Software Impacts 14 (2022) 100446

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

IDS-ML: An open source code for Intrusion Detection System development using Machine Learning

Li Yang *, Abdallah Shami

Department of Electrical and Computer Engineering, University of Western Ontario, 1151 Richmond St, London, Ontario, Canada N6A 3K7,

ARTICLE INFO

Keywords:
Intrusion Detection System
Machine Learning
Ensemble learning
Hyperparameter optimization
Cybersecurity
Zero-day attacks

ABSTRACT

Due to the expansion and development of modern networks, the volume and destructiveness of cyber attacks are continuously increasing. Intrusion Detection Systems (IDSs) are essential techniques for maintaining and enhancing network security. IDS-ML is an open-source code repository written in Python for developing IDSs from public network traffic datasets using traditional and advanced Machine Learning (ML) algorithms. With optimized ML models, the IDSs developed in the repository can identify various types of cyber-attacks to protect modern networks. This code repository can be easily implemented and reproduced on any intrusion detection datasets to solve problems in the cybersecurity field.

Code metadata

Current code version	V1.0
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2022-260
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/8297382/tree/v1
Legal Code License	MIT License
Code versioning system used	none
Software code languages, tools, and services used	Python, Jupyter Notebook
Compilation requirements, operating environments & dependencies	Python 3.6+, Scikit-learn, Xgboost, Lightgbm, Catboost, FCBF, Scikit-optimize, Hyperopt, River
If available Link to developer documentation/manual	https://github.com/Western-OC2-Lab/Intrusion-Detection-System-Using-Machine-Learning/blob/main/README.md
Support email for questions	lyang339@uwo.ca

1. Introduction to IDS and ML

With the rapid expansion of the Internet and communication technologies, as well as the vast number of applications accessible on the network, network security has become a serious issue that must be addressed. Various cybersecurity mechanisms and protection systems have been introduced to protect modern networks, such as firewalls, authentication techniques, cryptography methods, and Intrusion Detection Systems (IDSs) [1]. IDS monitors network traffic in order to identify abnormal activities or malicious cyber attacks [2]. When suspicious behavior is detected, an IDS will generate an alarm and reports it to the network administrator. Additionally, corresponding countermeasures will then be taken to defend against the ongoing attack and prevent future attacks [3].

IDSs can be categorized as signature-based IDSs, anomaly-based IDSs, and hybrid IDSs [4]. The signature-based IDSs are developed to detect known attacks whose patterns or signatures have already

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: lyang339@uwo.ca (L. Yang), abdallah.shami@uwo.ca (A. Shami).

https://doi.org/10.1016/j.simpa.2022.100446
Received 10 November 2022; Accepted 15 November 2022

been defined in the system [5]. Although signature-based IDSs usually achieve high performance on known attack detection tasks, they are unable to detect new or zero-day attacks since their patterns are unknown. On the other hand, anomaly-based IDSs are designed to detect zero-day attacks by distinguishing unknown attacks from pre-defined normal activities [6]. However, their performance on known attack detection is often lower than the performance of signature-based IDSs. Hybrid IDSs are designed to detect both known and unknown attacks by integrating signature-based IDSs and anomaly-based IDSs.

Machine Learning (ML) techniques have recently become promising solutions for developing IDSs. ML is a collection of techniques that employ mathematical formulae to automatically discover, examine, and extract patterns from data [7]. Extracting and acquiring meaningful information helps ML models make informed judgments and predictions. ML algorithms can be classified as supervised and unsupervised learning algorithms [8]. Supervised learning algorithms are a class of ML algorithms that map input variables to a target variable using labeled data for training, such as K-Nearest Neighbors (KNN) [9], Decision Tree (DT) based models [10], and Deep Learning (DL) algorithms [11], etc. Unsupervised learning algorithms are utilized to discover patterns from unlabeled data, such as k-means [12], Gaussian Mixture Model (GMM) [13], isolation forest [14], etc. For IDS development, supervised learning algorithms are often used to develop signature-based IDSs by training on labeled network datasets, while unsupervised learning algorithms can be used in anomaly-based IDSs to distinguish outliers from normal data.

Effectively identifying cyberattacks is a critical challenge for network operators and managers, particularly in the rapidly evolving modern networks. To improve intrusion detection accuracy and defend against more attacks, many advanced ML techniques can be used to develop IDSs, including ensemble learning, Transfer Learning (TL), and Hyper-Parameter Optimization (HPO). Ensemble learning techniques are designed to improve model learning performance by integrating the output of multiple single ML algorithms as base models, including voting, bagging, stacking, etc. [15,16]. TL is an advanced technology that transfers pre-trained models on other datasets or tasks to the target data to improve model training efficiency [17]. HPO is the process of automatically tuning the hyperparameters of ML models to obtain optimized ML models with improved performance [18]. In the IDS-ML code repository, three novel IDS frameworks are provided using advanced ML techniques.

2. The IDS-ML code functionalities and key algorithms

IDS-ML is a code repository that allows researchers to design IDSs to protect modern networks using various ML algorithms. IDS-ML provides solutions to the following research questions:

- What is the general process of intrusion detection system development?
- How can we use ML algorithms to design different types of IDSs (i.e., signature-based IDSs, anomaly-based IDSs, and hybrid IDSs)?
- How can we improve intrusion detection performance with advanced techniques (i.e., ensemble learning, TL, and HPO)?

A high-level overview of IDS-ML is illustrated in Fig. 1. The IDS-ML code repository provides the code implementations for the development of three innovative IDSs: the tree-based IDS [19], the Leader Class and Confidence Decision Ensemble (LCCDE) IDS [20], and the Multi-Tiered Hybrid IDS (MTH-IDS) [21]. Specifically, the IDS-ML code repository includes the following code files:

1. *Tree-based_IDS_GlobeCom19.ipynb*: This code is the implementation of the tree-based IDS proposed in [19] to detect various types of known cyber-attacks. The proposed IDS trains four common ML models, Decision Tree (DT), Random Forest (RF), Extra Trees (ET), and Extreme Gradient Boosting (XGBoost), as base models, and then uses stacking, an ensemble learning method, to construct a robust ensemble model by integrating the four base models. Using the stacking ensemble for final decision-making can further improve intrusion detection accuracy.
2. *LCCDE_IDS_GlobeCom22.ipynb*: This code is the implementation of an innovative IDS framework named LCCDE [20] to identify various types of known cyber-attacks. It is developed by identifying the best-performing ML model among three advanced ML algorithms (XGBoost, LightGBM, and CatBoost) for each attack class or type. The class leader models and their prediction confidence values are then used to make accurate decisions about the detection of distinct cyberattack types. The main advantage/improvement of the proposed LCCDE framework is that it can achieve the highest performance on all the classes (all types of attack detection) in the datasets among the base models. Thus, its overall performance can be improved.
3. *MTH_IDS_IoTJ.ipynb*: This code is the implementation of a comprehensive IDS named the MTH-IDS [21]. It detects both known and unknown attacks by combining a signature-based IDS with an anomaly-based IDS. The signature-based IDS is created by expanding the tree-based IDS model by using Bayesian Optimization (BO), an intelligent HPO approach, to tune the hyperparameters of ML models and generate optimized ML models. On the other hand, the anomaly-based IDS is developed by proposing a Cluster Labeling (CL) k-means method and biased classifiers to distinguish unknown attacks from normal activities, and their performance is improved by tuning their hyperparameters with BO. By implementing the comprehensive MTH-IDS framework, both known and zero-day attacks can be detected effectively.

Additionally, the code repository introduces the public code of a Transfer Learning-Convolutional Neural Network (TL-CNN) IDS [22] and a general HPO tutorial [18]. In the TL-CNN code [22], it implements an intelligent IDS to develop an improved IDS framework using TL and optimized CNN techniques. Specifically, it employs TL techniques by transferring four cutting-edge CNN models, including VGG16, VGG19, Xception, Inception, and InceptionResnet [23], to the intrusion detection tasks by transforming network traffic data into pictures. Consequently, a novel mechanism for data transformation is also presented. In addition, it employs Particle Swarm Optimization (PSO) [24], a robust HPO technique, to automatically modify the hyperparameters of CNN models in order to get optimized CNN models. Lastly, the fundamental CNN models are combined using two ensemble procedures, confidence averaging and concatenation, to boost the intrusion detection performance. The HPO code repository [18], which has received more than 1000 GitHub stars, introduces the general HPO techniques that can be used to tune the hyperparameters of common ML models to optimize their performance. Details of each algorithm in all code can be found in [18]–[22].

The software was developed in Python programming language and based on several Python packages, including Scikit-learn [25], Numpy [26], Pandas [27], Xgboost [28], Lightgbm [29], Catboost [30], FCBF [31], Scikit-optimize [32], Hyperopt [33], and River [34]. A public benchmark cybersecurity dataset, CICIDS2017 [35], is used to evaluate the proposed IDS frameworks in the software. It is a cutting-edge dataset for network security that contains the most current attack patterns. The CICIDS2017 dataset contains various types of cyber-attacks, including Denial of Service (DoS) attacks, port-scan attacks, brute-force attacks, web attacks, botnets, and infiltration attacks.

3. Software impacts

Cybersecurity is an essential challenge in the current and future generations of networks. Although there are many existing papers for IDS development, the public and complete code for ML-based IDSs is limited. IDS-ML's source code and datasets are made available to the

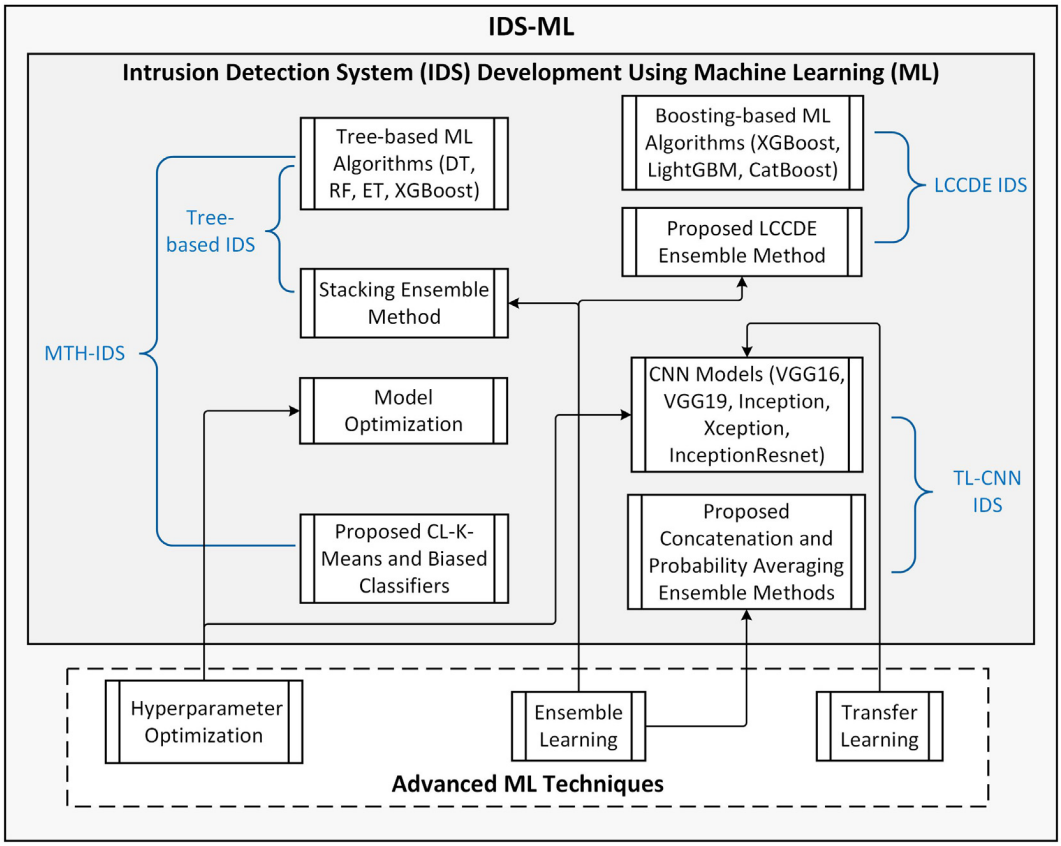

Fig. 1. A high-level overview of the IDS-ML code repository.

general public under the MIT license to facilitate further study in this field. IDS-ML is an innovative and practical project that fills the gap of open source intrusion detection system development.

As the code is publicly available, many researchers and network data analysts have reproduced and used this code in their projects or tasks. Currently, it has received 135 stars and 23 forks on GitHub. Additionally, the corresponding papers of the code have received more than 120 citations. Reproducibility and transparency are two other advantages of this software, which are important for general ML and big data analytics projects to improve the general public's interest and trust. It is expected to attract broader attention and usage in the near future. The telecommunications industries can also design IDSs with this code to protect their networks.

Another strength of the software is that it is completely written in Python, a programming language with an easy-to-understand syntax that has been widely employed in recent ML-related development projects. The flexibility of Python enables the proposed software to be reused, extended, and integrated with various other libraries in the intrusion detection field.

From the technical perspective, most existing IDS code repositories and software are developed based on traditional and basic ML or DL algorithms. The IDS-ML repository improves the existing IDS research by introducing many advanced techniques, such as ensemble learning, transfer learning, and hyperparameter optimization. Through these techniques, the detection accuracy and efficiency of existing IDSs can be significantly improved. Therefore, network researchers and administrators can benefit from the IDS-ML software by learning advanced techniques to improve their IDSs. With the wider application of effective IDSs driven by this IDS-ML repository, cyberattacks in the next generation of networks can be better addressed to enhance cybersecurity.

Lastly, in addition to network users, the ML techniques used in the IDS-ML code repository can be used as generic models to solve general classification problems [36], such as image classification, disease diagnostics, user behavior recognition, etc. Thus, general ML researchers and data analysts can benefit from this software.

4. Conclusions and future research directions

Cyber attacks are becoming more damaging and sophisticated. Detecting different types of attacks and understanding their patterns are crucial procedures in network security frameworks. The IDS-ML code repository provides easy-to-use IDS frameworks to apply traditional and advanced ML techniques to the state-of-the-art network traffic dataset for intrusion detection in modern networks. Network and cybersecurity researchers can take advantage of this code due to its easy implementation and clear explanation.

This research project can be extended and improved in two primary research directions. Firstly, the zero-day attack detection performance still has much room for improvement, as it is still an unsolved issue. Advanced unsupervised anomaly detection techniques and online adaptive approaches, such as Extreme Gradient Boosting Outlier Detection (XGBOD) and Performance Weighted Probability Averaging Ensemble (PWPAE), are promising solutions to improve zero-day attack detection performance. Secondly, as 6G networks are expected to be zero-touch networks that enable fully autonomous attack detection and recovery, Automated ML (AutoML) techniques should be deployed to realize automated intrusion detection. Although in IDS-ML, we have used HPO, an important procedure of AutoML, to automatically optimize ML models, there are still many other AutoML procedures that are worth exploring, such as automated data collection, automated data preprocessing, automated feature engineering, automated model selection, and automated model updating/concept drift adaptation.

CRediT authorship contribution statement

Li Yang: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Abdallah Shami:** Conceptualization, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partially supported by The Canadian Urban Transit Research & Innovation Consortium (CUTRIC). The authors thank Abdallah Moubayed, Ismail Hamieh, Gary Stevens, and Stephen DeRusett for their support in the original papers.

References

[1] S. Khan, E. Sivaraman, P.B. Honnavalli, Performance evaluation of advanced machine learning algorithms for network intrusion detection system, in: M. Dutta, C.R. Krishna, R. Kumar, M. Kalra (Eds.), Proc. Int. Conf. IoT Incl. Life, ICIIL 2019, Springer Singapore, Singapore, NITTTR Chandigarh, India, 2020, pp. 51–59.

[2] R. Zhao, G. Gui, Z. Xue, J. Yin, T. Ohtsuki, B. Adebisi, H. Gacanin, A novel intrusion detection method based on lightweight neural network for internet of things, IEEE Internet Things J. (2021) 1, http://dx.doi.org/10.1109/JIOT.2021.3119055.

[3] L. Yang, A. Moubayed, A. Shami, P. Heidari, A. Boukhtouta, A. Larabi, R. Brunner, S. Preda, D. Migault, Multi-perspective content delivery networks security framework using optimized unsupervised anomaly detection, IEEE Trans. Netw. Serv. Manag. 19 (2022) 686–705, http://dx.doi.org/10.1109/TNSM.2021.3100308.

[4] A.K. Dwivedi, Anomaly detection in intra-vehicle networks, 2022, pp. 1–11, arXiv http://arxiv.org/abs/2205.03537.

[5] A. Garg, P. Maheshwari, Performance analysis of snort-based intrusion detection system, ICACCS 2016-3rd Int. Conf. Adv. Comput. Commun. Syst. Bringing To Table, Futur. Technol. from Arround Globe. 01 (2016) 1–5, http://dx.doi.org/10.1109/ICACCS.2016.7586351.

[6] Mohana A. Vikram, Anomaly detection in network traffic using unsupervised machine learning approach, in: 2020 5th Int. Conf. Commun. Electron. Syst., 2020, pp. 476–479, http://dx.doi.org/10.1109/ICCES48766.2020.9137987.

[7] M. Injadat, A. Moubayed, A.B. Nassif, A. Shami, Machine learning towards intelligent systems: applications challenges, and opportunities, Artif. Intell. Rev. (2021) http://dx.doi.org/10.1007/s10462-020-09948-w.

[8] L. Yang, A. Shami, IoT data analytics in dynamic environments: From an automated machine learning perspective, Eng. Appl. Artif. Intell. 116 (2022) 1–33, http://dx.doi.org/10.1016/j.engappai.2022.105366.

[9] W. Zuo, D. Zhang, K. Wang, On kernel difference-weighted k-nearest neighbor classification, Pattern Anal. Appl. 11 (2008) 247–257, http://dx.doi.org/10.1007/s10044-007-0100-z.

[10] S. Rasoul, L. David, A survey of decision tree classifier methodology, IEEE Trans. Syst. Man. Cybern. 21 (1991) 660–674.

[11] R.A. Khalil, N. Saeed, M. Masood, Y.M. Fard, M.-S. Alouini, T.Y. Al-Naffouri, Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications, IEEE Internet Things J. (2021) 1, http://dx.doi.org/10.1109/JIOT.2021.3051414.

[12] K. Alsabti, S. Ranka, V. Singh, An efficient k-means clustering algorithm, 1997, p. 43, https://surface.syr.edu/eecs/43.

[13] L. Li, R.J. Hansman, R. Palacios, R. Welsch, Anomaly detection via a Gaussian Mixture Model for flight operation and safety monitoring, Transp. Res. Part C 64 (2016) 45–57, http://dx.doi.org/10.1016/j.trc.2016.01.007.

[14] F.T. Liu, K.M. Ting, Z.H. Zhou, Isolation forest, in: Proc. - IEEE Int. Conf. Data Mining, ICDM, 2008, pp. 413–422, http://dx.doi.org/10.1109/ICDM.2008.17.

[15] T.G. Dietterich, Ensemble methods in machine learning, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), in: LNCS, vol. 1857, 2000, pp. 1–15.

[16] J. Jiang, F. Liu, W.W.Y. Ng, Q. Tang, W. Wang, Q.-V. Pham, Dynamic incremental ensemble fuzzy classifier for data streams in green internet of things, IEEE Trans. Green Commun. Netw. (2022) 1, http://dx.doi.org/10.1109/TGCN.2022.3151716.

[17] M.M. Leonardo, T.J. Carvalho, E. Rezende, R. Zucchi, F.A. Faria, Deep feature-based classifiers for fruit fly identification (Diptera: Tephritidae), in: 2018 31st SIBGRAPI Conf. Graph. Patterns Images, 2018, pp. 41–47, http://dx.doi.org/10.1109/SIBGRAPI.2018.00012.

[18] L. Yang, A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice, Neurocomputing 415 (2020) 295–316, http://dx.doi.org/10.1016/j.neucom.2020.07.061.

[19] L. Yang, A. Moubayed, I. Hamieh, A. Shami, Tree-based intelligent intrusion detection system in internet of vehicles, in: 2019 IEEE Glob. Commun. Conf., 2019, pp. 1–6, http://dx.doi.org/10.1109/GLOBECOM38437.2019.9013892.

[20] L. Yang, A. Shami, LCCDE: A decision-based ensemble framework for intrusion detection in the internet of vehicles, in: 2022 IEEE Glob. Commun. Conf., 2022, pp. 1–6.

[21] L. Yang, A. Moubayed, A. Shami, MTH-IDS: A multitiered hybrid intrusion detection system for internet of vehicles, IEEE Internet Things J. 9 (2022) 616–632, http://dx.doi.org/10.1109/JIOT.2021.3084796.

[22] L. Yang, A. Shami, A transfer learning and optimized CNN based intrusion detection system for internet of vehicles, in: 2022 IEEE Int. Conf. Commun., 2022, pp. 1–6, http://dx.doi.org/10.1109/ICC45855.2022.9838780.

[23] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, pp. 1–14, http://arxiv.org/abs/1409.1556.

[24] B. Chopard, M. Tomassini, Particle swarm optimization, Nat. Comput. Ser. (2018) 97–102, http://dx.doi.org/10.1007/978-3-319-93073-2_6.

[25] F. Pedregosa, et al., Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[26] C.R. Harris, et al., Array programming with NumPy, Nature 585 (2020) 357–362, http://dx.doi.org/10.1038/s41586-020-2649-2.

[27] W. McKinney, Data structures for statistical computing in Python, in: Proc. 9th Python Sci. Conf., vol. 1, 2010, pp. 56–61, http://dx.doi.org/10.25080/majora-92bf1922-00a.

[28] T. Chen, T. He, Xgboost: extreme gradient boosting, R packag. Version 0.4-2, 2015, pp. 1–4.

[29] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.Y. Liu, LightGBM: A highly efficient gradient boosting decision tree, Adv. Neural Inf. Process. Syst. 2017 (2017) 3147–3155.

[30] L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush, A. Gulin, Catboost: Unbiased boosting with categorical features, Adv. Neural Inf. Process. Syst. 2018 (2018) 6638–6648.

[31] S. Egea, A. Rego Manez, B. Carro, A. Sanchez-Esguevillas, J. Lloret, Intelligent IoT traffic classification using novel search strategy for fast-based-correlation feature selection in industrial environments, IEEE Internet Things J. 5 (2018) 1616–1624, http://dx.doi.org/10.1109/JIOT.2017.2787959.

[32] T. Head, MechCoder, G. Louppe, E. Al, Scikit-optimize/scikit-optimize: v0.5.2, 2018, http://dx.doi.org/10.5281/zenodo.1207017.

[33] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, D.D. Cox, Hyperopt: A python library for model selection and hyperparameter optimization, Comput. Sci. Discov. 8 (2015) http://dx.doi.org/10.1088/1749-4699/8/1/014008.

[34] J. Montiel, M. Halford, S.M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H.M. Gomes, J. Read, T. Abdessalem, A. Bifet, River: Machine learning for streaming data in python, J. Mach. Learn. Res. 22 (2021) 1–8.

[35] I. Sharafaldin, A. Habibi Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, 2018, pp. 108–116, http://dx.doi.org/10.5220/0006639801080116.

[36] B. Groza, P.S. Murvay, Efficient intrusion detection with bloom filtering in controller area networks, IEEE Trans. Inf. Forensics Secur. 14 (2019) 1037–1051, http://dx.doi.org/10.1109/TIFS.2018.2869351.