Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1311–1315
DOI: 10.15439/2023F4457
ISSN 2300-5963 ACSIS, Vol. 35

Gradient boosting models for cybersecurity threat detection with aggregated time series features

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

***Abstract*—The rapid proliferation of Internet of Things (IoT) devices has revolutionized the way we interact with and manage our surroundings. However, this widespread adoption has also brought forth significant cybersecurity challenges. IoT devices, with their interconnectedness and varying functionalities, present a unique threat landscape that requires tailored detection techniques. Traditional approaches to cybersecurity, primarily focused on network monitoring and anomaly detection, often fall short in effectively identifying threats originating from IoT devices due to their dynamic and complex behaviors. This paper addresses our solution for FedCSIS 2023 Challenge: Cybersecurity Threat Detection in the behavior of IoT Devices. First, we aggregated time series features, and then at the feature selection stage, we filtered and combined different categorical and numerical features to generate four different feature sets. The Gradient boosting models, i.e. lightgbm, catboost and xgboost, are applied and trained individually with hyper-parameter tuning. The final three submissions are two best individual lightgbm models with the AUC scores of 0.9999 and 0.9998, respectively on the different feature sets, which secured the 4th place with a final score of 0.9993, and one ensemble result with a AUC score of 0.9998 from combination of xgboost, catboost and lightgbm, which has the final score of 0.9997 while unluckily was missing in the final three evaluation entries.**

***Index Terms*—Cybersecurity threat detection, Gradient Boosting Trees, CatBoost, XGBoost, LightGBM, Stacking, Ensemble Learning.**

I. Introduction

WITH the exponential growth of Internet of Things (IoT) devices, ensuring the security and integrity of these interconnected systems has become a paramount concern. The dynamic and heterogeneous nature of IoT devices presents a unique challenge for traditional cybersecurity approaches.

The survey paper in [1] comprehensively discusses the security issues faced in IoT environments and presents an overview of the existing security mechanisms and solutions. It covers various aspects of IoT security, including authentication, access control, privacy preservation, secure communication, and intrusion detection. The paper also highlights the unique security challenges posed by IoT and provides insights into ongoing research efforts to address those challenges.

Machine learning (ML) techniques have emerged as promising solutions for addressing IoT device security challenges. ML algorithms can analyze vast amounts of data collected from IoT devices to identify patterns, detect anomalies, and make predictions. By leveraging ML, IoT security can be improved through proactive threat detection, effective intrusion detection, and robust anomaly detection. In [2], various machine learning methods applied into IoT have been explored and a Support Vector Machine (SVM) application use case is presented. In [3], this survey paper offers insights into the application of various ML/DL techniques, such as deep learning, support vector machines, and decision trees, in enhancing IoT device security. It discusses the challenges faced in securing IoT devices and highlights the potential of ML methods in addressing those challenges. The paper also provides an overview of different use cases, datasets, and evaluation metrics used in the context of IoT device security.

In this paper, a model utilizing gradient boosting decision trees in conjunction with effective feature engineering and optimized model hyper-parameters, forming an ensemble learning approach has been developed for predicting the cybersecurity breaches to address the task given in the FedCSIS 2023 Challenge [4][1]. The objective of the challenge, which is sponsored by Łukasiewicz Research Network - Institute of Innovative Technologies EMAG and EFIGO sp. z o.o. companies, is to detect the cybersecurity breaches in log data from IoT devices.

The remainder of the paper is organized as follows. The FedCSIS 2023 Challenge is briefly described in Section II. Time series data aggregation and feature engineering is presented in Section III, followed with the description of the gradient boosting models and ensemble learning in Sections IV and V, respectively. The experimental results in Section VI. Concluding remarks are provided in Section VII.

II. FedCSIS 2023 challenge

The FedCSIS 2023 data mining competition focused on the detection of cybersecurity breaches in log data from IoT devices. The data sets contain 1-minute logs of all related system calls. The task for the competition participants is to develop a model that assesses the chances that a cyber attack was ongoing during the monitored period. Such a model could play a vital role in improving the safety of IoT systems. The knowledgepit.ai platform, on which the competition was hosted operated a leaderboard, which provided the feedback to the competitive model prediction submissions in a form of the preliminary AUC score [2] computed over the small subset of the testing set, while the final AUC score for the complete testing set - constituting the final results, were provided after the submissions' closure.

[1]https://knowledgepit.ml/fedcsis-2023-challenge
[2]https://en.wikipedia.org/wiki/Receiver_operating_characteristic

III. Data aggregation and feature engineering

The available training data provided by the competition organizers contain 15027 log files each given in a .csv table format with a uuid4 random name, in which the 1-minute logs of all related system calls are listed together with the timestamps given in the datetime formmat of yyyy-mm-dd-hh:mm:ss, eg. 2023-04-12-00:00:00. A small fraction in the training data is indicated to be hit by a cyberattack. The test data contains 5017 log files having the same format and naming scheme as the training files, while the cyberattack indication is not given.

To consolidate the feature sets, we wrote a python code to aggregate the time series log files, a generic aggregation filter fitting into all columns of the dataset as described below:

1) For numerical columns eleven self-explanative aggregators were applied, including:
   - *minimum*,
   - *maximum*,
   - *mean*,
   - *median*,
   - *sum*,
   - and *standard deviation*

   across all timestamps.
2) For categorical columns the aggregation treatment was made dependent on the most common frequency number of unique values.

In this way, each of the log files has been converted to a vector of aggregation features, representing a sample in both training and testing datasets. A binary label is given to each training sample, which represents whether or not a cyberattack is experienced.

IV. Gradient boosting models

Gradient boosting decision trees (GBDT) algorithms have emerged as a powerful and widely used technique in machine learning and data mining. GBDT combines the strengths of decision trees and boosting, resulting in a highly accurate and robust predictive model. This approach has been successfully applied in various domains, including finance, healthcare, and online advertising [5]. In [6], it provides a comprehensive overview of GBDT algorithms, explaining their theoretical foundations, practical implementation details, and empirical results. The fundamental idea behind GBDT is to iteratively train weak decision trees and sequentially add them to an ensemble, where each subsequent tree aims to correct the mistakes made by the previous ones. This process is guided by a loss function that measures the discrepancy between the actual and predicted values. By minimizing the loss function, GBDT optimizes the model's ability to make accurate predictions. One of the key advantages of GBDT is its ability to handle both numerical and categorical features effectively. Through a process called feature engineering, GBDT algorithms transform raw data into meaningful and informative representations, enhancing the model's predictive capabilities. Efficient feature engineering techniques play a crucial role in improving the accuracy and interpretability of the GBDT model. We applied three popular GBDT algorithms XGBoost, CatBoost, and lightGBM to construct the ensemble learning model for predicting the cybersecurity breaches for this challenge. Moreover, for years our team has been participating in the data science competitions series organized by the KnowledgePit platform[3] using GBDT related algorithms for classification, regression and other related tasks [7] - [22] with outstanding results. Gradient boosting decision trees algorithms, with their ability to handle diverse data types, efficient feature engineering, and model hyper-parameter optimization, have proven to be a powerful tool for predictive modeling in various domains.

Initial tests conducted on the primary dataset provided evident findings, demonstrating that gradient boosting models outperformed other methods in terms of predictive accuracy, while also exhibiting favorable computational efficiency. Notably, when compared to simple linear regression and deep networks, the performance of gradient boosting models was significantly superior. Within the category of gradient boosting models, specifically XGBoost, LightGBM, and CatBoost, were employed and subsequently fine-tuned during the competition. Different variations of these models, trained with diverse parameters, were utilized in second-level ensembles, employing both simple aggregation and stacked retraining techniques.

Modern Machine Learning models have reached a level of sophistication where they offer extensive customization and adaptability to cater to diverse options, versions, and parametric configurations during the model construction process. Gradient boosting models serve as prime examples of such models, as they provide numerous algorithmic, representational, modeling, and statistical parameters that can be fine-tuned to effectively capture and represent the data. The ultimate goal is to learn a reliable regression function that accurately predicts continuous output based on the input variables, ensuring robust generalization on unseen data.

In order to handle the challenge of tuning a large number of parameters for each distinct model, we opted to utilize a fast and efficient rotational grid search approach built on the general grid search hyperparameter tuning [23] for the gradient boosting models: XGBoost, CatBoost, and LightGBM. This method involves assigning up to a set of unique values to each optimizable parameter, covering a comprehensive range within the parameter's search space, regardless of whether it is numerical or categorical.

Unlike an exhaustive parametric grid search, which would be computationally infeasible given the number of parameters involved, our approach focuses on incrementally finding local optima for a specific parameter while keeping the remaining parameters fixed. This process continues in a rotational manner, moving on to the next parameter only after no further improvement can be achieved from any local changes. By adopting this strategy, we can efficiently explore the parameter space without exhaustive evaluation.

[3] https://knowledgepit.ai/

To enhance the reliability of the best parameter configurations discovered, we applied Repeated Stratified 10-Fold cross validator. This technique helps eliminate the possibility of accidentally selecting configurations with unusually high performance. However, to mitigate the additional computational cost associated with cross-validation, we simplified the process of searching for local optima for each parameter. Specifically, we only performed a pair of neighboring checks for each turn, evaluating the performance above and below the current parameter value. The optimal value was then adjusted to the value that exhibited the maximum performance improvement.

By employing this rotational grid search hyperparameter tuning approach and integrating 10-fold cross-validation, we aimed to efficiently and effectively determine the most suitable parameter configurations for the gradient boosting models, ensuring reliable and high-performing models for our purposes.

Optimizing the hyper-parameters of the GBDT model is essential to achieve superior performance. Determining the appropriate values for hyper-parameters such as the learning rate, tree depth, and regularization parameters can significantly impact the model's predictive accuracy and generalization ability. Therefore, model hyper-parameter optimization is a crucial step in harnessing the full potential of GBDT algorithms.

This parameters optimization process is terminated when no improvement in cross-validated AUC performance was found from any local changes of parameters.

V. Ensemble Model

For the final ensemble construction, we employed three fundamental gradient boosting models: XGBoost (XGB), LightGBM (LGBM), and CatBoost (CatB). To enhance the generalization performance of these models, we applied filters. The purpose of filter techniques is to expand the classifier into multiple versions that differ from each other, train them on either the entire training set or subsets of it, and then apply them to the testing set. The outputs of these model versions are subsequently aggregated together.

To further enhance diversity and seek improved predictive performance, we trained all baseline regression 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.

By employing the combination of baseline gradient boosting models, diversification filters, and diverse feature subsets, we aimed to construct a final ensemble that not only exhibited enhanced diversity but also delivered superior predictive performance.

Furthermore, in order to explore additional avenues for performance improvement, we introduced an additional stacked layer consisting of simple linear regression. This layer was trained using the outputs generated by the baseline models. To facilitate the stacking layer's integration, we divided the training data into two distinct parts. The first part was utilized to construct the baseline models, while the second part was reserved specifically for learning the parameters of the linear regression model within the stacking layer.

In the end, we merged the outputs of each individual model and the outputs from the linear regression-based stacking by taking their average. The architecture, represented as a flow chart, showcasing the structure of the final ensemble, can be observed in Figure 1.

VI. Experimental Results

Throughout the competition, we used sklearn packages, xgboost, lightgbm and catboost under Python3 Jupyter Notebook[4] in a Windows Server Virtual Machine with 128G RAM memory and Intel(R) Xeon(R) Gold 6230R CPU@2.10GHz, 2 Processors to run simulations. We did intensive feature aggregation and different feature combination by removing or filtering some columns as mentioned below in Table I.

The different features sets, along with their corresponding impact on the performance of individual models on the limited and sparse training and testing datasets, is summarized in Table II.

While numerous parametric variants demonstrated strong performance throughout the competition, we obtained our best individual model scores by utilizing specific model parameters, as indicated in Table III.

It can be easily seen from Table II, the performance of each model for different feature datasets has only slightly difference, even for 40 features only we can also achieve AUC 0.9999, it is most probably due to the very limited and unbalanced training and test dataset, and as AUC score is near 1, so honestly speaking it is very difficult to improve the model performance consider the trade off potential or already model overfitting problem which is very much challenging in terms of considering model stability rather than accuracy performance.

To be safe and ensure our model is more robust enough, we only consider the full aggregated features of 149 features and the compact version of 40 features, and also build our ensemble model on top of the three individual model LGBM, XGB and CatB to make it more robust. Our chosen datasets, (AUC) results, along with the optimal model parameters determined for each model are described below:

1) Feature set 1 with 149 features:
   - CatB (learning rate 0.02, depth 3, iterations 1000): 0.9983
   - LGBM (learning rate 0.02, depth 3, iterations 1000): 0.9999
   - XGB (learning rate 0.01, depth 3, iterations 3000): 0.9976
2) Feature set 2 with 40 features:
   - CatB (learning rate 0.02, depth 3, iterations 1000): 0.9986
   - LGBM (learning rate 0.02, depth 3, iterations 1000): 0.9998

[4]https://jupyter.org/

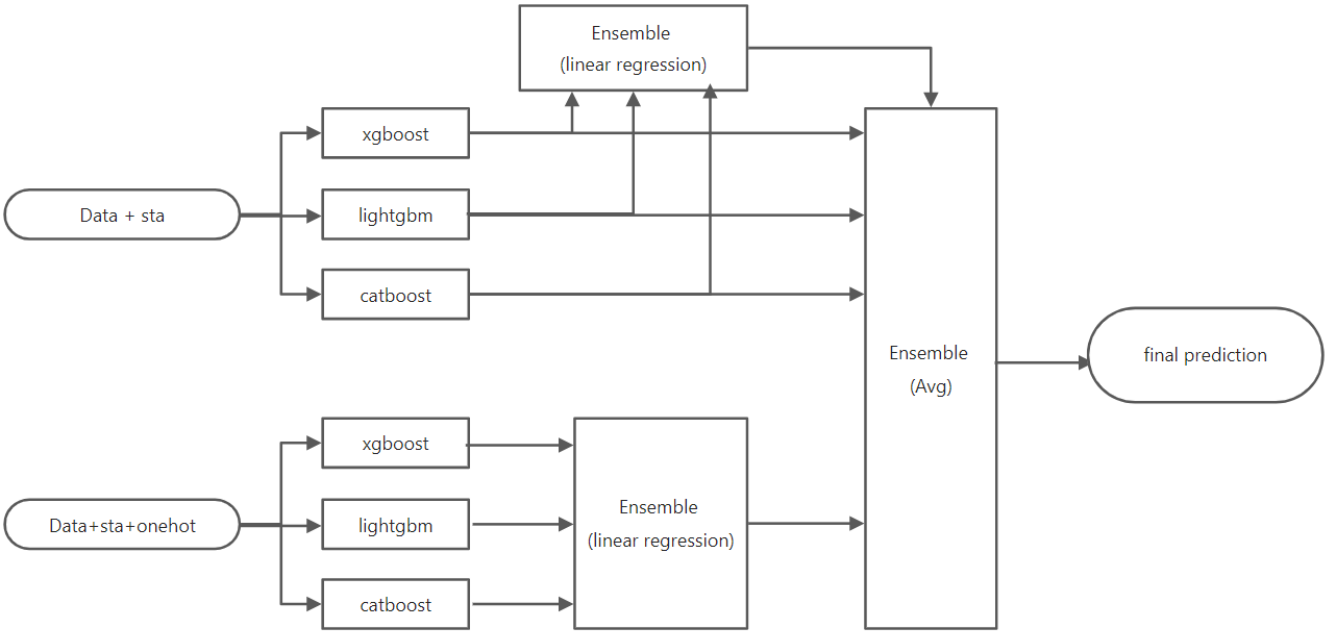

Figure 1. Flowchart of the final ensemble model.

Table I
VERSION OF FEATURES COMBINATION DATASETS

Version	Number of Features	Ignored Columns Numbers	Ignored columns names
V1	149	0	Full aggregated columns
V2	40	19 columns ignored on top of V1	"USER_AUTH","USER_MGMT_COUNT","CRED_COUNT", "USER_ERR_COUNT","USYS_CONFIG_COUNT","CHID_COUNT", "SELINUX_ERR_COUNT","SYSTEM_COUNT","SERVICE_COUNT", "DAEMON_COUNT","NETFILTER_COUNT","SECCOMP_COUNT", "AVC_COUNT","ANOM_COUNT","INTEGRITY_COUNT", "KERNEL_COUNT","RESP_COUNT","SELINUX_MGMT_COUNT", "CUSTOM_openSockets"
V3	28	2 columns ignored on top of V2	"KILL_process","KILL_uid"
V4	23	5 columns ignored on top of V3	"SYSCALL_exit_hint_common", "USER_ACTION_op_common", "USER_ACTION_src_common", "USER_ACTION_res_common", "USER_ACTION_addr_common"

Table II
FEATURES AND AUC-MEASURED INDIVIDUAL MODEL PERFORMANCE

Version	Number of Features	LGBM	XGB	CatB
V1	149	0.9999	0.9983	0.9976
V2	40	0.9998	0.9988	0.9986
V3	28	0.9993	0.9975	0.9982
V4	23	0.9990	0.9976	0.9981

Table III
OPTIMIZED INDIVIDUAL MODEL PARAMETERS

Model	Encoder	Iterations	Learning Rate	Tree Depth
LGBM	onehot	1000	0.02	3
XGB	ordinal	3000	0.02	3
CAT	onehot	1000	0.01	3

- XGB (learning rate 0.01, depth 3, iterations 3000): 0.9988

And, stacking ensemble models based on linear regression were trained using the outputs from the diversified individual models. These stacking models exhibited preliminary AUC performance as follows:

- Stacking model with 149 features: 0.9998
- Stacking model with 40 features: 0.9996

The final predictions are obtained by averaging the outcomes of both stacking models and diversified individual baseline models using an ensemble approach. This ensemble averaging results in an AUC of approximately 0.9998 in the preliminary score. However due to some mistakes, this

ensemble model submission was missing in the three final entries, so we only submitted two entries of LGBM which scored the 0.9993 as the 4th place, while this ensemble model can achieve 0.9997 AUC score evaluated with the final released test labels which means the ensembled model is more robust than the single model.

VII. Conclusions

We endeavored to enhance the predictive performance of the already robust regression models within the gradient boosting family, namely XGBoost, LGBM, and CatBoost. To accomplish this challenge, we applied a range of GBDT methods, combined with different ensemble combination techniques and observed improved performance achieved through aggregating the expanded set of diverse model versions. Additionally, we employed linear regression-based stacking and selected the most effective ensemble candidates based on the trade-off between performance and diversity.

We applied this proposed ensemble approach to the challenging task of advance prediction of cybersecurity breaches in IoT device log data, which involved various types and forms of data. Our solution was implemented and evaluated within the competitive framework of the FedCSIS 2023 data mining challenge. In the preliminary leader-board of the challenge, our proposed solution achieved the fifth position with an AUC of 0.9999, while in the final ranking we are 4th place with AUC score 0.9993 even though unluckily our ensemble model entry was missing in the final three entries and this ensemble model can achieve the AUC score 0.9997 with most stable and robust compared the AUC score 0.9998 in the preliminary leader-board. Our solution holds potential for enabling network service providers to better anticipate hacker threats and bolster their cybersecurity measures.

References

[1] F. Alaba, M. Othman, I. Hashem, F. Alotaibi, Internet of Things Security: A Survey, *Journal of Network and Computer Applications*, vol. 88, pp. 10-28, 2017.

[2] M. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, A. Sheth, Machine learning for internet of things data analysis: a survey, *Digital Communications and Networks*, 2018.

[3] M. Garadi, A. Mohamed, A. Ali, X. Du, I. Ali and M. Guizani, A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security, *IEEE Communications Surveys & Tutorials*, 2020.

[4] A. Janusz, A. Kozłowski, B. Adamczyk, D. Iwanicki, M. Brzęczek, M. Michalak, M. Tynda, M. Czerwiński, P. Biczyk, Predicting the Cybersecurity Threat Detection in the Behavior of IoT Devices: Analysis of Data Mining Competition Results, *Proceedings of the 18th Conference on Computer Science and Intelligent Systems (FedCSIS)*, 2023.

[5] 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 Information Processing Systems 12: 512–518, MIT Press, 1999.

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

[7] 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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] 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.

[21] 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.

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

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