






WHITE: Weighted Hoeffding Tree Ensemble for Network Attack Detection at Fog-IoMT

Shilan S. Hameed^{1,2} , Ali Selamat^{1,3,4,5} , Liza Abdul Latiff⁶, Shukor A. Razak³, and Ondrej Krejcar^{1,5} 

¹ Malaysia-Japan International Institute of Technology (MJIIT), University Teknologi Malaysia, 54100 Kuala Lumpur, Malaysia

hameed.s@graduate.utm.my, asemamat@utm.my

² Directorate of Information Technology, Koya University, 44023 Koya, Iraq

³ School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Skudai, Johor Bahru 81310, Johor, Malaysia

shukorar@utm.my

⁴ Media and Games Center of Excellence (MagicX), Universiti Teknologi Malaysia, Skudai 81310, Johor Bahru, Malaysia

⁵ Center for Basic and Applied Research, Faculty of Informatics and Management, University of Hradec Kralove, Rokitanskeho 62, Hradec Kralove 50003, Czech Republic

ondrej.krejcar@uhk.cz

⁶ Razak Faculty of Technology and Informatics, Universiti Teknologi Malaysia, 54100 Kuala Lumpur, Malaysia

liza.kl@utm.my

Abstract. The fog-based attack detection systems can surpass cloud-based detection models due to their fast response and closeness to IoT devices. However, current fog-based detection systems are not lightweight to be compatible with ever-increasing IoMT big data and fog devices. To this end, a lightweight fog-based attack detection system is proposed in this study. Initially, a fog-based architecture is proposed for an IoMT system. Then the detection system is proposed which uses incremental ensemble learning, namely Weighted Hoeffding Tree Ensemble (WHITE), to detect multiple attacks in the network traffic of industrial IoMT system. The proposed model is compared to six incremental learning classifiers. Results of binary and multi-class classifications showed that the proposed system is lightweight enough to be used for the edge and fog devices in the IoMT system. The ensemble WHITE took trade-off between high accuracy and low complexity while maintained a high accuracy, low CPU time, and low memory usage.

Keywords: Intrusion detection · Machine learning · Incremental ensemble classifier. Fog-computing · Attack detection

1 Introduction

The Internet of Things (IoT) is a rapidly evolving technology that uses networking to connect infrastructure, digital devices, physical objects, applications, and persons [1].

The Internet of Medical Things (IoMT) is a use of the Internet of Things (IoT) in the health care sector [2, 3]. It is undeniable that smart medical gadgets have made life simpler and healthier for many people. However, security and privacy issues in the IoMT system remain unsolved issue [4, 5]. Hence, cyber-attack detection systems are considered as a defensive layer for the IoMT devices and networks. Machine learning and deep learning have been employed for intrusion and cyber-attack detection for the IoMT system. Solutions include on gadgets embedded models to cloud based systems. However, the chips and gadgets are not much efficient to hold the models and the IoMT network data. Additionally, cloud-based systems are centralized, and their detection is associated with delay. Hence, new approaches of network cyber-attack detection is required to overcome those limitations.

Fog computing is a novel concept that was developed to address the cloud's latency, centralization, and privacy problems [6]. Some cloud computing responsibilities will be moved closer to the smart devices in fog oriented IoT [7]. Moreover, a fog node might serve as the initial defense line for small devices that lack security features [8]. Fog-based attack detection is not widely used, especially for the IoMT system. Few studies have proposed a fog-based detection system. The authors in [9] presented a distributed Intrusion Detection System (IDS) which works based on fog-computing principle. Their system is designed for smart medical system that uses an online method specifically sequential extreme learning machine (EOS-ELM). They demonstrated that their proposed system is superior to cloud-based systems regarding detection time and true positive rate. In another study [10], the authors have used an ensemble learning for binary network cyber-attack detection using ensemble of (Decision Tree, Naive Bayes, and Random Forest) and XGBoost classifiers in the IoMT system following fog-cloud architecture. Because their system is too heavy for fog devices, they recommend using cloud computing for training and fog computing for testing. Another study [11] employs an ensemble incremental learning technique for fog devices for network intrusion detection in medical IoT networks. However, the dataset utilized isn't a recent IoT. Then, based on the current research gaps, a fog-based attack detection system is proposed using incremental ensemble learning for the IoMT system. The proposed system has two-folds advantages; firstly, the cyber-attacks will be detected accurately and soon they appear; secondly, the system is lightweight and does not use many resources.

2 Methodology

2.1 Datasets

We used two datasets of NSL-KDD and ToN-IoT. The NSL-KDD is a well-known benchmarking dataset, which was originally developed for conventional network and used by many researchers [12]. Hence, we have included this for comparison purpose. The dataset has 41 features and total of 148,517 samples in both train and test samples. The dataset contains Normal, Denial of Service (DoS), Probe, User to Root (u2r), and Root to Local (r2l) samples. The detail of each class count is shown in Fig. 1. The second dataset is a new cyber-attack dataset which was developed for IoT and IIoT systems. The dataset was built in a real-world IoT network context, using seven different sensors and telemetry services. As a result, the dataset exemplifies the IoT system's diversity. The dataset is the

IoT system’s network traffic, converted into NetFlow files [13]. NetFlow format is lighter than payload data as it only uses metadata instead of the packet contents. Additionally, since it does not use the packets holding the patient’s data, it does not violate the privacy rules, making the approach more compatible with an IoMT system. This version of data was curated and most informative records and features are selected to achieve a high performance [13]. The number of data records is 1,379,274, while the feature count is 13. There are multiple attacks available in the dataset as their sample counts are shown in Fig. 1.

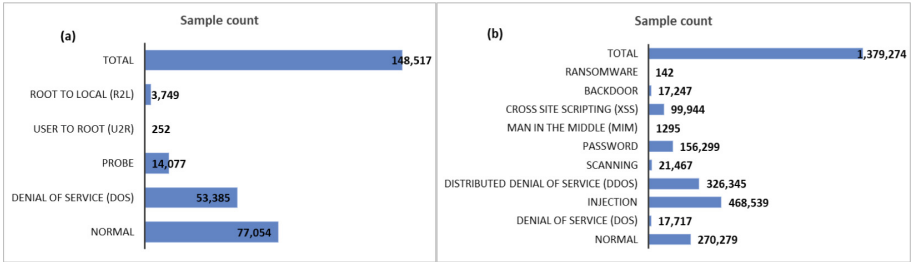


Fig. 1. The sample count of the attacks and normal class in the (a) NSL-KDD and (b) ToN-IoT_NetFlow datasets.

2.2 A Lightweight Network Attack Detection System

The proposed system in this study follows the guidelines of fog-computing architecture by IEEE [8]. Figure 2 illustrates the proposed system.

The absent security measures are shown by the red alert symbol. As a result of security absence, the medical devices and their network communication at edge-fog layers are exposed to various attacks. The amount of network data arriving through fog devices will grow with time, resulting in massive data, but fog devices are inefficient at storing it. As a result, training the data in stages would be preferable to retrain the whole data every time they aggregate. We have used a sliding window setting to train the classifiers incrementally. Unlike batch learning which uses cross-validations and hold-out, in our online learning a prequential evaluation was utilized, which uses the samples incrementally to train and test each sample record at a time. In this experiment, the maximum memory was set to 5 thousand samples, while the sliding window was set to 1000 samples at a time.

Compared Incremental Classifiers. In this study, a collection of single incremental classifiers and Bagging Hoeffding Tree ensemble was utilized to be compared to the proposed WHITE ensemble model. Each of them was deployed with their best tuned parameters using the same experimental environment. The following list is the utilized single classifiers:

- Incremental K-Nearest Neighbor (IKNN) [6].

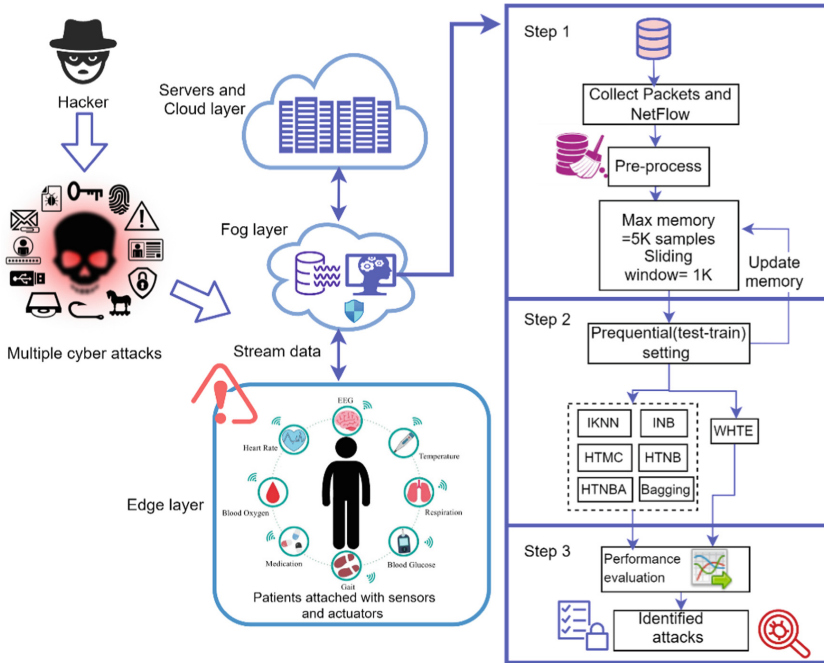


Fig. 2. The proposed fog-based network attack detection system and its architecture.

- Incremental Naïve Bayes (INB) [14].
- Hoeffding Tree-based Majority Class (HTMC) [15].
- Hoeffding Tree Naïve Bayes (HTNB) [15].
- Hoeffding Tree Naïve Bayes Adaptive (HTNBA) [16].
- Bagging Hoeffding Tree [17]

Weighted Majority Hoeffding Tree Ensemble (WHITE). The previously mentioned single classifiers may not produce high performance when the data is heterogeneous such as the IoT data. Hence, the ensemble of the single classifiers could maximize their performance and minimize their weakness. As a result, we propose an ensemble strategy in which a collection of single classifiers, particularly distinct types of Hoeffding Tree classifiers, are combined (HTMC, HTNB, HTNBA). Figure 3 depicts a summary of our ensemble technique flowchart. The ensemble is called Weighted Hoeffding Tree Ensemble (WHITE), which uses a weighted majority approach. It considers all of the classifiers’ decisions equally at the beginning [18]. It will, however, penalize a classifier if they make a wrong decision by not treating their decisions as significant as they formerly were [19]. The overall performance of the ensemble is the maximum because the errors created by the entire algorithm will essentially be the same as a constant error made by the best approach. When the expert makes a mistake in the initial weighted majority algorithm, the weighted value is doubled by $\frac{1}{2}$. As a result, the error bound equation is as follows:

$$M \leq 2.41(m + \log N) \tag{1}$$

where, m is the total of mistakes of the best classifier, M is the total of mistakes of the ensemble, and N is the total number of single classifiers. A randomized form of weighted majority algorithm can be used to reduce the error value to a minimum, which reduces the error equation’s constant value to close to one by adding (Beta β) to the equation. Hence, for the WHITE ensemble, the error equation can be defined as follow:

$$M \leq \frac{m \ln (1/\beta) + \ln N}{1 - \beta} \tag{2}$$

The value of β is set to be 0.5. Hence, the value of M for each iteration or a sample at a time will be counted as follow:

$$M \leq 1.39m + 2 \ln N \tag{3}$$

Performance Metrics. Multiple metrics are used to evaluate the proposed method in the current study. The detail of each metric is given in Table 1.

Table 1. The utilized evaluation metrics for evaluating the proposed method.

Metric	Discretion
Average accuracy	It is the average of all the sliding windows’ accuracies. Its equation is given below $\text{Average accuracy} = \frac{\sum_i acc}{N} \dots (4)$ While acc is the accuracy per each i sliding window over N total of the sliding windows
Average time (s)	The cumulative learning method’s average CPU time for all sliding windows
Average memory (MiB)	It is the average memory usage taken by each method for all datasets while considering the device’s memory

3 Results and Discussion

First, the proposed methods were evaluated on NSL-KDD dataset for the purpose of comparison with literature. As shown in Table 2, the WHITE ensemble outperformed the other single and ensemble classifiers with a high accuracy of 98.0%. Also, it is an ensemble method which recorded lower memory usage and CPU time compared to the Bagging ensemble.

After that the proposed model was evaluated on the ToN-IoT_NetFlow dataset using binary and multi-class classification. In the binary classification, the results were much better than multi-class classification, as expected. This is because multi-classification of 10 (refer to Fig. 1) classes in incremental fashion reduces the accuracy. Table 3 shows

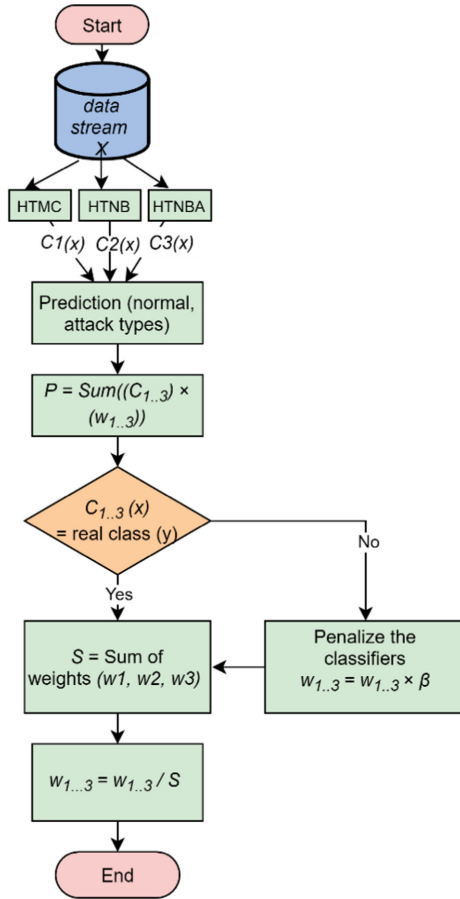


Fig. 3. The flowchart of the proposed ensemble WHITE method.

Table 2. An average performance of the WHITE classifier compared to the other single and ensemble classifiers for the NSL-KDD dataset.

Method	Average accuracy (%)	Average time (s)	Average memory (MiB)
IKNN	96.94	61.88	2.46
INB	87.83	1.83	0.03
HTMC	94.12	3.94	1.71
HTNB	97.0	3.29	1.71
HTNBA	97.50	3.19	1.71
Bagging	95.0	20.97	16.40
WHITE	98.0	8.94	5.19

that the model’s average accuracy for the ensemble WHITE was 100%. In addition, Bagging had 99.40% average accuracy and took the second place. The average memory use for the WHITE technique was 0.37 MiB and Bagging recorded the highest of 8.63 MiB while the HTNBA, HTNB, and HTMC methods used 0.08 MiB each. The average CPU time required to identify all intrusions was just 12.89 s for the WHITE technique, while Bagging needed 77.49 s. The IKNN approach, on the other hand, has highest complexity. In the multiclass classification, WHITE again took tradeoff between accuracy and complexity. Table 3 shows that the proposed ensemble had higher accuracy than single classifiers. Although its accuracy was slightly better than Bagging, its time and memory complexity were much lower. This is what we need for the lightweight devices.

Table 3. An average performance of the WHITE classifier compared to the other single and ensemble classifiers for ToNIoT-Netflow dataset using binary and multiclass classification

Method	Average accuracy (%)	Average time (s)	Average memory (MiB)	Average accuracy (%)	Average time (s)	Average memory (MiB)
Binary classification			Multiclass classification			
IKNN	98.79	184.69	1.15	70.50	255.79	1.25
INB	97.62	8.47	0.22	60.03	15.08	0.24
HTMC	99.01	3.90	0.08	70.82	24.92	9.85
HTNB	98.94	5.75	0.08	69.16	26.84	9.85
HTNBA	99.01	5.02	0.08	70.07	27.82	9.87
Bagging	99.40	77.49	8.63	71.08	299.89	86.52
WHITE	100.00	12.89	0.37	72.01	115.85	29.78

For the rest of the analysis, we have chosen the results of binary classification due to avoiding multiple and duplicate figures. To see the effect of concept drift on each classifier, we ha each technique’s incremental accuracy per five thousand records is conceptualized. From Fig. 4, it can be observed that the INB classifier was sensitive to the concept drift, and it had instability in its accuracy. Comparably, the rest of the classifiers looked much more stable due to the figure’s high variance in INB accuracy.

Hence, to see the other classifiers’ performance, INB is removed from the illustration presented in Fig. 5. It was seen that the HTNB and IKNN were more sensitive to the changes in the data, and their accuracy was constantly changing. Notably, the WHITE classifier showed a stable accuracy of 100% for each frequent sample. Moreover, Bagging, HTMC and HTNBA performed better instability than the rest of the classifiers.

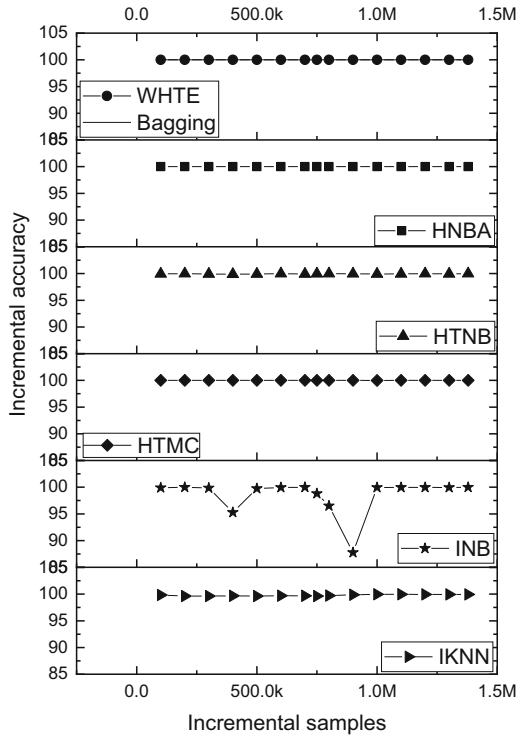


Fig. 4. The incremental accuracy per 5K sliding window samples. The accuracy was averaged for every 100K samples for clear visualization.

In terms of total CPU time per each sample frequencies, a 3D waterfall color surface was drawn, as demonstrated in Fig. 6. It is obvious that IKNN’s and Bagging’s CPU time were significantly impacted by the rising arrived samples to the system, in which the surface color rises from red to dark blue. Though, for other classifiers the CPU time was risen linearly with the increased samples.

A comparison has been made between the current work and related studies, as shown in Table 4. The proposed system outperformed the previous studies. Additionally, the current system is lightweight, and the system’s complexity is comprehensively analyzed, while previous studies were not lightweight nor considered these metrics for their evaluation.

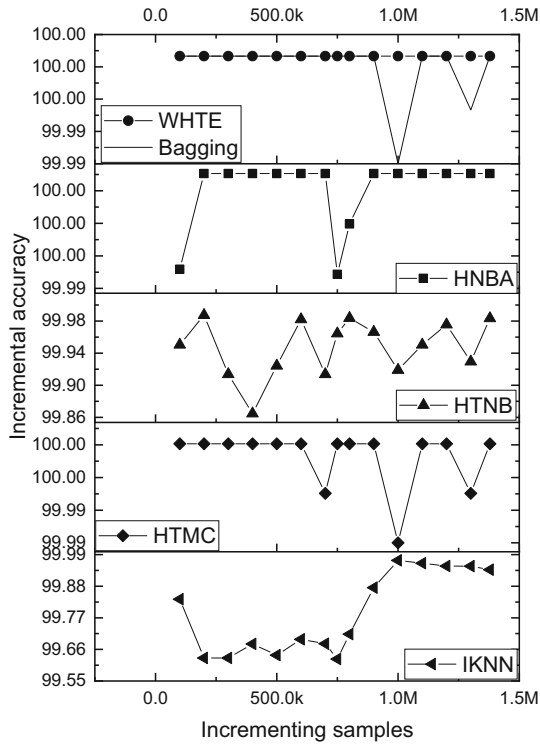


Fig. 5. Except for INB, the incremental accuracy of the utilized classifiers per 5K sliding window samples. The accuracy was averaged for every 100K samples for clear visualization.

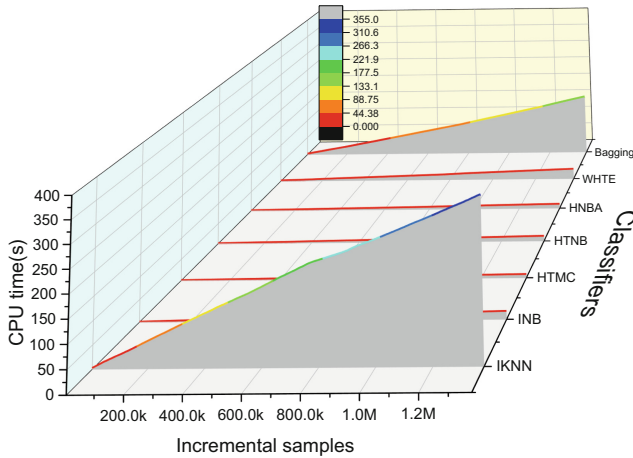


Fig. 6. The incremental CPU time per each subset of data samples for each classifier represented by a 3D waterfall colormap surface

Table 4. A comparison between the proposed system and related fog-based attack detection systems.

Ref	Architecture	Lightweight	Device specs	Best testing accuracy (%)	Type/Name of dataset	Type of learning	Complexity metrics	Splitting method
[10]	Cloud-Fog	No	CPU 2.20 GHz (10 cores, 13.75 MB L3 Cache), and 128 GB RAM	96.35	Network packet/ToNIoT	Batch	Not considered	Holdout Train-test (80:20)
[9]	Fog	No	Intel core i7 CPU processor and 16 GB RAM	98.19	Network packet/NSL-KDD	Batch	Not considered	Holdout Train-test (80:20)
[11]	Fog	No	CPUs \approx 2.2GHz and 8GB RAM	98.95	Network packet/NSL-KDD	Incremental batch	Not considered	Cross-validation
This work	Edge-Fog	Yes	CPUs \approx 2.2GHz (4 cores, 3 MB L3 Cache), and 8GB RAM	100.00	Network packet and NetFlow/NSL-KDD and ToNIoT	Incremental	Considered	Prequential sliding window evaluation

4 Conclusions

In this study, a lightweight network attack detection was proposed for the fog devices of the IoMT system. For this purpose, we have proposed a fog-based architecture and an incremental ensemble called WHITE which its performance was compared to another six incremental learning methods. It was seen that the system detects attacks with high accuracy of 100.0. In addition to that, the model is considered lightweight as it uses less low memory and CPU time. As a result, the proposed approach surpassed the earlier conducted solutions.

Acknowledgments. The authors sincerely thank Universiti Teknologi Malaysia (UTM) under Malaysia Research University Network (MRUN) Vot 4L876, for the completion of the research. This work was also partially supported/funded by the Ministry of Higher Education under the Fundamental Research Grant Scheme (FRGS/1/2018/ICT04/UTM/01/1) and Universiti Tenaga Nasional (UNITEN). The work and the contribution were also supported by the SPEV project “Smart Solutions in Ubiquitous Computing Environments”, University of Hradec Kralove, Faculty of Informatics and Management, Czech Republic (under ID: UHK-FIM-SPEV-2022–2102). We are also grateful for the support of student Michal Dobrovolny in consultations regarding application aspects.

References

1. Farahani, B., Firouzi, F., Chang, V., Badaroglu, M., Constant, N., Mankodiya, K.: Towards fog-driven IoT eHealth: promises and challenges of IoT in medicine and healthcare. *Futur. Gener. Comput. Syst.* **78**, 659–676 (2018)
2. Alsubaei, F., Abuhussein, A., Shiva, S.: A framework for ranking IoMT solutions based on measuring security and privacy. In: Arai, K., Bhatia, R., Kapoor, S. (eds.) *FTC 2018, AISC*, vol. 880, pp. 205–224. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-02686-8_17
3. He, D., Ye, R., Chan, S., Guizani, M., Xu, Y.: Privacy in the Internet of Things for smart healthcare. *IEEE Commun. Mag.* **56**(4), 38–44 (2018)
4. Yaacoub, J.-P.A., et al.: Securing internet of medical things systems: limitations, issues and recommendations. *Futur. Gener. Comput. Syst.* **105**, 581–606 (2020). <https://doi.org/10.1016/j.future.2019.12.02812>
5. Rathore, H., Al-Ali, A.K., Mohamed, A., Du, X., Guizani, M.: A novel deep learning strategy for classifying different attack patterns for deep brain implants. *IEEE Access* **7**, 24154–24164 (2019)
6. Hameed, S.S., et al.: A hybrid lightweight system for early attack detection in the IoMT fog. *Sensors* **21**(24), 8289 (2021)
7. Cisco, C.: Fog computing and the Internet of Things: extend the cloud to where the things are. [Электронный ресурс]. https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf (дата обращения: 10.03. 2019) (2015)
8. Group, O.C.A.W.: OpenFog reference architecture for fog computing. *OPFRA001* **20817**, 162 (2017)
9. Alrashdi, I., Alqazzaz, A., Alharthi, R., Aloufi, E., Zohdy, M.A., Ming, H.: FBAD: fog-based attack detection for IoT healthcare in smart cities. In: 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). IEEE, pp. 0515–0522 (2019)

10. Kumar, P., Gupta, G.P., Tripathi, R.: An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks. *Comput. Commun.* **166**, 110–124 (2021). <https://doi.org/10.1016/j.comcom.2020.12.003>
11. Hameed, S.S., Hassan, W.H., Latiff, L.A.: An efficient fog-based attack detection using ensemble of MOA-WMA for Internet of Medical Things. In: Saeed, F., Mohammed, F., Al-Nahari, A. (eds.) *IRICT 2020. LNDECT*, vol. 72, pp. 774–785. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-70713-2_70
12. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications, pp. 1–6. IEEE (2009)
13. Sarhan, M., Layeghy, S., Moustafa, N., Portmann, M.: Netflow datasets for machine learning-based network intrusion detection systems. arXiv preprint [arXiv:2011.09144](https://arxiv.org/abs/2011.09144) (2020)
14. Alaei, P., Noorbehbahani, F.: Incremental anomaly-based intrusion detection system using limited labeled data. In: 2017 3th International Conference on Web Research (ICWR), pp. 178–184. IEEE (2017)
15. Gama, J., Medas, P., Rodrigues, P.: Learning decision trees from dynamic data streams. In: *Proceedings of the 2005 ACM Symposium on Applied computing*, pp. 573–577 (2005)
16. Holmes, G., Kirkby, R., Pfahringer, B.: Stress-testing hoeffding trees. In: Jorge, A.M., Torgo, L., Brazdil, P., Camacho, R., Gama, J. (eds) *European Conference on Principles of Data Mining and Knowledge Discovery, PKDD 2005. Lecture Notes in Computer Science*, vol. 3721. Springer, Berlin, Heidelberg (2005). https://doi.org/10.1007/11564126_50
17. Oza, N.C., Russell, S.J.: Online bagging and boosting. In: *International Workshop on Artificial Intelligence and Statistics*, pp. 229–236. PMLR (2001)
18. Kolter, J.Z., Maloof, M.A.: Dynamic weighted majority: an ensemble method for drifting concepts. *J. Mach. Learn. Res.* **8**, 2755–2790 (2007)
19. Littlestone, N., Warmuth, M.K.: The weighted majority algorithm. *Inf. Comput.* **108**(2), 212–261 (1994)