*, Mumbai, India, 2022, pp. 49-53, doi: 10.1109/ICAST55766.2022.10039560.

[3] R. Karim, M. K. Alam and M. R. Hossain, "Stock Market Analysis Using Linear Regression and Decision Tree Regression," *2021 1st International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, Sana'a, Yemen, 2021, pp. 1-6, doi: 10.1109/eSmarTA52612.2021.9515762.

[4] Z. Liu, Z. Dang and J. Yu, "Stock Price Prediction Model Based on RBF-SVM Algorithm," *2020 International Conference on Computer Engineering and Intelligent Control (ICCEIC)*, Chongqing, China, 2020, pp. 124-127, doi: 10.1109/ICCEIC51584.2020.00032.

[5] Y. Wei and V. Chaudhary, "The Directionality Function Defect of Performance Evaluation Method in Regression Neural Network for Stock Price Prediction," *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, Sydney, NSW, Australia, 2020, pp. 769-770, doi: 10.1109/DSAA49011.2020.00108.

[6] M. Faraz, H. Khaloozadeh and M. Abbasi, "Stock Market Prediction-by-Prediction Based on Autoencoder Long Short-Term Memory Networks," *2020 28th Iranian Conference on Electrical Engineering (ICEE), Tabriz*, Iran, 2020, pp. 1-5, doi: 10.1109/ICEE50131.2020.9261055.

[7] J. Creighton and F. H. Zulkernine, "Towards building a hybrid model for predicting stock indexes," *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, 2017, pp. 4128-4133, doi: 10.1109/BigData.2017.8258433.

[8] A. Durgapal and V. Vimal, "Prediction of Stock Price Using Statistical and Ensemble learning Models: A Comparative Study," *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical*, Electronics and Computer Engineering (UPCON), Dehradun, India, 2021, pp. 1-6, doi: 10.1109/UPCON52273.2021.9667644.

[9] P. K. Aithal, U. D. Acharya, M. Geetha, R. Sagar and R. Abraham, "A Comparative Study of Deep Neural Network and Statistical Models for Stock Price Prediction," *2022 3rd International Conference for Emerging Technology (INCET)*, Belgaum, India, 2022, pp. 1-5, doi: 10.1109/INCET54531.2022.9824487.

[10] X. Zheng, J. Cai and G. Zhang, "Stock Trend Prediction Based on ARIMA-LightGBM Hybrid Model," *2022 3rd Information Communication Technologies Conference (ICTC)*, Nanjing, China, 2022, pp. 227-231, doi: 10.1109/ICTC55111.2022.9778304.

[11] R. Jaiswal and B. Singh, "A Hybrid Convolutional Recurrent (CNN-GRU) Model for Stock Price Prediction," *2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT)*, Indore, India, 2022, pp. 299-304.

[12] Aleksandar M. Rakicevi, Pavle D. Milosevic, Ivana T. Dragovic, Ana M. Poledica, Milica M. Zukanovic, Andrzej Janusz, Dominik Slezak: Predicting Stock Trends Using Common Financial Indicators: FedCSIS 2024 Data Science Challenge on KnowledgePit.ai Platform, *Proceedings of the 19th Conference on Computer Science and Intelligent Systems (FedCSIS)*, 2024.

[13] L. Mason, J. Baxter, P.L. Bartlett, and M. Frean. Boosting Algorithms as Gradient Descent In S.A. Solla and T.K. Leen and K. Müller. *Advances in Neural Inf. Processing Sys.* 12: 512–518, MIT Press, 1999.

[14] J.H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* 29(5): 1189-1232, 2001.

[15] M. Liu, L. Cen and D. Ruta, "Gradient Boosting Models for Cybersecurity Threat Detection with Aggregated Time Series Features," *2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS)*, Warsaw, Poland, 2023, pp. 1311-1315, doi: 10.15439/2023F4457.

[16] D. Ruta, M. Liu and L. Cen, "Beating Gradient Boosting: Target-Guided Binning for Massively Scalable Classification in Real-Time," *2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS)*, Warsaw, Poland, 2023, pp. 1301-1306, doi: 10.15439/2023F7166.

[17] D. Ruta, M. Liu, L. Cen. Feature Engineering for Predicting Frags in Tactical Games. *Proc. Int. Conf. 2023 IEEE International Conference on Multimedia and Expo*, 2023.

[18] D. Ruta, M. Liu, L. Cen and Q. Hieu Vu. Diversified gradient boosting ensembles for prediction of the cost of forwarding contracts. *Proc. Int. 17th Conf. on Computer Science and Intelligence Systems*, 2022.

[19] Q. Hieu Vu, L. Cen, D. Ruta and M. Liu. Key Factors to Consider when Predicting the Costs of Forwarding Contracts. *Proc. Int. Conf. 2022 17th Conf. on Computer Science and Intelligence Systems*, 2022.

[20] D. Ruta, L. Cen, M. Liu and Q. Hieu Vu. Automated feature engineering for prediction of victories in online computer games. *Proc. Int. Conf on Big Data*, 2021.

[21] Q. Hieu Vu, D. Ruta, L. Cen and M. Liu. A combination of general and specific models to predict victories in video games. *Proc. Int. Conf. on Big Data*, 2021.

[22] D. Ruta, L. Cen and Q. Hieu Vu. Deep Bi-Directional LSTM Networks for Device Workload Forecasting. *Proc. 15th Int. Conf. Comp. Science and Inf. Sys.*, 2020.

[23] L. Cen, D. Ruta and Q. Hieu Vu. Efficient Support Vector Regression with Reduced Training Data. *Proc. Fed. Conf. on Comp. Science and Inf. Sys.*, 2019.

[24] D. Ruta, L. Cen and Q. Hieu Vu. Greedy Incremental Support Vector Regression. *Proc. Fed. Conf. on Computer Science and Inf. Sys.*, 2019.

[25] Q. Hieu Vu, D. Ruta and L. Cen. Gradient boosting decision trees for cyber security threats detection based on network events logs. *Proc. IEEE Int. Conf. Big Data*, 2019.

[26] L. Cen, A. Ruta, D. Ruta and Q. Hieu Vu. Regression networks for robust win-rates predictions of AI gaming bots. *Int. Symp. Advances in AI and Apps (AAIA)*, 2018.

[27] Q. Hieu Vu, D. Ruta, A. Ruta and L. Cen. Predicting Win-rates of Hearthstone Decks: Models and Features that Won AAIA'2018 Data Mining Challenge. *Int. Symp. Advances in Artificial Intelligence and Apps (AAIA)*, 2018.

[28] L. Cen, D. Ruta and A. Ruta. Using Recommendations for Trade Returns Prediction with Machine Learning. *Int. Symp. on Methodologies for Intelligent Sys. (ISMIS)*, 2017.

[29] A. Ruta, D. Ruta and L. Cen. Algorithmic Daily Trading Based on Experts' Recommendations. *Int. Symp. on Methodologies for Intelligent Systems (ISMIS)*, 2017.

[30] Q. Hieu Vu, D. Ruta and L. Cen. An ensemble model with hierarchical decomposition and aggregation for highly scalable and robust classification. *12th Int. Symp. Advances in AI and Applications (AAIA)*, 2017.

[31] L. Cen and D. Ruta. A Map based Gender Prediction Model for Big E-Commerce Data. *The 3rd IEEE Int. Conf. on Smart Data*, 2017.

[32] D. Ruta and L. Cen. Self-Organized Predictor of Methane Concentration Warnings in Coal Mines. *Proc. Int. Joint Conf. Rough Sets, LNCS*, Springer, 2015.

[33] https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/.

[34] https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc/.