DOI: 10.15439/2023F7136
ISSN 2300-5963 ACSIS, Vol. 35

Spotting Cyber Breaches in IoT Devices

Sławomir Pioroński
Faculty of Mathematics and Computer Science
Adam Mickiewicz University
Uniwersytetu Poznańskiego 4 Street
61-614 Poznań, Poland
Email: slawomir.pioronski@amu.edu.pl

Tomasz Górecki
Faculty of Mathematics and Computer Science
Adam Mickiewicz University
Uniwersytetu Poznańskiego 4 Street
61-614 Poznań, Poland
Email: tomasz.gorecki@amu.edu.pl

***Abstract*—In the ever-growing realm of the Internet of Things (IoT), ensuring the security of interconnected devices is of paramount importance. This paper discusses the process of spotting cyber breaches in IoT devices, a significant concern that needs urgent attention due to the susceptibility of these devices to hacking and other cyber threats. With billions of IoT devices worldwide, the detection and prevention of cybersecurity breaches are critical for maintaining the integrity and functionality of networks and systems.**

In this paper, we showcase the outcomes achieved by employing the LightGBM technique for a cyberattack prediction challenge, which was a part of the FedCSIS 2023 conference.

***Index Terms*—cybersecurity, data mining competition, LightGBM**

I. Introduction

As we step further into the digital era, the Internet of Things (IoT) continues to reshape the landscape of our daily lives, driving advancements in various sectors such as healthcare, transportation, smart homes, and industrial automation. Despite the remarkable benefits, the rapid proliferation of IoT devices has significantly heightened the stakes in the domain of cybersecurity. The interconnected nature of these devices poses unique vulnerabilities, making them attractive targets for cyberattacks. An essential part of combating this growing threat involves the ability to effectively identify and predict cybersecurity breaches in IoT systems.

Numerous machine learning methodologies can be deployed for the prediction of cyberattacks [1]. However, we opted for a gradient boosting algorithm, specifically LightGBM [2], due to its impressive combination of speed and precision. In this paper, we aim to highlight the effectiveness of our strategy. Our discussion will serve to underscore the integral role of data science in augmenting cybersecurity measures in an increasingly interconnected world. By delving into this topic, we hope to provide valuable insights for future research endeavors and practical applications aimed at advancing the field of cybersecurity for IoT.

The organization of this paper is as follows: after this introduction, we review relevant literature and provide a brief overview of the FedCSIS 2023 challenge. In Section IV, we delve into the processes involved in data handling and preparation. We detail the model deployed in our experiment in Section V, followed by a comprehensive presentation of our findings in the succeeding section. We conclude in Section VII by summarizing our observations and contemplating potential avenues for future exploration.

II. Related work

The practice of automatically detecting cyberattacks has a well-established history in the field. A diverse range of methods have been employed to accomplish this task. It has been suggested through numerous studies that machine learning techniques could be potentially beneficial, with many researchers opting to use unsupervised algorithms to navigate identification challenges [3], [4]. However, there is a notable drawback to using unsupervised machine learning methods for recognizing anomalies in a network, distinguishing between standard cyberattacks, and detecting outliers. The sparse occurrence of these outliers can have an asymmetric impact on both the success rate and the identification of abnormalities.

To achieve more dependable results, supervised machine learning methods are often employed. These algorithms are trained using metadata with labels indicating whether the given instances have previously been classified as cyberattacks. Examples of such supervised learning algorithms include Support Vector Machines and Artificial Neural Networks [5], Random Forests [6], the k-Nearest Neighbor (k-NN) technique [7], the Naive Bayes algorithm [8], and LightGBM [9].

In our solution, we decided to use LightGBM (Light Gradient Boosting Machine) due to several reasons [2], [10], [11]:

- **Efficiency**. LightGBM uses a novel technique of Gradient-based One-Side Sampling (GOSS) to filter out the data instances for finding a split value, which can result in a more efficient learning process. This is particularly useful when dealing with large volumes of data generated by IoT devices.
- **High Performance**. LightGBM can handle large data sets while maintaining high efficiency. It uses the leaf-wise tree growth algorithm, unlike the traditional level-wise tree growth algorithm, which can result in a better performance in terms of speed and accuracy.
- **Handling Categorical Features**. LightGBM can naturally handle categorical features, which can be very beneficial when dealing with IoT data, as IoT devices can produce a variety of data types.

- **Scalability**. LightGBM is highly scalable and can work well with large datasets that often characterize IoT networks.
- **Accuracy**. LightGBM can achieve lower prediction errors by employing complex tree architectures, boosting its accuracy, which is crucial for detecting subtle signs of cyberattacks in IoT networks.

It is also worth noting that gradient-boosting models have been used in previous data mining competitions. In the IEEE BigData 2019 Cup: Suspicious Network Event Recognition, the best solutions used tree-based boosting models [12]. In particular, first place went to an ensemble of two models [13], LightGBM and XGBoost [14]. In another competition, the FedCSIS 2020 Challenge: Network Device Workload Prediction [15], the situation was similar. The 2nd and 3rd place solutions used XGBoost models. Of course, it is important to remember that proper preprocessing is required to use these models. Furthermore, we have used a similar approach (LightGBM + appropriate preprocessing) for other competitions with outstanding results [16].

III. Challenge description

A. Data

The data provided consists of CSV table log files, each with a randomized uuid4 name. All original timestamps have been standardized to a specific timestamp, which is 2023-04-12-00:00:00. A separate TXT file was provided for the training set, containing the names of log files associated with cyber attacks. After the competition concluded, similar information regarding the test set was also made available. The sizes of the datasets are as follows:

- training data: 15 027 files (522 indicates cyberattack),
- test data: 5 017 files (176 indicates cyberattack).

As we can see, a small number of files indicated a cyberattack (3.48% for the training dataset and 3.50% for the test dataset).

B. Task

Our goal is to develop an accurate method that can detect cyberattacks on an IoT system based on its logs.

C. Evaluation

In this competition, participants submitted their solutions to the online evaluation system as text files that included predictions for the test instances. Each test instance in the solution file was accompanied by a single number within the $[0, 1]$ range, representing the probability of a cyberattack. These predictions were arranged according to the lexicographic ordering of the log files from the test set.

The effectiveness of the submitted entries was assessed using the ROC AUC (Receiver Operating Characteristic Area Under Curve) metric, a widely used evaluation metric for binary classification problems [17]. The ROC curve is a plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It is created by plotting the true positive rate (TPR), against the false positive rate (FPR), at various threshold settings. Precisely

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}},$$
$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}},$$

where TP is the number of True Positives, FN is the number of False Negatives, FP is the number of False Positives and TN is the number of True Negatives.

Calculation of the AUC (Area Under the ROC Curve) is slightly more complex as it involves integration over all possible classification thresholds. But practically, it's usually calculated using the trapezoidal rule [18]. An AUC of 1 indicates a perfect classifier, while an AUC of 0.5 signifies a classifier that performs no better than random chance [19].

Initial scores were evaluated via the KnowledgePit online platform [20] and published on a challenge leaderboard calculated on a small subset of the test set fixed for all participants. The final score was published after the challenge using the remainder of the test data set.

IV. Data preprocessing

Due to the format of the data (we have a separate file for each observation, and therefore a separate table with data), we had to process it in an appropriate way. We focused on the approach to have one row of data for a single observation. Each file contains 40 columns: 21 numeric, 17 string, and 2 with only null values (based on training data). We skip these two columns with nulls and now proceed to preprocess the data by type.

A. Numerical data

We focused on the numerical data first. For each file, we took the smallest, largest, and average values (omitting features with the same values, we obtained 17 features). With this simple approach, we will get very good predictions. So, we now move on to data of string type.

B. String data

The main idea was to focus on finding significant differences in this data type without considering numerical data. Of particular note is the "Custom_openFiles" feature. To begin with, we selected unique values for this feature separately for the files that represented the logs with and without the attack. Then, from the unique values from files with attacks, we removed all the values that were present in files without attacks. Finally, for each file, an indicator was created indicating whether any value from the "Custom_openFiles" column belonged to that set. Using the same set this was repeated for the test data. Passing this indicator as the probability of a cyberattack, we obtained a score of 96.77% (by ROC AUC measure) on the public part of the test set.

C. *Addtional preprocessing*

In the test set, the shortest log file contains 68 items. So we took only those files from the training set that are not shorter than it. Thus, we get rid of 65 (0.43%) files from the training set. In addition, we replaced one of the string-type features with a numeric one. Namely, in the feature "SYSCALL_exit _hint" we have numeric and string type values. So in place of the string, for example, "ENOENT(No such file or directory)", we inserted nulls. Then we calculated the average, minimum, and maximum as in Section IV-A.

V. Model

We used the gradient boosting model for testing, and the choice was LightGBM [2]. We used Microsoft's FLAML library [21] to optimize the hyperparameters.

With the above preprocessing, the models achieve high predictive quality very quickly. We can see a comparison of the performance of the models for different subsets of features with hyperparameter optimization taking 3 minutes in Table I.

TABLE I
STRATIFIED 5-FOLD CROSS-VALIDATION RESULTS FOR DIFFERENT FEATURE SETS (3 MIN OF HYPERPARAMETER OPTIMIZATION, AUC MEASURE) AND THE RESULT ON THE TEST SET.

Feature set	Cross-validation	Test set
IV-A	0.99932	**0.99954**
IV-A + IV-B	0.99993	0.99951
IV-A + IV-C	0.99813	0.98224
IV-A + IV-B + IV-C	**1.00000**	0.99603

In Table II, we have a list of optimized hyperparameters and values for the best model from Table I.

TABLE II
FINAL MODEL HYPERPARAMETERS FOR FEATURE SET IV-A AT 3 MINUTES OF OPTIMIZATION (TO FIVE DECIMAL PLACES).

Hyperparameter	Value
n_estimators	1098
num_leaves	120
min_child_samples	5
learning_rate	0.19275
max_bin	1023
colsample_bytree	0.73337
reg_alpha	0.00098
reg_lambda	0.24821

VI. Experimental results

We see that the first two cases in Table I gave the best results on the test set. In both cases, we have another feature that is most relevant according to gain importance [14]. The most significant feature for the 1st model is “SYSCALL_pid_max” (which is the maximum of the “SYSCALL_pid” feature from each file) as we see in Figure 1.

On the other hand, the graph for the second model appears quite similar, except that the newly added feature (indicator based on “Custom_openFiles”, described in IV-B) is now positioned at the beginning.

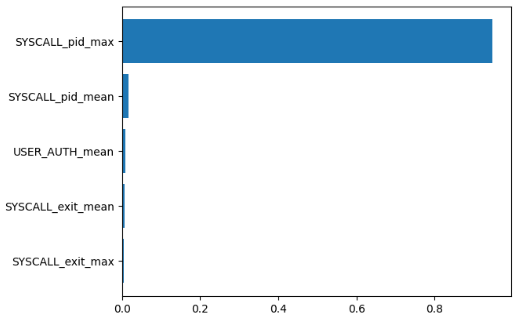

Fig. 1. Top 5 features by gain for the first model. The values were divided by the sum of all gains.

We now set the search times for hyperparameters to 30 minutes. We can see the results in Table III.

TABLE III
STRATIFIED 5-FOLD CROSS-VALIDATION RESULTS FOR DIFFERENT FEATURE SETS (30 MIN OF HYPERPARAMETER OPTIMIZATION, AUC MEASURE) AND THE RESULT ON THE TEST SET.

Feature set	Cross-validation	Test set
IV-A	0.99946	**0.99959**
IV-A + IV-B	0.99996	0.99901
IV-A + IV-C	0.0.99888	0.0.98733
IV-A + IV-B + IV-C	**1.00000**	0.99660

The outcomes show minimal variation from the 3-minute version, as demonstrated more accurately in the learning curve of one feature set presented in Figure 2.

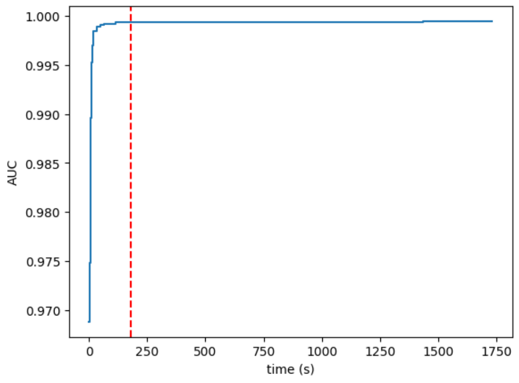

Fig. 2. Learning curve for model trained on IV-A feature set. The red line marks 3 minutes.

We have 4.5 times more false positives than false negatives (in the case of the first model), which is good behavior since it is better to verify claims with no attacks than to omit those with attacks. We can see this in the confusion matrix in Figure 3.

VII. Conclusions

To detect cyberattacks, we utilized the renowned LightGBM model along with some data preprocessing. Our approach

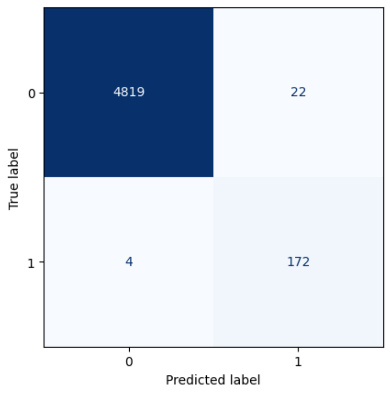

Fig. 3. Confusion matrix for model trained on IV-A feature set.

was successfully trained within a mere 3 minutes, securing a commendable 3rd place in the competition. The top 4 results were closely matched, with only a marginal difference of 0.0002 between the following positions.

The achieved result was already commendable, making it difficult to anticipate a substantial enhancement in performance. Nonetheless, for future endeavors, it is crucial to concentrate on extracting valuable insights from string-type attributes. Furthermore, it is possible to expect that more advanced data preprocessing techniques may contribute to marginal enhancements in performance.

REFERENCES

[1] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*, ser. Springer Series in Statistics. Springer, 2009.

[2] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017, pp. 3146–3154.

[3] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, "Towards developing network forensic mechanism for botnet activities in the IoT based on machine learning techniques," in *Mobile Networks and Management*, J. Hu, I. Khalil, Z. Tari, and S. Wen, Eds. Cham: Springer International Publishing, 2018, pp. 30–44.

[4] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in *Network and Distributed Systems Security (NDSS) Symposium*, 2018.

[5] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks using artificial neural network intrusion detection system," in *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, 2016, pp. 1–6.

[6] Y. Meidan, M. Bohadana, A. Shabtai, M. Ochoa, N. O. Tippenhauer, J. D. Guarnizo, and Y. Elovici, "Detection of unauthorized IoT devices using machine learning techniques," 2017.

[7] M. Aljabri, A. A. Alahmadi, R. M. A. Mohammad, F. Alhaidari, M. Aboulnour, D. M. Alomari, and S. Mirza, "Machine learning-based detection for unauthorized access to IoT devices," *Journal of Sensor and Actuator Networks*, vol. 12, no. 2, 2023.

[8] C. Malathi and I. N. Padmaja, "Identification of cyber attacks using machine learning in smart iot networks," *Materials Today: Proceedings*, vol. 80, pp. 2518–2523, 2023.

[9] M. Al-kasassbeh, M. A. Abbadi, and A. M. Al-Bustanji, "LightGBM algorithm for malware detection," in *Intelligent Computing*, K. Arai, S. Kapoor, and R. Bhatia, Eds. Cham: Springer International Publishing, 2020, pp. 391–403.

[10] A. Anghel, N. Papandreou, T. Parnell, A. D. Palma, and H. Pozidis, "Benchmarking and optimization of gradient boosting decision tree algorithms," 2019.

[11] C. Bentéjac, A. Csörgo, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, pp. 1937–1967, 2020.

[12] A. Janusz, D. Kałuza, A. Chądzyńska-Krasowska, B. Konarski, J. Holland, and D. Ślęzak, "Ieee bigdata 2019 cup: Suspicious network event recognition," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019. doi: 10.1109/BigData47090.2019.9005668 pp. 5881–5887.

[13] Q. H. Vu, D. Ruta, and L. Cen, "Gradient boosting decision trees for cyber security threats detection based on network events logs," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019. doi: 10.1109/BigData47090.2019.9006061 pp. 5921–5928.

[14] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016. doi: 10.1145/2939672.2939785 pp. 785–794.

[15] A. Janusz, M. Przyborowski, P. Biczyk, and D. Ślęzak, "Network device workload prediction: A data mining challenge at knowledge pit," in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, 2020. doi: 10.15439/2020F159 pp. 77–80.

[16] S. Pioroński and T. Górecki, "Using gradient boosting trees to predict the costs of forwarding contracts," in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, 2022. doi: 10.15439/2022F299 pp. 421–424.

[17] K. A. Spackman, "Signal detection theory: Valuable tools for evaluating inductive learning," in *Proceedings of the Sixth International Workshop on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, pp. 160–163.

[18] R. L. Burden, *Numerical analysis*, 8th ed. Thomson Brooks/Cole, 2005.

[19] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[20] "FedCSIS 2023 challenge: Cybersecurity threat detection in the behavior of IoT devices," https://knowledgepit.ml/fedcsis-2023-challenge/, accessed: 2023-07-05.

[21] C. Wang, Q. Wu, M. Weimer, and E. Zhu, "FLAML: A fast and lightweight AutoML library," in *MLSys*, 2021.