sciendo

BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **22**, No 3
Sofia • 2022 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2022-0033

A New Network Digital Forensics Approach for Internet of Things Environment Based on Binary Owl Optimizer

Hadeel Alazzam[1], *Orieb AbuAlghanam*[2], *Qusay M. Al-zoubi*[1], *Abdulsalam Alsmady*[3], *Esra'a Alhenawi*[4]

[1]*Department of Intelligence Systems, Al-Balqa Applied University, Al-Salt, Jordan*
[2]*Department of Networks and Information Security, Al-Ahliyya Amman University, Amman, Jordan*
[3]*Department of Computer Engineering, Jordan University of Science and Technology, Irbid, Jordan*
[4]*Department of Computer Science, The University of Jordan, Amman, Jordan*
E-mails: hadeel.alazzam@bau.edu.jo O.AbuAlghanam@ammanu.edu.jo qusayz@bau.edu.jo aralsmady15@cit.just.edu.jo esra_a_2008@live.com

***Abstract*:** *The Internet of Things* (*IoT*) *is widespread in our lives these days* (*e.g., Smart homes, smart cities, etc.*). *Despite its significant role in providing automatic real-time services to users, these devices are highly vulnerable due to their design simplicity and limitations regarding power, CPU, and memory. Tracing network traffic and investigating its behavior helps in building a digital forensics framework to secure IoT networks. This paper proposes a new Network Digital Forensics approach called* (*NDF IoT*). *The proposed approach uses the Owl optimizer for selecting the best subset of features that help in identifying suspicious behavior in such environments. The NDF IoT approach is evaluated using the Bot IoT UNSW dataset in terms of detection rate, false alarms, accuracy, and f-score. The approach being proposed has achieved 100% detection rate and 99.3% f-score and outperforms related works that used the same dataset while reducing the number of features to three features only.*

***Keywords*:** *Cybersecurity, Internet of things, Intrusion detection, Feature selection, Network Digital Forensics.*

1. Introduction

Internet of Things (IoT) is a concept for a network of physical devices of surrounded human things of living space [1]. These devices are equipped with a Unique IDentifier (UID) and have the ability to transfer data through the network without any human interaction [2, 3]. Due to widespread of IoT in our lives and the level of simplicity of these devices make them vulnerable to be infiltrated [4, 5]. Thus, IoT devices can be easily manipulated and hacked rather than other laptops/PCs devices, which make it a target for establishing Distributed Denial of Service (DDoS) attacks [6]. Denial of Service (DoS) is a type of cyberattack, where the attacker overloads a

target system or network connected to the Internet by flooding the system with excessive requests to make it unavailable for its intended users [7].

One of the most popular cyberattacks is a botnet attack [8, 9]. The word "botnet" is a combination of two words "bot" which refers to the robot and "net" that refers to the network [10]. A botnet is a collection of interconnected devices infected by malware that gives the criminal privilege to control them [7]. Infected devices by a botnet are hard to detect, since the botnet does not use much computing power and the device can still function normally [11]. In 2016, the Mirai botnet was the largest botnet attack have been launched, where the criminal infects more than 500,000 IoT devices to launch a DDoS attack that aimed to shutdown Netflix [12].

Defending against cyberattacks in IoT environment is challenging [13]. Since, there is not a single standard architecture design for IoT, which mean there are many protocols such as Zonal Intercommunication Global-standard (Zigbee), Long Range Radio (LoRa), and Message Queuing Telemetry Transport (MQTT) that can interact together. This results in increasing the heterogeneity and complexity of the system [14]. Moreover, new attacks that rely on zero-day exploits have been preferred by attackers. These attacks cannot be avoided using traditional security countermeasures. With the heterogeneity of the IoT deployment models, there is a need for an effective network digital forensics framework that will help in tracing and investigating such attacks.

Digital forensics can be defined as using the scientifically derived and proven methods by obtaining intelligence from digital evidence to using it in investigations, or in criminal procedures [15]. Digital evidence plays a vital role in solving digital forensic cases. The National Institute of Justice definition of the digital evidence can be summarized as evidence that refers to information and data of value to an investigation that is received, stored on, or transmitted by digital devices [16].

According to [17], digital Forensics can be classified into multiple categories with respect to its domain; Disk Forensics, network forensics, memory forensics, cloud forensics, and mobile Android forensics. In this paper, we will focus on network forensics, since all IoT services are based on network connections.

Network forensics focuses on the security incidents in networks; the digital evidence for network forensics can be collected by analysing and investigating network traffic. Experts can use several tools to capture the traffic (e.g., Wireshark) [18, 19]. After collecting the data, the characters or features of packets should be selected to identify the malicious packet [20]. Researchers use several feature selection algorithms and methods to select the features that help in identifying traffic class [21]. In this paper, a modified binary Owl optimizer is used to select the optimal subset of features from the collected dataset.

This paper aims to develop a Network Digital Forensics (NDF) that investigates and traces traffic in IoT environments that helps in blocking any suspicious or anomaly packet that passes the network. The NDF approach uses a modified binary Owl optimizer to select the optimal subset of features that help in the investigation process.

The following points summarize the main contributions of this paper:

- Propose a Network Digital Forensics Approach for IoT environment.

- Summarize some of the state-of-the-art related work related to network digital forensics.
- Using a modified binary Owl optimizer to select the optimal subset of features that affect the investigation process.
- Evaluate the proposed (NDF) approach using BOT-IoT UNSW dataset in terms of Detection Rate and False Alarm.

The rest of this paper is organized as follows; Section 2 summarizes some of the state-of-the-art works, Section 3 presents the used dataset and illustrates the used methodology, Section 4 discusses the results, and Section 5 concludes the paper. The Appendix presents list of abbreviations.

2. Related works

A new realistic dataset for forensics in IoT environment called Bot IoT is developed in [22]. The dataset contains both benign traffic for IoT and other network traffic and has four types of cyberattacks: information theft, probing, and Denial of Service (DoS). The probing attack is a malicious activity that aims to scan a remote machine in order to gather information. The probing attack is further classified into a passive and active probing attack. In an information theft attack, the attacker tries to gain data and download it on an authorized remote machine. Information theft attacks can be subcategories into data theft and keylogging. The developed dataset has been evaluated using three machine-learning algorithms Support Vector Machine (SVM), Recurrence Neural Network (RNN), and Long Short-Term Memory RNN (LSTM-RNN). The SVM achieves the best accuracy against other classifiers.

A Particle Deep Framework (PDF) for network forensics is proposed in [23]. The proposed framework aims to discover and trace cyber-attacks in IoT environment. The authors use the Particle Swarm Optimizer (PSO) to adapt the best hyper-parameters of deep learning. The proposed framework consists of Multi-layer Perception Neural Network that has been trained and evaluated using BOT IoT dataset.

A network forensic approach that based on neural network and Genetic algorithm for IoT is developed in [24]. The proposed approach uses the Genetic algorithm to optimize neural network parameters. Authors apply the proposed approach on BOT IoT, and use the recommended 10-best features by [22]. The approach being proposed evaluates in terms of accuracy, precision, recall, and f-score.

A network forensics model to detect and identify cyber-attacks is proposed in [25]. The model being proposed focuses on detecting flooding attacks and finding infected IoT Arduino Bluetooth devices. The authors use the Wireshark tool to collect p.cap files in addition to log data for investigation. The forensics model proposed is able to detect three IP addresses that have committed illegal actions, which had led to overload traffic.

A unified Intrusion Detection System (IDS) for the IoT environment is proposed in [27]. The designed IDS aims to defend the network from four popular types of attacks: DoS, generic, probe, and exploit. The IDS being proposed uses a set of rules

generated by five types of decision trees, and an information gain is used to select features. The five selected decision trees has been trained on 22 to 13 features based on a predetermined threshold. The IDS being proposed is evaluated using the UNSW-NB15 dataset and compared with ENAD and DENDRON proposed by [27, 28], respectively.

A three-layer intrusion detection system for smart home IoT devices is proposed in [29]. The first layer considers profiling the behavior of each IoT device in the network, the second layer determines benign traffic from normal, while the third layer defines the type of attack detected in layer two. The system has been evaluated in a smart home testbed with 8 IoT devices against 12 deployed attacks from four attack types namely: DoS, a man in the middle attack, replay, and reconnaissance. The proposed system has achieved 90% and 98% f-Score for layer two and layer three, respectively.

A two-level anomaly detection model for the IoT environment has been developed in [30]. The first level uses the decision tree classifier to define the benign traffic from anomalies; then the anomalies are sent to level two for a deeper investigation. Level two uses the recursive feature elimination to select the significant features, while using the Synthetic Minority Oversampling TEchnique (SMOTE) oversampling and a modified version of the nearest neighbour for cleaning the data. The model being proposed has been evaluated using CICIDS2017 and UNSW-NB15 datasets in terms of precision, recall, and f-Measure.

A supervised intrusion detection system for IoT is proposed in [31]. The proposed system aims to predict unknown types of attacks, where the training set consists of four types of attacks only, while the testing set contains 10 types of attacks. The system being developed uses a random forest classifier and has been evaluated using N-BaIoT dataset. The system being proposed has been able to detect the new attacks with a 99% detection rate and near-zero false alarms.

An intrusion detection system for IoT based on suppressed fuzzy clustering is proposed in [32]. The system being proposed starts by detecting by high frequency and low frequency. Then, the clustered data are analysed with Principal Component Analysis (PCA) and the features are eliminated, and finally detects the data with frequency self-adjustment. The results indicate that the proposed system enhances accuracy and reduces false alarms.

A specification heuristic intrusion detection system for the IoT environment is proposed in [33]. The proposed system is based on discovering a unique n-gram pattern for sequential attributes values. The system being proposed has been evaluated using UNSWNB15 and has achieved a high accuracy and detection rate compared with related works.

Based on the mentioned studies, most of the related works evaluate their proposed system using datasets like UNSWNB15, CICIDS2017, or simulated data. Not all those datasets have realistic IoT traffic. Only the study by [22] have proposed a realistic dataset for IoT and used it to evaluate their proposed system. In this paper, we will use the same dataset used by [22] and will compare our system with their system.

3. The proposed network forensics approach for IoT

The basic mechanism of the digital forensics' framework consists of six main stages as illustrated in Fig. 1: the identification stage, preservation stage, evidence collection stage, examination, and finally analysis and presentation stage. This model has been proposed by the first Digital Forensics Research Workshop [34]. All proposed models and digital forensics frameworks have been inspired by this model. The following clarifies each stage of the basic digital forensics model as shown in Fig. 1:

- **Identification Stage.** Involves the identification of possible evidence. At this stage, the number of possible evidence is constrained.
- **Preservation Stage.** Ensures the integrity of the data to be collected.
- **Collection Stage.** Ensures the usage of the appropriate tools and techniques to collect the data. The collected data must be identified as important data to the case based on the first stage.
- **Examination Stage.** Involves identifying traces and possible evidence.
- **Analysis Stage.** Interprets the identified evidence.
- **Presentation Stage.** Presents the finding of the investigation comprehensively.

In this paper, the network forensics approach being proposed coincides with the previous model, with minor revisions to make the model appropriate for investigation at the network level of an IoT environment. Moreover, since, IoT devices have constrained memory, that makes it irrelevant for investigation at the device level. All possible evidence will be collected from network traffic. Fig. 3 illustrates the proposed NDF approach. The next subsections clarify the components of the proposed approach.

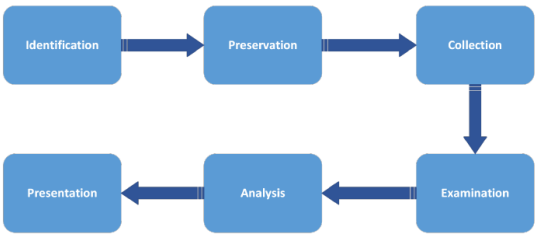


Fig. 1. Main stages of digital forensics model

3.1. Data collection and understanding

The proposed NDF IoT uses the BOT IoT dataset, which it a realistic dataset that have traffic from IoT and non-IoT devices. The BOT IoT dataset has 19 attributes [35, 36]. It contains IoT traffic of smart home, five devices have been involved: a smart garage door which opens or closes based on probabilistic input; a weather station that generates information about temperature, humidity, and air pressure; a smart fridge that regulates the fridge temperature automatically, when necessary; a smart thermostat which setup the home temperature by starting the air-conditioning system; and motion-activated lights which turns light on or on based on the motion-

sensor signal [23, 37, 38]. The dataset contains both benign traffic for IoT and other network traffic and has four types of cyberattacks: information theft, probing, Denial of Service [22]. Table 2 illustrates the set of features in the dataset with their description and type. In this paper, only 5% of the BOT IoT dataset have been used. Table 2 illustrates the data distribution for both training and testing set. As Table 2 shows that, the data is highly imbalanced.

3.2. Data examination

To examine the collected data, the data first should be prepared in an appropriate structure. The preprocessing procedures applied to BOT IoT dataset are four.

- **Eliminate extra attributes.** The developed IDS IoT aims to classify the network traffic into two classes (normal and attack). The Bot-IoT dataset contains two extra attributes that determine the category and the subcategory of attacks. The first step of preparing the dataset for IDS IoT is eliminating these extra attributes.
- **Label transfer and data transfer.** Here all nominal values are transformed to numeric values. Also, one of the attributes contains the IP address; the format of this field contains digits and dots, and this raises a problem with the panadas library in Python. To solve this issue, all dots are replaced by commas, and the field is transformed into a string.

Table 1. Set of features for the Bot-IoT dataset

No	Feature	Description	Type
1	N IN Conn P SrcIP	Number of inbound connections per source IP	Generated flow
2	N IN Conn P DstIP	Number of inbound connections per destination IP	
3	pkSeqID	Row identifier	Features
4	proto	Textual representation of transaction protocols presented in network flow	
5	saddr	Source IP address	
6	Sport	Source port number	
7	Daddr	Destination IP address	
8	dport	Destination port number	
9	seq	Argus sequence number	Network Flow Extracted
10	stddev	Standard deviation of aggregated records	
11	min	Minimum duration of aggregated records	
12	state number	Numerical representation of feature state	
13	mean	Average duration of aggregated records	
14	drate	Destination-to-source packets per second	
15	srate	Source-to-destination packets per second	
16	max	Maximum duration of aggregated records	
17	attack	Class label: 0 for Normal traffic, 1 for Attack Traffic	
18	category	Traffic category	
19	subcategory	Traffic subcategory	

- **Data normalization.** That is an important step to scale the values of each attribute into a unified scale; this will make all attributes treated equally by the classifier. The next equation presents the normalization formula [39]:

(1) $$X_{\text{normalized}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}.$$

- **Duplicate removal.** All duplicate records in the training set have been eliminated.

After preprocessing the data, an important step that affects the overall system accuracy and performance is feature selection. In this paper, a modified binary Owl optimizer is used for feature selection process. The next subsection presents the feature selection process using the modified Owl optimizer.

Table 2. 5% of the Bot-IoT dataset distribution

Data	Attack	Normal	Total
Training	1,048,457	118	1,048,575
Testing	733,598	107	733,705

3.2.1. Modified binary Owl optimizer for Feature selection

The Owl optimizer is a recent algorithm used for solving optimization problems. The Owl optimizer is inspired by nature and simulated the behaviour of the hunting Owl. The hunting Owl relies on their hearing to find their prey at night when sight is impossible in such circumstances. The first Owl search algorithm was first developed by [40]. Since the Owls' behaviour is based on their hearing, they have a special auditory system with vertical asymmetry of the ears, which makes the sound reach one ear before the other. This asymmetry of the ears helps in developing a special sound localization system to find the prey precisely [41]. Fig. 2 clarifies how the asymmetry of Owl ears facilitates the localization process towards the prey. The general approach of the Owl optimizer contains seven phases [42], as follow.

- **Initial population.** A random set of solutions can be viewed as a set of Owls in the forest. Where each solution/Owl is represented as a feature vector with a length equal to the total number of features of the examined dataset

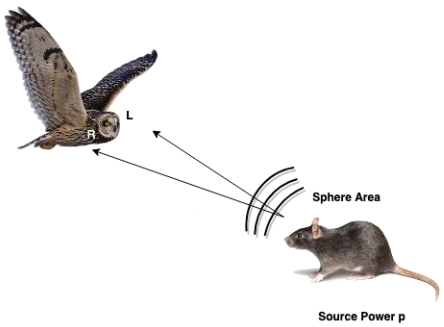


Fig. 2. Barn Owl hunting mechanism

- **Owl Evaluation.** To prefer one solution/Owl over another, all Owls in the population will be evaluated according to their fitness value. An evaluation function (i.e., Fitness function) is used in the evaluation process. The evaluation function is an evaluation criterion that is based on the problem being solved. Since the Owl optimizer is based on their auditory system, the fitness value will be presented with respect to the intensity information received by the Owl's ears as shown in Fig. 2. The next equation presents the intensity normalization of Owl O_i with respect to fitness value $f(O_i)$:

(2) $$\text{Intensity}(O_i) = \frac{F(O_i) - O_{\text{worst}}}{O_{\text{best}} - O_{\text{worst}}}.$$

- **Owl location update.** All Owls update their location toward the prey. The prey is near the Owl that has the best fitness value. The next equation presents the formula of the distance R_i between the Owl O_i and the prey,

(3) $$R_i = ||O_i, V||_2,$$

which will be used to calculate the intensity change IC_i toward the prey in the equation

(4) $$\mathrm{IC}_i = \frac{\mathrm{Intensity}\,(O_i)}{Ri^2} + \alpha,$$

where α is a random number between range [0, 0.5] and represents the noise. Based on both Equation (3) and (4) the Owl position will be updated by equation

(5) $$O_i{}^{t+1} = \begin{cases} O_i^t + \beta \times \mathrm{IC}_i \times ||\alpha V - O_i^t|| \; P_{\mathrm{vm}} < 0.5, \\ O_i^t - \beta \times \mathrm{IC}_i \times ||\alpha V - O_i^t|| \; P_{\mathrm{vm}} \geq 0.5, \end{cases}$$

where β is a linear decreasing number from 0 to 1.9 [41]. β introduces large changes initially and promotes the exploration of the search space, and P_{vm} is the probability of Owl movement. The binary version of the Owl optimizer is customized to fit the feature selection process. The modified binary version of the Owl optimizer has the same phases as the general Owl optimizer; however, the formula and the presentation of the solution has been modified. The following are the main steps for the modified binary version of the Owl optimizer for feature selection for digital forensics purposes:

- **Solution representation.** Each Owl in the population represent a solution, which is a binary fixed length vector. The length of the vector is equal to the number of features in the dataset. The "0" input indicates that the corresponding feature is not included in the solution, while "1" input indicates that the corresponding feature is included in the solution.
- **Modified intensity change.** Updating the Owl location in the binary version of the Owl optimizer is modified by calculation the distance between the Owl and the prey as the similarity between two binary vectors (i.e., the binary vector of the Owl and the binary vector of the prey) using the equation bellow. For example, the number of same features between "10011" and "00011" is 4, then the value of the R_i will be 0.8:

(6) $$\mathrm{Distance}\,(\mathrm{R}_i) = \frac{\#\ \mathrm{of\ Same\ Features}}{\mathrm{Length\ of\ feature\ vector}}.$$

Based on the similarity value the intensity change IC_i for Owl O_i will be calculated using the next equation, where α is a random number between [0, 0.5]:

(7) $$\mathrm{IC}_i = \frac{\mathrm{Intensity}\,(O_i)}{R_i} + \alpha.$$

- **Owl location update.** The Owl location will be updated according to

(8) $$O_i{}^{t+1} = \begin{cases} O_i^t, & O_{\mathrm{best}} < r, \\ O_{\mathrm{best}}, & O_i \geq r, \end{cases}$$

where r is a uniform random number.

- **Escape local optima.** All solutions/Owls are stored in a set. When there is more than one Owl/solution that are the same, then new Owl/solution will be generated randomly and join the set. In this way, the optimizer will have a chance to escape local optima.

Fig. 3 presents the overall proposed network forensics framework based on the Owl optimizer and ensemble learning. Random Forest classifier is used as an

ensemble learning to develop the final model. During the feature selection step, each solution will be used to train the model, then it will be evaluated according to the fitness function. The next equation presents the fitness function formula. The fittest solution is the one that has the minimum value against the others:

$$\text{FitnessFunction} = \text{FPR} + \frac{1}{\text{TPR}}, \tag{9}$$

where FPR is the False Positive Rate, and TPR is the True Positive Rate.

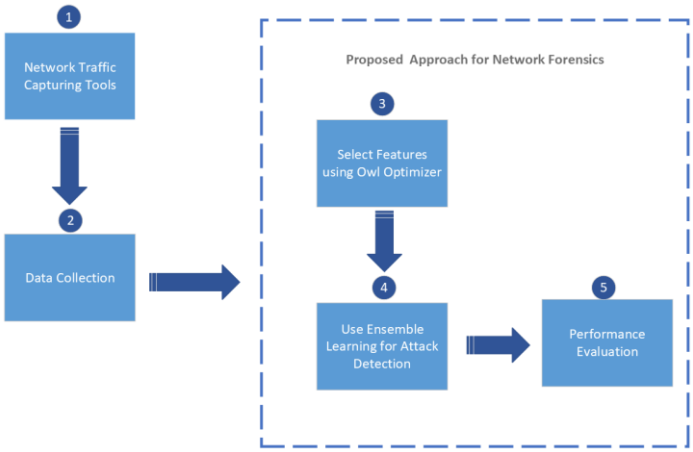


Fig. 3. The proposed network forensics approach using Owl optimizer

4. Results and discussions

4.1. Performance evaluation

Four main performance measures have been used to evaluate the proposed NDF IoT approach. All of the network forensics approaches having been proposed have been evaluated in terms of detection rate (True Positive Rate (TPR)) and false alarms (False Positive Rate (FPR)). Since the dataset used in this paper is highly imbalanced, f-Score measure and accuracy are used for better evaluation [43-45]. Table 3 illustrates the confusion matrix while the next four equations present the calculation for TPR, FPR, f-Score, and accuracy, respectively, based on the values from the confusion matrix:

$$\text{Detection Rate (TPR)} = \frac{\text{TP}}{(\text{TP}+\text{FN})}, \tag{10}$$

$$\text{False Alarms (FPR)} = \frac{\text{FP}}{(\text{TN}+\text{FP})}, \tag{11}$$

$$\text{f-Score} = \frac{2\times\text{TP}}{(2\times\text{TP}+\text{FP}+\text{FN})}, \tag{12}$$

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{(\text{TP}+\text{TN}+\text{FP}+\text{FN})}. \tag{13}$$

Table 3. Confusion matrix

Predicted type	Actually Attack (1)	Actually Normal (0)
Predicted Attack (1)	TP	FP
Predicted Normal (0)	FN	TN

4.2. Results

This section evaluates the proposed approach and compares it against all features in the dataset, and the recommended set of features in [22], the approach proposed in [24] and with the Particle deep learning approach proposed in [37]. The examined approaches are evaluated in terms of accuracy, TPR, FPR, F-score and number of features. Table 4 lists the selected approaches from the related works for evaluation. The three selected features by the Owl optimizer are “pkSeqID”, “seq”, “srate”. Fig. 4 illustrates the detection rate for all examined approaches in Table 4. The results show that the proposed approach achieves the highest detection rate against the other examined approaches. The LSTM achieved the second-best results in terms of detection rate with a value 0.997. While the approach used the Genetic with neural networks achieves the worst detection rate results with 0.938.

Table 4. Selected approaches for evaluation purpose using BOT IoT

Reference	Approach	Number of features
[22]	LSTM	10
[24]	Genetic and Neural Network (GNN)	10
[37]	Particle Swarm and Deep learning (PSD)	13
Proposed approach	Owl and Random Forest (OWL RF)	3

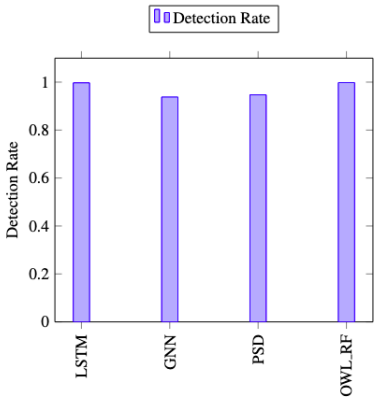


Fig. 4. Detection rate with all features for various classifiers

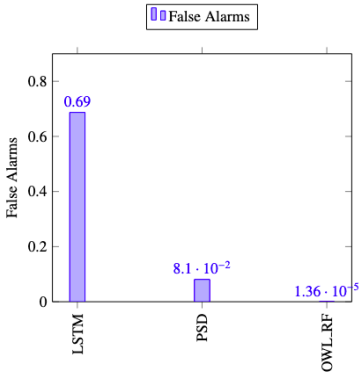


Fig. 5. False alarms results for all examined approaches

Fig. 5 illustrates the false alarm (FPR) results for all examined approaches in Table 4. The results show that the proposed approach achieves the lowest false alarm against the other examined approaches with 1.36×10^{-05} The LSTM achieves the worst false alarm results with a value 0.69. While the approach useing the Genetic with neural networks has not reported any false alarm result.

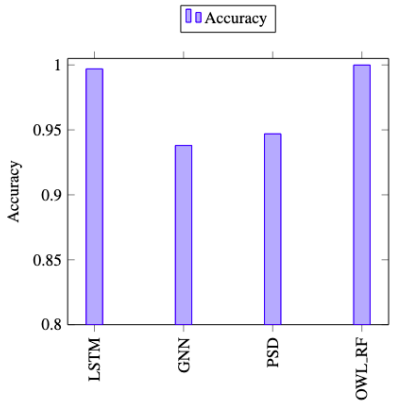


Fig. 6. Accuracy results for all examined approaches

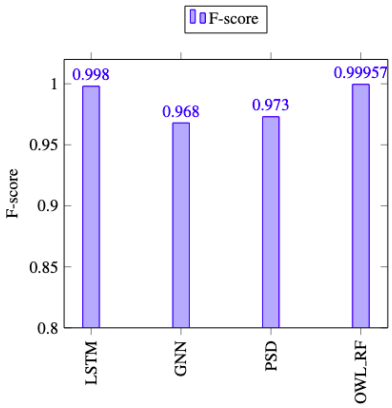


Fig. 7. f-Score results for all examined approaches

5. Conclusion

Nowadays, IoT devices have a significant impact on our lives, starting from smart home devices, wearable health monitoring devices to the industrial sector. Thus, they bring benefits and rais security issues. In this paper, a new network digital forensics approach for smart home devices is proposed. The proposed approach has been evaluated using Bot-IoT UNSW dataset in terms of detection rate, false alarms, accuracy and f-Score. It uses the Owl optimizer as a feature selection method to eliminate redundant and irrelevant features. The proposed approach reduces the features from 19 features to only three features which accelerated the training model

time and enhanced the detection rate. The results are promising and outperform related works.

6. Declaration of competing interest and data availability statement

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.

The data that support the findings of this study are openly available in UNSW Canberra at ADFA at **https://research.unsw.edu.au/projects/bot-iot-dataset**, reference number [22].

References

1. Nolin, J., N. Olson. The Internet of Things and Convenience. – Internet Research, Vol. **22**, No 2, pp. 361-376.
2. Abualghanam, O., L. Albdour, O. Adwan. Multimodal Biometric Fusion Online Handwritten Signature Verification Using Neural Network and Support Vector Machine. – Transactions, Vol. **12**, 2021, No 5, pp. 1691-1703.
3. Abualghanam, O., M. Qatawneh, W. Almobaideen. A Survey of Key Distribution in the Context of Internet of Things. – Journal of Theoretical and Applied Information Technology, Vol. **97**, 2019, No 22, pp. 3217-3241.
4. Abualghanam, O., M. Qatawneh, W. Almobaideen, M. Saadeh. A New Hierarchical Architecture and Protocol for Key Distribution in the Context of IoT-Based Smart Cities. – Journal of Information Security and Applications, Vol. **67**, 2022.
5. Castelo Ǵomez, J. M., J. Carrillo Monďejar, J. Roldďan Ǵomez, J. L. Marťınez Marťınez. A Context-Centered Methodology for IoT Forensic Investigations. – International Journal of Information Security, Vol. **20**, 2021, No 5, pp. 647-673.
6. Atamli, A. W., A. Martin. Threat-Based Security Analysis for the Internet of Things. – In: Proc. of International Workshop on Secure Internet of Things, IEEE, 2014, pp. 35-43.
7. Carl, G., G. Kesidis, R. R. Brooks, S. Rai. Denial-of-Service Attack-Detection Techniques. – IEEE Internet Computing, Vol. **10**, 2006, No 1, pp. 82-89.
8. Lysenko, S., O. Savenko, K. Bobrovnikova, A. Kryshchuk. Self-Adaptive System for the Corporate Area Network Resilience in the Presence of Botnet Cyberattacks. – In: Proc. of International Conference on Computer Networks, Springer, 2018, pp. 385-401.
9. Ozawa, S., T. Ban, N. Hashimoto, J. Nakazato, J. Shimamura. A Study of IoT Malware Activities Using Association Rule Learning for Darknet Sensor Data. – International Journal of Information Security, Vol. **19**, 2020, No 1, pp. 83-92.
10. Xing, Y., H. Shu, H. Zhao, D. Li, L. Guo. Survey on Botnet Detection Techniques: Classification, Methods, and Evaluation. – Mathematical Problems Engineering, Vol **2021**, 2021, pp. 1-24.
11. Regan, C., M. Nasajpour, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, K. K. R. Choo. Federated IoT Security Attack Detection Using Decentralized Edge Data. – Machine Learning with Applications, Vol. **8**, 2022, 100263.
12. Kumar, A., T. J. Lim. Early Detection of Mirai-Like IoT Bots in Large-Scale Networks through Sub-Sampled Packet Traffic Analysis. – In: Proc. of Future of Information and Communication Conference, Springer, Vol. **70**, 2019, pp. 847-867.
13. Meneghello, F., M. Calore, D. Zucchetto, M. Polese, A. Zanella. IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices. – Internet of Things Journal, Vol. **6**, 2019, No 5, pp. 8182-8201.
14. Datta, P., B. Sharma. A Survey on IoT Architectures, Protocols, Security and Smart City Based Applications. – In: Proc. of 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT'17), IEEE, 2017. pp. 1-5.

15. W u, T., F. B r e i t i n g e r, I. B a g g i l i. IoT Ignorance is Digital Forensics Research Bliss: a Survey to Understand IoT Forensics Definitions, Challenges and Future Research Directions. – In: Proc. of 14th International Conference on Availability, Reliability and Security, ACM, 2019, pp. 1-15.
16. S e d a, M., B. K. P. K r a m e r. A Comparison of US Forensic Accounting Programs with the National Institute of Justice Funded Model Curriculum. – Journal of Forensic & Investigative Accounting, Vol. **7**, 2015, No 2, pp. 144-177.
17. P a u l J o s e p h, D., J. N o r m a n. An Analysis of Digital Forensics in Cyber Security. – In: Proc. of 1st International Conference on Artificial Intelligence and Cognitive Computing, Springer; 2019, pp. 701-708.
18. J o r d a a n, J. The Role of In Cybercrime Investigation Digital Forensics. – Servamus Community-Based Safety and Security Magazine, Vol. **112**, 2019, No 10, pp. 33-37.
19. S o n m e z, Y. U., A. V a r o l. Review of Evidence Collection and Protection Phases in Digital Forensics Process. – International Journal of Information Security Science, Vol. **6,** 2017, No 4, pp. 39-45.
20. P r a k a s h, A., R. P r i y a d a r s h i n i. An Intelligent Software Defined Network Controller for Preventing Distributed Denial of Service Attack. – In: Proc. of 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT'18), IEEE, 2018, pp. 585-589.
21. S h a f i q, M., Z. T i a n, A. K. B a s h i r, X. D u, M. G u i z a n i. IoT Malicious Traffic Identification Using Wrapper-Based Feature Selection Mechanisms. – Computers & Security, Vol. **94**, 2020, 101863.
22. K o r o n i o t i s, N., N. M o u s t a f a, E. S i t n i k o v a, B. T u r n b u l l. Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-Iot Dataset. – Future Generation Computer Systems, Vol. **100**, 2019, pp.779-796.
23. K o r o n i o t i s, N., N. M o u s t a f a, E. S i t n i k o v a. A New Network Forensic Framework Based on Deep Learning for Internet of Things Networks: A Particle Deep Framework. – Future Generation Computer Systems, Vol. **110,** 2020, pp. 91-106.
24. O ř ě s k i, D., D. A n d r ŏ c e c. Genetic Algorithm and Artificial Neural Network for Network Forensic Analytics. – In: Proc. of 43rd International Convention on Information, Communication and Electronic Technology (MIPRO'20), IEEE, 2020, pp. 1200-1205.
25. R i z a l, R., I. R i a d i, Y. P r a y u d i. Network Forensics for Detecting Flooding Attack on Internet of Things (IoT) Device. – Int. J. Cyber-Security Digit Forensics, Vol. **7**, 2018, No 4, pp. 382-390.
26. K u m a r, A., T. J. L i m. Early Detection of Mirai-Like IoT Bots in Large-Scale Networks through Sub-Sampled Packet Traffic Analysis. – In: Proc. of Future of Information and Communication Conference, Springer, 2019, pp. 847-867.
27. M o u s t a f a, N., J. S l a y. The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems. – In: Proc. of 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS'15), IEEE, 2015, pp. 25-31.
28. P a p a m a r t z i v a n o s, D., F. G. M´a r m o l, G. K a m b o u r a k i s. Dendron: Genetic Trees Driven Rule Induction for Network Intrusion Detection Systems. – Future Generation Computer Systems, Vol. **79**, 2018, pp. 558-574.
29. A n t h i, E., L. W i l l i a m s, M. S ł o w i´n s k a, G. T h e o d o r a k o p o u l o s, P. B u r n a p. A Supervised Intrusion Detection System for Smart Home IoT Devices. – IEEE Internet of Things Journal, Vol. **6**, 2019, No 5, pp.9042-9053.
30. U l l a h, I., Q. H. M a h m o u d. A Two-Level Hybrid Model for Anomalous Activity Detection in IoT Networks. – In: Proc. of 16th IEEE Annual Consumer Communications & Networking Conference (CCNC'19), IEEE, 2019, pp. 1-6.
31. A l a z z a m, H., A. A l s m a d y, A. A. S h o r m a n. Supervised Detection of IoT Bot-8 Net Attacks. – In: Proc. of 2nd International Conference on Data Science, e-Learning and Information Systems, ACM, 2019, pp. 1-6.
32. L i u, L., B. X u, X. Z h a n g, X. W u. An Intrusion Detection Method for Internet of Things Based on Suppressed Fuzzy Clustering. – EURASIP Journal on Wireless Communications and Networking, Vol. **2018**, 2018, No 1, pp. 1-7.

33. B a b u, M. J., A. R. R e d d y. SH-IDS: Specification Heuristics Based Intrusion Detection System for IoT Networks. – Wireless Personal Communications, Vol. **112**, 2020, No 3, pp. 2023-2045.
34. P o l l i t t, M. Computer Forensics: An Approach to Evidence in Cyberspace. – In: Proc. of National Information Systems Security Conference, Vol. **2**, 1995, pp. 487-491.
35. K o r o n i o t i s, N., N. M o u s t a f a, E. S i t n i k o v a, J. S l a y. Towards Developing Network Forensic Mechanism for Botnet Activities in the IoT Based on Machine Learning Techniques. – In: Proc. of International Conference on Mobile Networks and Management, Springer, 2017, pp. 30-44.
36. K o r o n i o t i s, N. Designing an Effective Network Forensic Framework for the Investigation of Botnets in the Internet of Things. University of New South Wales, Sydney, Australia, 2020.
37. K o r o n i o t i s, N., N. M o u s t a f a. Enhancing Network Forensics with Particle Swarm and Deep Learning: The Particle Deep Framework. – Future Generation, Vol. **110**, 2020, pp. 91-106.
38. K o r o n i o t i s, N., N. M o u s t a f a, F. S c h i l i r o, P. G a u r a v a r a m, H. J a n i c k e. A Holistic Review of Cybersecurity and Reliability Perspectives in Smart Airports. – IEEE Access, Vol. **8**, 2020, pp. 209802-209834.
39. A l a z z a m, H., A. S h a r i e h, K. E. S a b r i. A Lightweight Intelligent Network Intrusion Detection System Using OCSVM and Pigeon Inspired Optimizer. – Applied Intelligence, Vol. **52**, 2022, No 4, pp. 3527-3544.
40. J a i n, M., S. M a u r y a, A. R a n i, V. S i n g h. Owl Search Algorithm: A Novel Nature-Inspired Heuristic Paradigm for Global Optimization. – Journal of Intelligent & Fuzzy Systems, Vol. **34**, 2018, No 3, pp. 1573-1582.
41. L a i, G., L. L i, Q. Z e n g, N. Y o u s e f i. Developed Owl Search Algorithm for Parameter Estimation of PEMFCs. – International Journal of Ambient Energy, Vol. **2020**, 2020, pp. 1-10.
42. E l-A s h m a w i, W. H., D. S. A b d E l m i n a a m, A. M. N a b i l, E. E l d e s o u k y. A Chaotic Owl Search Algorithm Based Bilateral Negotiation Model. – Ain Shams Engineering Journal, Vol. **11**, 2020, No 4, pp. 1163-1178.
43. M o u l a h i, T., S. Z i d i, A. A l a b d u l a t i f, M. A t i q u z z a m a n. Comparative Performance Evaluation of Intrusion Detection Based on Machine Learning in In-Vehicle Controller Area Network Bus. – IEEE Access, Vol. **9**, 202, pp. 99595-99605.
44. I s t i a q u e, S. M., A. I. K h a n, Z. A l H a s s a n, S. W a h e e d. Performance Evaluation of a Smart Intrusion Detection System (IDS) Model. – European Journal of Engineering and Technology Research, Vol. **6**, 2021, No 2, pp. 148-152.
45. S a l i h, A. A., A. M. A b d u l a z e e z. Evaluation of Classification Algorithms for Intrusion Detection System: A Review. – Journal of Soft Computing and Data Mining, Vol. **2**, 2021, No 1, pp. 31-40.

Appendix. List of abbreviations

Abbreviation	Definition
DDoS	Distributed Denial of Service
DoS	Denial of Service
FPR	False Positive Rate
GNN	Genetic and Neural Network
IDS	Intrusion Detection System
IoT	Internet of Things
LoRa	Long Term Radio Communication
LSTM	Long Short-Term Memory
MQTT	Message Queuing Telemetry Transport
NDF_IoT	Network Digital Forensics for Internet of Things
PCA	Principle Component Analysis
PSD	Particle Swarm and Deep learning
SMOTE	Synthetic Minority Oversampling Technique
TPR	True Positive Rate
UID	Unique Identifier
Zigbee	Zonal Intercommunication Global standard

Received: 01.07.2022; Second Version: 27.07.2022; Accepted: 05.08.2022 (fast track)