Communication Papers of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 185–190 DOI: 10.15439/2024F7856 ISSN 2300-5963 ACSIS, Vol. 41

FedCSIS 2024 Data Science Challenge: Predicting Stock Trends by a Multi-Dimensional Approach

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

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

***Abstract*—Predicting stock market trends is a challenge that is extremely difficult to solve, yet keeps captivating financial analysts, economists and small investors alike. Stock prices are very volatile, trends follow complex trajectories while the whole financial markets are marred by uncertainty and efficiency principles claiming its unpredictability comes from the fact that any useful evidence in the market is immediately discounted and priced in, such that the price actions of stocks resemble random walk. This very challenge has been proposed as an objective of the FedCSIS'2024 Competition concerned with prediction of optimal equity trade actions based on the established fundamental analysis indicators derived from financial statements and published reports. The dataset comprising thousands of such statements for 300 S&P 500-listed companies from 11 different sectors spanning a period of a decade has been made available along with the optimal trade action labels attached for the training part based on the future return. To address this challenge we have proposed a robust multidimensional model that leverages multiple supervised ML mechanisms to achieve alternative and diverse predictors that are eventually combined in an efficient ensemble to reach the final predictions. Our pragmatic approach vetted with the strict validation set complexity control achieved a very good generalization abilities and won the 2^{nd} place in the competition surpassing in the final evaluation very many competitive models that turned out to be massively overfitted.**

***Index Terms*—Stock trend prediction, Multi-dimensional approach, Classification, Regression, Ensemble, Stacking.**

I. Introduction

PREDICTING stock market trends is sometimes considered a task to predict the unpredictable. Efficient markets immediately exploit any emerging shred of useful evidence and leave the price actions to follow trajectories that resemble random walks. Inherent volatility, complexity and countless of subtle possible impacts make this task even more hopeless. However, there are certainly some market participants that seem to achieve a consistent risk-adjusted positive returns over long period of time. When considering how this could be possible we might take a look at various time resolutions and predictive horizons and it seems that indeed at the low frequency trading resolution of days and beyond there seems to be a lot of useful merit-based fundamental evidence that offers much more that random guess in relation to the company immediate future return. Several approaches have been typically adopted to address the stock prediction challenge, of which the most popular are:

- Technical Analysis: is an approach that predicts future stock movements by using statistical based methods such as Moving Average (MA), Relative Strength Index (RSI), Bollinger Bands (BB) etc, through the analysis of the stocks' historical price and volume that is typically extrapolated into the future.
- Fundamental Analysis: is an approach that relies on the analysis of the company's financial statements, earnings reports, and economic indicators to determine the stock value from which to predict the stock price direction.
- Machine Learning Models: is an advanced approach that employs supervised machine learning algorithms from a wide range of the traditional models such as decision trees and random forests to deep learning models such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) to predict future stock prices with models trained on historical data.
- Sentiment Analysis: is a different approach that assumes public sentiment have considerable impacts, and hence correlation with stock prices. As a result, the solution involves monitoring the sentiments from news and social media to identify likely stock price movements.

The 2024 FedCSIS Data Science Challenge[1], dedicated to Predicting Stock Trends, commemorates the 10th event hosted by the FedCSIS Conference on Computer Science and Intelligence Systems[2]. This special anniversary edition focuses on financial data, challenging participants to forecast the performance of selected stocks across different industry sectors. The competition is sponsored by Yettel.Bank (formerly known as Mobi Banka)[3] and the FedCSIS Conference itself.

In the FedCSIS 2024 competition [1], this challenge is revisited. The objective of the competition is to predict stock trends of companies across 11 different industrial sectors, based on the provided dataset containing 58 key financial indicators with annual changes of the 300 pre-selected S&P 500-listed companies from the last 10 years. The indicators are the classic measures used in the fundamental analysis to comprehensively capture the company's state, financial health and growth prospect and are derived from its financial statements and published financial reports. This set is matched with its annual change figures to capture the company's dynamics. This paper presents our solution that won the 2^{nd} place in the

[1] https://knowledgepit.ml/fedcsis-2024-challenge/
[2] https://fedcsis.org/
[3] https://www.yettelbank.rs/en/

competition based on final model evaluation on the unseen testing set. The prevailing idea of our solution is a multi-dimensional approach consisting of different machine learning models, each of which focuses on a separate dimension of the forecast, from which the results are combined to make a final decision. Specifically, our multi-dimensional approach consists of the following machine learning models:

- As the outcome of the challenge is to predict whether a stock should be bought, sold, or held, our first model is a classification model that classifies the stock into these three classes accordingly.
- As the classification model often mis-classifies examples along the inter-class boundary, we came up with a second model that represents the task as a regression problem and fine-tuned the thresholds to make a better separation between sell and hold as well as hold and buy.
- Our third model also aims to improve separation among classes of trading actions. This model is a combination of a binary classification buy-or-not model, which determines whether a stock should be bought or not, and a binary classification sell-or-not model, which tries to separate sell class from the rest.
- The final model incorporates the continuous return into consideration. The purpose of this model is to provide a better look at the performance dimension of stocks to make a final decision.

Details of each model along with the engineered features will be presented in the later parts of this paper. The rest of the paper is organized as follows. Section II presents Related Work. Section III introduces features engineered for the models. Section IV discusses the details of our approach, and its implementation into a final solution. Section V discusses parametric optimisation and fine-tuning carried out to maximize model performance. Finally, Section VI makes some concluding remarks.

II. Related Work

The use of computational techniques in stock market prediction has been explored extensively over the decades. Initial efforts primarily focused on statistical models like Auto Regressive Integrated Moving Average (ARIMA) [2], which were well-regarded for their predictive accuracy in linear data series. However, as financial market data complexity increased, these models became less sufficient.

With the rise of machine learning, researchers shifted towards models that could capture non-linear relationships in data. For instance, research in [3] demonstrated that Support Vector Machines (SVM) outperform traditional ARIMA models in forecasting stock prices. More comprehensive studies by [4] showed that Random Forests could effectively predict stock direction, providing better accuracy when combined with feature engineering techniques.

The advent of deep learning has introduced more sophisticated AI models like Long Short-Term Memory networks (LSTM) and Convolutional Neural Networks (CNNs), which are particularly adept at handling sequences and spatial structures in data. A pivotal study in [5] illustrated the superiority of LSTMs over traditional machine learning models in predicting stock market trends due to their ability to remember long-term dependencies.

Recent research has also explored the effectiveness of hybrid models that combine multiple AI techniques to enhance prediction accuracy. For example, in [6] developed a genetic algorithm assisted LSTM-CNN hybrid model integrating the LSTM with a complex event processing system to predict stock prices in real-time, showing an remarkable increase in prediction accuracy over using LSTM alone.

Despite these advancements, AI models for stock market prediction face significant challenges, primarily due to the noisy and non-stationary nature of financial data explored in [7]. Moreover, the problem of overfitting and the lack of transparency in deep learning models pose significant hurdles in their practical implementation.

The review of the literature underscores the transition from statistical to more complex AI models in stock market prediction, each offering improvements over previous methods. However, the field continues to face challenges such as model overfitting, data quality issues, and the need for model interpret-ability.

As a team we had actively been participating in the data science competitions [8] - [25] on the KnowledgePit platform[4], predominantly using GBDT-based algorithms for classification, regression, and other tasks and achieving high rankings and insightful experiences.

III. Feature Engineering

A. Dataset Description

The objective is to build an accurate method for predicting optimal trading actions (buy, sell, hold). The provided training data consists of 8,000 instances of fundamental financial data in a tabular CSV format. Each instance represents a financial statement announcement for one of the 300 pre-selected companies and includes information about the company's sector, values for 58 key financial indicators, 1-year (absolute) changes for each of these indicators, target class information (in the 'Class' column), and the return performance for a period following the announcement (in the 'Perform' column).

B. Target and Evaluation

The test data, consisting of 2,000 instances, is also provided in the same tabular CSV-formatted file and follows the same naming scheme as the training data but does not include the 'Class' and 'Perform' columns.

Participants were expected to submit their solutions to the online evaluation system as a text file containing exactly 2,000 lines with predictions for the test instances. Each line in the submission must contain a single number from the set 1, 0, -1, representing the predicted trading action for the event: buy,

[4]https://knowledgepit.ai/

hold, sell, and exactly matching the order of examples in the testing set.

Submissions are be assessed based on the cost-weighted average error, using the cost matrix as shown in Table I.

Table I
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

The final model evaluation error (err) used throughout the competition is calculated using the following formula:

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

C. Feature Grouping

Feature engineering over unknown dataset and without financial market expertise poses a real challenge. As a result, instead of leveraging domain knowledge, we attempted to randomly generate new features by blind brute-force combinations and aggregations of the original features (indicators), followed with evaluation of the importance of the new features during training to determine its value for the predictive task in hand. Specifically, for each group of financial indicators that we picked, we generated new features using basic statistical aggregation operators within groups as follows:

- Total number of financial indicators with positive values, total number of financial indicators with negative values and the difference between these two results.
- Sum of all financial indicators with positive values, sum of all financial indicators with negative values and sum of values from all financial indicators.
- Std of all financial indicators with positive values, std of all financial indicators with negative values and std of values from all indicators.
- Median, minimum, and maximum values from all financial indicators, as well as total number of financial indicators having NULL or N/A values.

Even though we had planned to perform brute-force search for good features, we quickly realized that the exponentially exploding number of possible combinations even among the basic 58 indicators makes it impossible to complete the task. As a result, to limit the complexity of the search, we tried to group financial indicators based on their perceived semantic similarity (e.g., those having similar keywords in the name or similar meaning). This way, we significantly reduced the number of group-items that we needed to test and hence, unlike for the individual features, it was possible to search though all the combinations of the feature groups. Preliminary baseline experiments evaluating predictive value of the new features revealed that statistics computed over the following groups of original features achieved promising results:

- Group of the 58 key financial indicators and group of the 1-year (absolute) change for each of the 58 indicators.
- Groups the following combinations: ($I1$, $I2$), ($I3$, $I4$), ($I8$, $I9$, 10), ($I11$, $I12$, $I13$, $I14$, $I15$), ($I17$, $I18$, $I19$, $I20$), ($I13$, $I21$, $I23$, $I36$), ($I30$, $I31$, $I32$), ($I39$, $I40$, $I41$, $I42$, $I43$, $I44$), and ($I45$, $I46$, $I54$, $I55$)
- Group of the similar combinations of the 1-year (absolute) change for financial indicators in the groups listed in the point above. For example, there exist group ($dI1$, $dI2$), which is similar to the group ($I1$, $I2$) and group ($dI3$, $dI4$), which is similar to the group ($I3$, $I4$).

With the above approach, we ended up with almost 500 different features for our selection. By using both K-Best and Recursive Feature Elimination (RFE) methods, we received the final set of approximately 270 features coming from all the three groups listed above. There are several interesting observations noted during our feature selection process that were listed below:

- There is no group of features that significantly outperforms other group of features. Feature importance analysis revealed that the important values of features do not vary significantly.
- Even though we used approximately 270 features in our final model, we could have achieved a similar performance with fewer than 100 features.
- While we could achieve similar performance with a much smaller number of features, the performance with reduced set of features is much less stable with respect to changes (e.g., changes of training parameters, number of folds, etc), compared to the performance obtained from a large number of features. Consequently, we chose to keep a large number of features for our final model.

IV. A MULTI-DIMENSIONAL APPROACH

As discussed in the previous section, feature engineering was not our strength in this competition and following preliminary testing we did not expect significant performance breakthroughs in this domain. Instead, we believe that our diversified multi-dimensional approach to model constriction was the key that led us to the very good second position in the final evaluation. The main idea behind our multi-dimensional approach was to look at the predictive problem at hand from many different points of view (or different dimensions) and try to derive the alternative and diverse predictive solutions that could be effectively combined in the final stage. In the following subsections, we present different approaches (dimensions) that we have implemented for this competition.

A. A classification model

Our first approach is a classification model. This approach follows from the original problem statement of: classifying a stock into the three trade action classes: buy, hold or sell. The challenge encountered in this approach, however, is not only the necessity to deal with the 3-class classification problem but also classes imbalance with only 14.21% of hold class and much larger sell (38.68%) and even larger buy (47.11%) class as visually depicted in Figure 1.

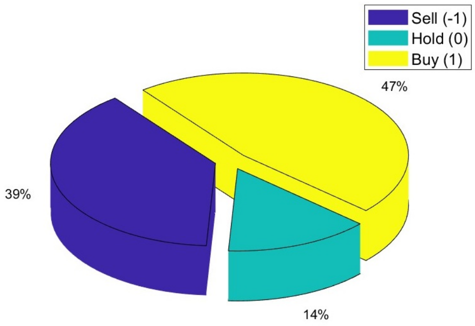

Figure 1. Distribution of trading action labels in the training dataset (8000 examples)

Here is a strategy we undertook to address the these challenges:

1) We apply hold-class up-sampling and buy/sell-classes down sampling to roughly equalize distribution among all three classes.
2) When training the model we supply prior class distribution to be used for internal weighting designed to compensate for class imbalance.
3) We do nothing during the training process. However, when we generate predictions, we use probability prediction instead of class prediction and distribute range of values for each class in the prediction range of values according to the distribution observed in the training data.
4) Instead of using 3 classes, we split the samples into more classes: 4 classes for buy and 3 classes for sell while keeping the same single class for hold. This way, we obtain the similar number of samples for each class and the model will be balanced hence trained without numerical issues. During prediction, any prediction that falls into the 4 classes of buy, receives the buy label, any prediction falling into one of the 3 sell classes receives a sell label while no change will be observed for predictions of hold class.

We ran a number of experiments to compare the results of these four options and found that the last option (option 4) yields the best performance. As a result, in our final model, we chose to split data into 4 classes of buy, 3 classes of sell and the same class of hold.

B. A regression model

Even though the Competition task is nominally a classification challenge, the classes can be ordinal labelled from sell (value -1) through hold (value 0) to buy (value 1) and thereby gain monotonic relationship with the the return, which in turn allows to represent the task and model it as a regression problem either against continuous return or against only three possible target values of {-1,0,1}. As a result, our second approach is a regression model. Specifically, when running experiments for our first classification model, we found that the model often misclassified cases along the inter-class boundary,

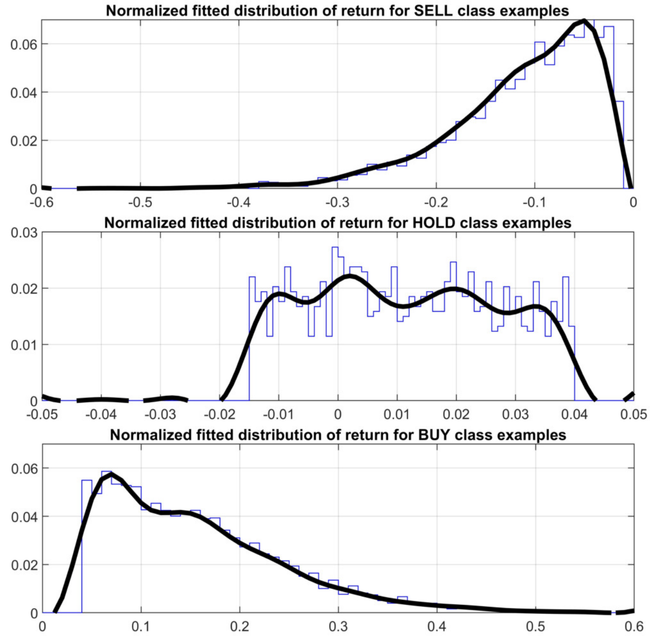

Figure 2. Gaussian process fitted normalized distribution of return within corresponding classes

e.g., stocks with strong features of both sell and hold classes or both hold and buy classes. This effect can be easily explained when investigating continuous return distributions within each class as depicted in Figure 2.

As evident from Figure 2 there are risky cut-offs of prevalent regions around both borders of the hold class with sell and buy classes. For many examples falling into this border regions the return remains virtually the same yet some may fall to different classes. Interestingly the largest chunk of sell class density occupies small negative return right next to the hold class, while buy class tails off more slowly along the growing positive returns. Clearly the regression would be able to better represent small return differences along the classification borders as opposed to step-size changes in the classification and potentially more effectively encourage separation among two adjacent classes: sell and hold as well as hold and buy. The process of fine-tuning threshold values for the class separation is carried out as follows:

- We use Stratified k-Fold to split samples for training and validation to maintain the same distribution of classes.
- We train a regression model using the training data set and generate predictions for the validation data set.
- We compute the distribution of the training data set and use this distribution to find thresholds in the validation data set. Specifically, based on the predictions generated from the model for the validation data set, we set the thresholds so that the distribution of predictions across the three classes is the same as the distribution that we get from the training data (as we use Stratified k-Fold for the splitting earlier).

C. A combination of two binary classification models

While the second model can help to make better separation for stocks in two adjacent classes: sell and hold as well as hold and buy, it cannot address the issue that a stock in sell class is falsely classified into buy class and vice versa. Our third model is designed to address this issue. This model is a combination of the two binary classification models:

- A buy model to determine whether a stock should be bought or not (buy or not-buy). To train this model, we combine samples from sell and hold classes into a single not-buy class while keeping the samples in the buy class unchanged.
- A sell model to determine whether a stock should be sold or not (sell or not-sell). Similarly to the case of the buy model, we combine samples from buy and hold classes into a single not-sell class while keeping samples in the sell class unchanged.

These two models are trained and fine-tuned together so that when combining them for a prediction the following logic is applied:

- A buy prediction of the first model and a not-sell prediction of the second model lead to a buy result.
- A not-buy prediction of the first model and a sell prediction of the second model lead to a sell result.
- A buy prediction of the first model and a sell prediction of the second model lead to a hold result.
- A not-buy prediction of the first model and a not-sell prediction of the second model lead to a hold result.

D. A model considering stock performance

As stock performance has not been considered in any of the above models, this model is designed for this purpose. Here is how the incorporation of the continuous return (stock performance) is separately proposed for classification and regression models:

- For the classification model: as discussed in IV-A, we use 4 classes of buy, 3 classes of sell and a single class of hold. For the buy classes, we put the top 25% of the stocks according to the performance metric into the first class, the next 25% of the stocks going to the second class, and so on. Similarly, we put the top 33% of the stocks into the first sell class, the next 33% of the stocks going to the second sell class, and so on. For the hold class, there is nothing changed as it is a single class.
- For the regression model: instead of using the original values -1, 0, 1 as the target values to train a model, we use the stock performance metric or return to update the target values and stretch it within constrained (-1:1) interval such that only the best performance stock in the buy class is given value 1.0 while other stock values are updated proportionally to their performance in a range from 0.1 to 1.0. Similarly, only the stock having the worst performance in the sell class gains value -1.0 and other values are re-normalized within -1.0 to -0.1 range. Consequently, the hold class examples are also adjusted within the range of values from -0.1 to 0.1.

E. The combination model

After receiving results from different four models, the final step of our multi-dimensional approach is to combine them together. To do this, we tried the two state-of-the-art methods: ensemble and stacking. Note that while these four individual models presented above share a large portion of common features, they do have separate independent features, which are only used exclusively within one model, but not in others. This way, these models are injected with the diversity that helps to combine their results better, i.e. synthetically elevate combined performance above any individual. Here is how the two combination models have been constructed:

- In the ensemble method, we first tried to use the average result from the predictions as the final result. This method suffered from a big issue of class imbalance as it always favors the hold class due to the fact that it falls in the popular middle between the buy and sell classes. To avoid this issue, instead of using the average aggregation, we chose the majority vote method, and received a significant performance gain as a result.
- In the stacking method, we trained a general model that combines the four individual sub-models together with few features. Note that in addition to the results obtained from the four sub-models, we also included features that are not the common features used by the four models when training the stacking model.

Between these two methods, while the stacking method tends to produce a better performance compared to the ensemble method, it is sensitive to changes from individual sub-models and absorbs significantly more time to complete the training process. In the end, we opted for a more stable ensemble method to produce the final predictions of our multi-dimensional model.

V. Parameter Selection

In addition to the multi-dimensional approach, we believe that selecting the right training parameters to avoid overfitting is also a key point that helped us to achieve a good score in the competition. Even though we first chose Grid Search cross-validation to search for the set of parameters maximizing the evaluation score on the validation sets, we stopped using it as soon as we realized the following two points.

- There are only 10% of the test data used for the public score evaluation, and hence the change of few results may already have a big impact on the public score.
- There is always a big gap between local training score and local validation score as well as between the local validation score and the public score evaluation.

According to our experience accumulated throughout many competitions organized on the KnowledgePit platform, in-line with these two points big re-shuffles in the leader board ranking are possible and in fact expected for such a complex and volatility-prone challenge as is the stock trends prediction. Our suspicion was later proven to be correct for this challenge as in the final evaluation on the full testing set our model surpassed all but one competitive models that turned out to be massively overfitted on the preliminary set. Given this justified expectation, we chose the parameters that may not have the best score, but produce stable results to avoid overfitting. Specifically, we chose to train our model with a narrow tree ($tree\_depth = 5$), a small ratio for feature and sample splitting during the training process ($bagging\_fraction = 0.5$, $feature\_fraction = 0.6$ and a high L1 regularization ($lambda\_l1 = 1.0$). This strategy helped us to achieve more stable result and jump from the rank beyond the top-10 in the preliminary set leader board to the 2^{nd} in the final evaluation. Actually, we are one of the only few teams that managed to achieve a big jump in the ranking from the initial public leader board to the final private leader board. This substantial leap demonstrates the effectiveness of our strategy and adjustments made throughout the competition, reflecting our deep understanding of the challenge and the ability to optimize our solution under different evaluation conditions.

VI. Conclusion

In this paper, we have presented in details how our multi-dimensional approach was designed, implemented, and fine-tuned to achieve a very good result in predicting optimal stock trade actions. While our feature engineering could be similar or even on the modest side compared to the competitive teams', given our inexperience in stock trading, we chose to focus on the aspects of financial predictions that are critical and often overlooked: producing a range of alternative very diverse models utilizing different ML paradigms and representations to produce stable, robust yet diverse predictors of the same target function. With such approach, further boosted with conservative cross-validation and hyper-parameter fine-tuning we managed to elevate the performance further using synthetic ensemble combination scheme rather than stacking offering larger though unstable gains, while invariably guarding every design decision with the careful and conservative validation set evaluation. Our model score the 2^{nd} place in the FedCSIS'2024 Competition and offers encouraging and optimistic outlook on the inherently difficult challenge of profitable stock market prediction.

References

[1] 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.

[2] Ariyo, Adebiyi A. and Adewumi, Adewumi O. and Ayo, Charles K., "Stock Price Prediction Using the ARIMA Model," *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 2014, doi=: 10.1109/UKSim.2014.67.

[3] Devi, K.Nirmala and Bhaskaran, V.Murali and Kumar, G.Prem, "Cuckoo optimized SVM for stock market prediction," *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2015, doi: 10.1109/ICIIECS.2015.7192906.

[4] Wijaya, Arya Yudhi and Fatichah, Chastine and Saikhu, Ahmad, "Prediction of Stock Trend Using Random Forest Optimization", *2023 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA)*, 2023, doi: 10.1109/ICAMIMIA60881.2023.10427958.

[5] Fauzan, Achmad and SusanAnggreainy, Maria and Nathaniel, Nicholas and Kurniawan, Afdhal, "Predicting Stock Market Movements Using Long Short-Term Memory (LSTM)", *2023 4th International Conference on Artificial Intelligence and Data Sciences (AiDAS)*, 2023, doi: 10.1109/AiDAS60501.2023.10284713.

[6] Nikhil, Shreyas and Sah, Rahul Kumar and Kumar Parki, Santosh and Tamang, Til Bikram and D, Somashekhara Reddy. and T R, Mahesh, "Stock Market Prediction Using Genetic Algorithm Assisted LSTM-CNN Hybrid Model", *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2023, doi: 10.1109/ICCCNT56998.2023.10306948.

[7] Reddy Beeram, Satyanarayana and Kuchibhotla, Swarna, "A Survey on state-of-the-art Financial Time Series Prediction Models", *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021, doi: 10.1109/ICCMC51019.2021.9418313.

[8] 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.

[9] 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.

[10] 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.
FEATURE ENGINEERING FOR PREDICTING FRAGS IN TACTICAL GAMES

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

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

[24] 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.

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