Wireless Networks
https://doi.org/10.1007/s11276-023-03435-0

ORIGINAL PAPER

# A novel intrusion detection system for internet of things devices and data

Ajay Kaushik[1] · Hamed Al-Raweshidy[1]

Accepted: 15 June 2023

**Abstract**

As we enter the new age of the Internet of Things (IoT) and wearable gadgets, sensors, and embedded devices are extensively used for data aggregation and its transmission. The extent of the data processed by IoT networks makes it vulnerable to outside attacks. Therefore, it is important to design an intrusion detection system (IDS) that ensures the security, integrity, and confidentiality of IoT networks and their data. State-of-the-art IDSs have poor detection capabilities and incur high communication and device overhead, which is not ideal for IoT applications requiring secured and real-time processing. This research presents a teaching-learning-based optimization enabled intrusion detection system (TLBO-IDS) which effectively protects IoT networks from intrusion attacks and also ensures low overhead at the same time. The proposed TLBO-IDS can detect analysis attacks, fuzzing attacks, shellcode attacks, worms, denial of service (Dos) attacks, exploits, and backdoor intrusion attacks. TLBO-IDS is extensively tested and its performance is compared with state-of-the-art algorithms. In particular, TLBO-IDS outperforms the bat algorithm and genetic algorithm (GA) by 22.2% and 40% respectively.

**Keywords** Data security · Internet of Things · Intrusion detection · Machine learning · Teaching-learning-based optimization

## 1 Introduction

The number of IoT devices has risen dramatically during the last decade [1]. IoT has emerged as a leading technology in digital transformation, connecting tiny devices such as smartphones and gadgets to the internet to enable effective communication between people and things [2]. IoT and wearable devices, like embedded systems, are equipped with a range of sensors and the capacity to connect to a network, allowing them to relay data [3]. Everything from a user's pulse to weather information can be tracked using IoT sensors. IoT plays an important role in domestic as well as corporate sectors. Visual sensors and similar devices are commonly used to monitor and safeguard private buildings, government offices, and healthcare facilities. A large amount of heterogeneous industrial data is sensed, processed, and aggregated using an IoT network.

A majority of digital transformation technologies lack security and privacy measures, jeopardizing the confidentiality, reliability, and integrity of an IoT network [4]. Hackers are increasingly targeting IoT devices and gadgets because of the abundance of useful data they collect. Furthermore, IoT devices and gadgets become more vulnerable to malware attacks due to their always-on network access. The security and privacy of IoT devices are either ignored or considered an afterthought by the manufacturers [4]. Time-to-market and lower retail costs are usually the driving forces behind a device's design and development. Individuals who want to protect themselves adopt software-level solutions like firmware signing and the execution of signed binaries [5]. IoT and wearable devices have different use patterns than conventional embedded systems or personal computers; therefore, these solutions may not be suitable. In addition, because of the emphasis on software-

✉ Hamed Al-Raweshidy
hamed.al-raweshidy@brunel.ac.uk

Ajay Kaushik
ajaykw55055@gmail.com

[1] Department of Electronic and Electrical Engineering, Brunel University London, London, United Kingdom

Published online: 21 August 2023

based security, hardware is usually left vulnerable, opening up new attack routes that may be exploited. Figure 1 shows the possible attacks that different IoT layers can face. It is evident from Fig. 1 that all IoT layers are prone to attacks by an outside intruder. Hence, a security framework is required which can protect IoT networks and their data from outside intrusion attacks.

### 1.1 Major scientific contributions

The profound scientific contributions of this research are the following.

- IoT intrusion detection and security algorithms could increase communication and device overhead, which are key performance parameters for an IoT network. This research presents a teaching-learning-based optimization enabled intrusion detection system (TLBO-IDS) that protects IoT networks and data from intrusion attacks owing to its excellent detection rate, while also ensuring low communication and device overhead, and optimal throughput at the same time.
- The proposed TLBO-IDS can detect a wide range of intrusion attacks on IoT networks and associated data, including analysis, fuzzing, shellcode, worms, DoS, exploits, and backdoor intrusion attacks.
- This research employs the metaheuristic approach TLBO for optimizing the functioning of the proposed IDS [6]. Extensive experiments are conducted to evaluate and compare the efficiency of TLBO-IDS against state-of-the-art approaches, namely the bat algorithm [7] and GA [8].

The rest of the paper is structured as follows. Related work is explained in Sect. 2. IoT design flow practices are explained in Sect. 3. Proposed algorithm is described in Sect. 4. Experimental results and conclusion are discussed in Sects. 5 and 6 respectively.

## 2 Related work

The security of IoT devices has received minimal attention in the literature [9, 10]. According to an early poll [11], IoT security and privacy issues must be solved before IoT devices are widely deployed. Network protocols are used to secure IoT devices, while encrypted communication is regarded as the most efficient way of protecting private data. IoT network security risks and their possible solutions have been summarised by the authors in [12–19], but these threat models are mostly focused on network security. Different IoT topologies were explored by the authors in [5] in an attempt to overcome the problem of IoT security, such as central and distributed designs. Again, network-based solutions emphasize high-level designs without considering whether the resources available on IoT devices

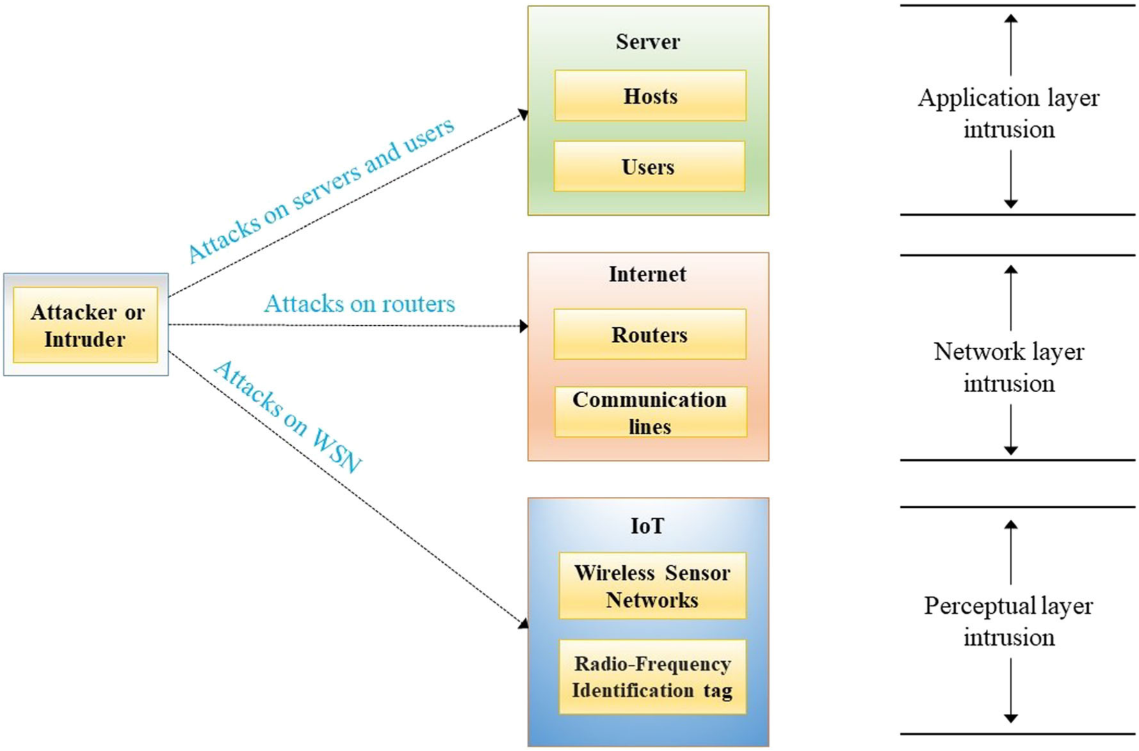


**Fig. 1** Possible attacks on different IoT layers

can support these topologies. Industry researchers have also sought to build secure processor/System-on-Chip (SoC) designs for IoT protection. SoC designers can choose from a large range of components that can perform certain tasks in the security environment thanks to TrustZone technology [20]. Intel proposed the concept of enclaves, where hardware-enforced access control regulations protect data, code, and the stack inside an enclave. Protective design is a priority for Samsung KNOX as well [23]. KNOX contains sensitive data such as business contacts and emails, and it provides a safe execution environment as long as the user is authenticated.

Authors in [24] proposed the concept of artificial immunity for intrusion detection. They implemented cloud, edge, and fog layers in their work. Authors in [25] focused on a lightweight IDS. They incorporated IDS on the application layer, and hence it applied only to small IoT networks. A voting-based IDS was proposed by [26]. For each IoT node, one of the three pre-trained models was selected for intrusion detection. Authors in [27] presented a dual intrusion detection algorithm where the IDS was applied to each node in the same layer of an IoT system to identify and detect malicious attacks. Although it detected each node twice, it lost accuracy. Authors in [28] proposed an IoT anomaly detection approach using decision trees (DT), naïve bayes (NB), and an artificial neural network. Authors in [29] employed neural networks to monitor and detect wireless networks to safeguard small factories and smart homes in the IoT. Authors in [30] proposed the use of a calibration curve to examine and evaluate the performance of various classifier methods such as k-nearest neighbour (KNN), NB, and support vector machine (SVM) to detect BotNet attacks. Authors in [31] employed a DT classifier to train an IDS using a layered approach. Authors in [32] employed fuzzy rule interpolation to present a detection model for an IoT-BotNet attack. The developed approach was applied to open-source IoT-Bot data. Authors in [33] presented a machine learning-based approach for the detection of distributed denial of service in cloud computing. They implemented KNN, random forest (RF), and NB in their approach. Authors in [7] employed machine learning for intrusion detection in industrial IoT. They used bat algorithm, particle swarm optimization (PSO), and machine learning approach RF for effective intrusion detection. After comparisons, their research concluded that bat algorithm has a better performance than PSO. Authors in [8] employed multiple classification models such as RF, NB, DT, and extra-trees for intrusion detection in industrial IoT. Their research used GA to enhance the feature selection in their research. Authors in [34] employed a combination of deep learning approaches to present an IDS. They used convolutional neural network for feature extraction and long short-term memory for classification. Authors in [35] proposed a cloud-based approach for ensuring data integrity and concluded that RF performs better than other machine learning approaches such as NB, SVM, and KNN. Authors in [36] proposed an IDS based on pattern recognition. They used two different layers for classifying network connections according to the user service and type of attack. Authors in [37, 38] presented a quantum secure cryptography framework that protects IoT devices and data from quantum attacks. They presented algorithms for public as well as private key encryption. Authors in [39] presented a cloud-based IDS to enhance the security of cloud data. They implemented feature engineering and RF in their approach. Authors in [40] proposed an enhanced IDS by customizing four state-of-the-art deep learning approaches. They trained the model on the cloud server before using it for classification.

Many hardware, as well as software-based IoT security solutions, are proposed in the literature. The passive nature of hardware-based security systems means that they cannot detect or mitigate threats. Furthermore, IoT security and intrusion detection algorithms can lead to performance issues such as increased overhead and low throughput. A majority of the software-based state-of-the-art IoT security solutions do not focus on the above-mentioned performance issues. The proposed TLBO-IDS safeguards IoT systems and data against analysis, fuzzing, shellcode, worms, DoS, exploits, and backdoor intrusion attacks, ensuring a low overhead at the same time.

## 3 IoT design flow practices

This section discusses IoT design flow practices. Open and closed software, cryptographic systems, debug interfaces, and hardware components of an IoT system are examined for security vulnerabilities.

### 3.1 Open versus closed source software

It is difficult to determine whether open-source or closed-source software is better for security. An attacker just hunts for bugs in the source code to discover a way to attack the device using open-source software. Linux-based stacks are widely used at the device firmware level, while some devices use FreeRTOS [41] or other open-source software to build Linux atop. Many other companies, such as Wind River and Blackberry, use their own proprietary software, such as VxWorks [42] and QNX [43]. In open-source software, manufacturers don't have to depend on system vendors for patching bugs, allowing for a quicker reaction time to the problem. A quick reaction to a bug also makes it difficult for an attacker to reverse engineer user interfaces in search of software faults. In closed-loop software, the manufacturers must depend on the suppliers after vulnerabilities are discovered. It is thus necessary to consider design needs, the availability of support, and the level of security provided by the stack.

### 3.2 Weak or bad cryptographic implementations

The security software must be able to verify the integrity and authenticity of the downloaded data to remotely update a device. Cryptographic algorithms are used for this purpose. Many vulnerabilities have been identified in security software and cryptography algorithms because of the complexity of the mathematics involved and implementation mistakes [44]. These flaws demonstrate how an inadequately built cryptographic system may be circumvented by an attacker, allowing a remote attack on the device. Remote exploitation of a device is possible through these vulnerabilities, where an attacker takes advantage of secure sockets layer (SSL) flaws to enable the installation of malicious firmware over the network via a fake distribution server.

### 3.3 Debug interfaces on production runs

A device must be functionally tested before going into mass production. A circuit board must include programming and testing points for all the components it contains. Even though these interfaces are typically unlabelled, they are not deleted after testing. Using these interfaces, an attacker may modify a unit's functionality or infect it with malicious code. In software components, compilers may output binaries with debugging symbols, which represent the source that generated a given block of machine code. This allows an attacker to recreate the source code and exploit vulnerabilities.

### 3.4 Hardware threats

IoT security might be jeopardized by hardware Trojans. If an integrated circuit is tampered with, it may leak sensitive information to an attacker, cause a device to function outside of set limits, or make the device completely futile. Hardware trojans are harder to detect; therefore, standard testing methods are ineffective in catching them, necessitating the use of more costly specialist procedures. Furthermore, a hostile attacker might use the cryptographic core of an IoT device's SoC to install a hardware Trojan. Random number generators can be affected by this Trojan when it is activated. As a result, an attacker would need a far smaller amount of processing power to decode the data.

## 4 Proposed algorithm

The proposed TLBO-IDS comprises data collection, data pre-processing, feature extraction using the metaheuristic approach TLBO, and model training and testing. This section explains the stepwise implementation of TLBO-IDS.

### 4.1 TLBO

The TLBO algorithm is based on the teaching–learning process. It is derived from the classroom practice of teaching and learning, where students first learn from a teacher and subsequently from each other [6]. TLBO is a population-based approach where a group of students form the population. As a result, a student in the class is a feasible solution. The subjects taught in the class are referred to as the design variables of the optimization problem, and the student's result is considered as the fitness function of the feasible solution. TBLO is divided into two stages, which are explained below.

#### 4.1.1 Teacher phase

The teacher phase involves students learning from the teacher. Generally, the most learned and knowledgeable person in the society is considered as a teacher. The teacher is responsible for educating the students and ensuring that they get good marks. On the other hand, students acquire knowledge and obtain marks based on the quality of teaching. Let us assume that there are $n$ subjects (design variables $j = 1, 2, 3. \ldots.. .n$) allotted to $N_p$ students. During a teaching–learning process (iteration $k = 1, 2, 3, 4. \ldots.n$), $M_j^k$ is the mean student result in a subject '$j$'. As we know that teacher is the most educated person in the society. To simulate this, we consider the best student (feasible solution) in the population as a teacher. Let $X_T^k$ is the most feasible solution at $k_{th}$ learning iteration and $X_{Tj}^k$ is the $j_{th}$ design variable in the best feasible solution. The difference between the teacher's result and the student's mean result in $j_{th}$ subject is given by Eq. (1).

$$D_j^k = r\left(X_{Tj}^k - T_F M_j^k\right) \tag{1}$$

where $T_F$ denotes the teaching factor that determines the mean value to be changed. $r$ is a random number that has a value between [0,1].

In this process, the feasible solutions (students) are iteratively improved by moving them toward the best feasible solution (teacher), while considering the current mean value of the feasible solutions. The $i_{th}$ feasible solution at $k_{th}$ learning cycle is updated as per Eq. (2).

$$X^k_{new,i,j} = X^k_{old,i,j} + D^k_j \tag{2}$$

$X^k_{new,i}$ is accepted only if it is better than $X^k_{old,i}$, otherwise, it is rejected. The accepted feasible solutions act as input to the student phase.

### 4.1.2 Student phase

Students gain expertise and knowledge by randomly interacting with each other. A student ($a$) learns from another student ($b$) in the class if student ($b$) has more knowledge than student ($a$). Hence, student ($a$) is moved towards student ($b$) if student ($b$) is better than the student ($a$). Otherwise, the student ($a$) would be moved away from the student ($b$). Two feasible solutions, $X^k_a$, $X^k_b$ are selected randomly from the class, where $a$ and $b$ are random integers in the range [1, $N_p$] and $a$ is not equal to $b$.

$$IfF(X^k_a) > F(X^k_b)$$

$$X^k_{new\_SP,a,j} = X^k_{a,j} + r\left(X^k_{a,j} - X^k_{b,j}\right) \tag{3}$$

$$Else$$

$$X^k_{new\_SP,a,j} = X^k_{a,j} + r\left(X^k_{b,j} - X^k_{a,j}\right) \tag{4}$$

$$Endif$$

where $F(X)$ is the fitness function of the feasible solution, $X^k_{new\_SP,a,j}$ denotes the $j_{th}$ design variable of the modified feasible solution in the student phase at $k_{th}$ teaching–learning cycle.

Afterward, the fitness value of $X^k_{new\_SP,a}$ is calculated.

$$IfF\left(X^k_{new_{SP},a}\right) > F(X^k_{new.a})$$

$$X^k_{new.a} = X^k_{new\_SP,a} \tag{5}$$

$$Else$$

$$X^k_{new.a} = X^k_{new.a} \tag{6}$$

$$Endif$$

## 4.2 Data collection and pre-processing

This research uses the UNSW-NB15 dataset. A total of 42 labelled features are generated, which are then divided into six categories, namely flow features, time features, content features, connection features, general purpose features, and labelled features. Apart from the normal data, this research considers seven types of attacks, namely analysis, fuzzing, shellcode, worms, DoS, exploits, and backdoor intrusion attacks. This research uses 174,160 records for the training set and 74,640 records for the testing set from the UNSW-NB15 dataset. Data pre-processing can minimize the size of raw data and speed up the model training process. Data quality is primarily determined by correctness, integrity, and consistency. But in the real world, databases, and data warehouses are filled with erroneous, incomplete, and inconsistent data. Following the data collection, the proposed research pre-processes the raw data to turn it into a structured form. As a part of data pre-processing, data is partitioned into training and testing sets. The proposed research uses 70% of the data for training and 30% of the data for testing. It is where the duplication and overlapping issues start appearing in the data. Data duplication corresponds to a situation when a data sequence occurs multiple times in a set. On the other hand, data overlapping is a condition when a data sequence appears in both sets. Data duplication and overlapping can result in an unreliable evaluation model. If the data pool comprises overlapped sequences, the same sequence could exist in both the training and testing set, compromising the model's overall performance. To mitigate this issue, the proposed model uses data cleaning and ensures that no duplicate or overlapping data sequence exists. Clean training and clean testing data sets are stored separately from the original uncleaned data.

## 4.3 Feature extraction and TLBO optimization

Feature extraction is the process of extracting important features from the dataset. This is an important step because it increases the calculation speed, preserves storage space, and avoids redundant features of the data. The process of feature selection involves the selection of an appropriate 0/1 string, where 1 represents the selection of a particular feature and 0 shows that the feature is not selected. The string length is the same as the number of features present in the dataset. The pre-processed data from Sect. 4.2 is input into TLBO for feature optimization. The TLBO optimization works as per Sect. 4.1.

## 4.4 Training and testing

The main purpose of an IDS is to place incoming traffic into normal and intrusive categories. TLBO-IDS is trained distinguish between normal and abnormal data. Figure 2 not only presents the architecture of TLBO-IDS but also depicts the training and testing processes involved. As shown in Fig. 2, the UNSW-NB15 dataset is pre-processed and feature extraction is done using TLBO. The pre-processing step includes splitting of data into training and test sets, and data cleaning. The model is then trained using the training data set and machine learning classification algorithm RF. The accuracy of the trained model is tested for intrusion detection using a test data set. Normal outcome indicates no intrusion whereas abnormal output shows that

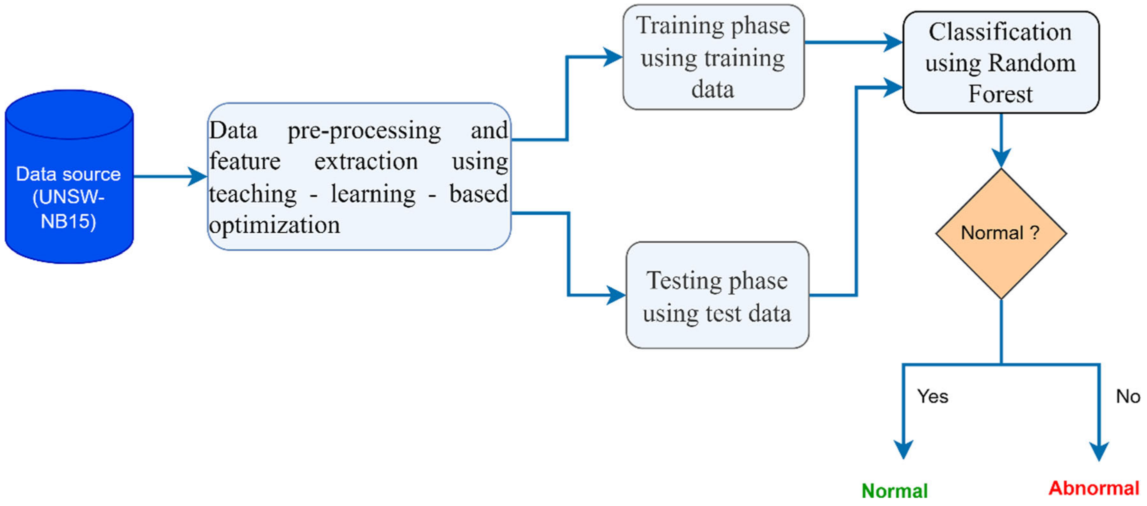

Data source (UNSW-NB15)
Data pre-processing and feature extraction using teaching - learning - based optimization
Training phase using training data
Testing phase using test data
Classification using Random Forest
Normal ?
Yes
No
Normal
Abnormal

**Fig. 2** TLBO-IDS architecture

there is an intrusion detected. TLBO-IDS uses RF for training purposes. RF is a bagging classifier that involves the use of numerous DTs. Each DT is fed with input by row and column sampling. One of the major drawbacks of DT is its low bias and big variance. This means that the tree will perform better in the training phase but it will have a poor performance in the testing phase. The proposed approach applies a voting strategy using RF which lowers the variance from high to low because the decision in the voting strategy is based on numerous trees rather than a single tree [7].

## 5 Experimental results

This section evaluates and compares the performance of the proposed TLBO-IDS with the state-of-the-art approaches, namely the bat algorithm [7] and GA [8]. The IDS performance parameters are formulated in Sect. 5.1 and the output with analysis is presented in Sect. 5.2.

### 5.1 Evaluation metrics

For intrusion detection, the prediction of data comprises four cases, which are discussed below.

True positive (TP). When both the actual and predicted labels are positive.

False negative (FN). When the actual label is positive and the predicted label is negative.

True negative (TN). When both actual and predicted labels are negative.

False positive (FP). When the actual label is negative and the predicted label is positive.

The parameters for testing the effectiveness of an IDS are calculated in Eqs. 7 and 8.

1. Detection rate. It is the ability of an IDS to effectively detect an intrusion.

$$Detectionrate = \frac{TP}{TP + FN} \tag{7}$$

2. Accuracy. It is measured as the proportion of correct prediction results out of the total number of samples.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{8}$$

### 5.2 Output and analysis

The experimental environment configuration for obtaining the output in TLBO-IDS is shown in Table 1. Figure 3 shows the detection rate of TLBO-IDS on normal and seven types of attack data, compared with the bat algorithm [7] and GA [8]. The detection rate is calculated as per Eq. (7). As seen in Fig. 3, TLBO-IDS has an excellent detection rate for the detection of analysis, fuzzing, shellcode, worms, DoS, exploits, and backdoor intrusion attacks. In particular, TLBO-IDS outperforms the bat

**Table 1** Experimental environment configuration of TLBO-IDS

| Name | Value |
|---|---|
| Training set | 170,254 records |
| Testing set | 78,547 records |
| No. of labelled features | 42 |
| TLBO iterations | 1000 |
| RAM during implementation | 8 gigabytes |
| Disk capacity | 1 terabyte |
| Processor | i5 |

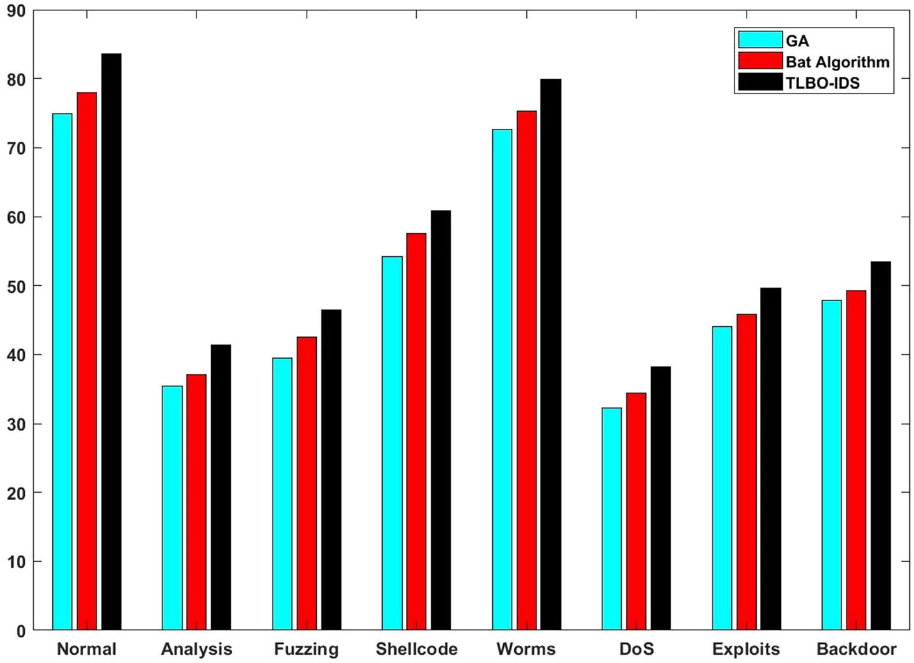


**Fig. 3** Comparison of TLBO-IDS with state-of-the-art approaches for detection rate

algorithm [7] and GA [8] by 7.1% and 11.4% respectively, in terms of detection rate.

Apart from the detection rate, accuracy is an important metric for measuring the effectiveness of an IDS. Therefore, the performance of the proposed research is evaluated and compared with state-of-the-art research for accuracy. Figure 4 shows that the proposed TLBO-IDS approach performs better than the bat algorithm [7] and GA [8] in terms of accuracy. TLBO-IDS shows the best accuracy compared to the existing approaches, while GA has the worst output.

System throughput is a key parameter for IoT performance and is calculated as packets per second. In this research, we compare the performance of the proposed TLBO-IDS with state-of-the-art research for system throughput. Figure 5 shows the throughput performance of TLBO-IDS in comparison with the bat algorithm [7] and GA [8]. The proposed approach has a better throughput

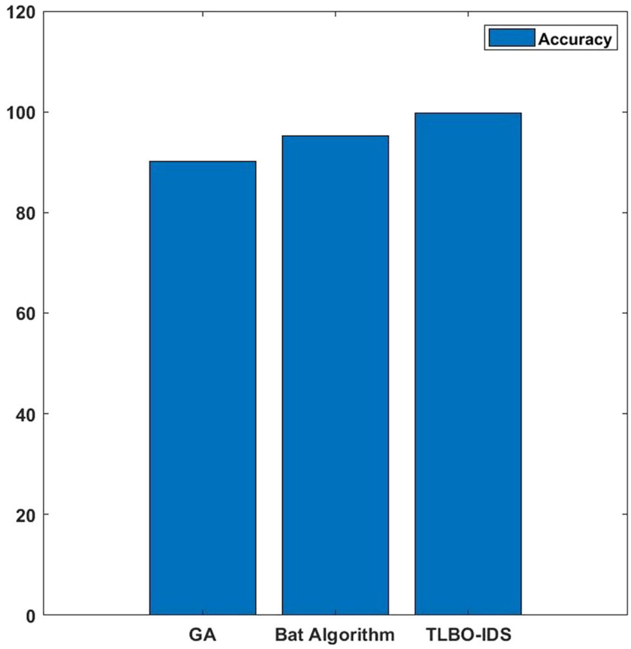


**Fig. 4** TLBO-IDS compared with state-of-the-art approaches for accuracy

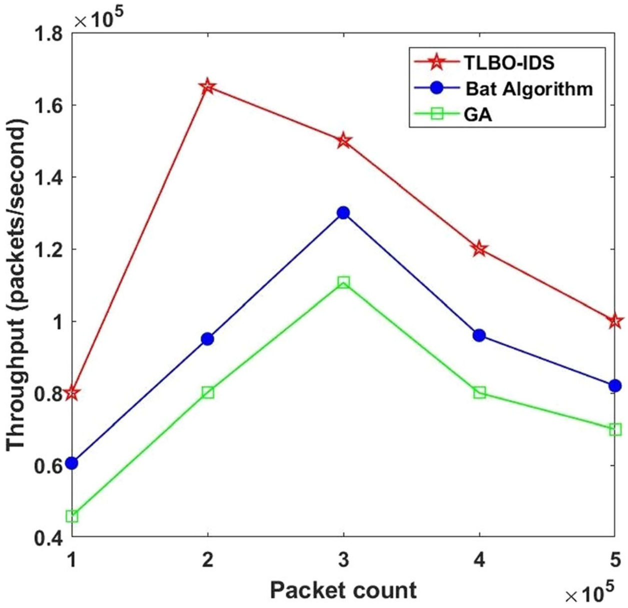


**Fig. 5** Comparison of TLBO-IDS with state-of-the-art approaches for throughput (packets/second)

than the state-of-the-art approaches. In particular, TLBO-IDS performs better than the bat algorithm [7] and GA [8] by 32.1% and 74.7% respectively in terms of system throughput.

The implementation of an IDS must not increase device or communication overhead. The proposed TLBO-IDS framework employs the metaheuristic approach TLBO, which optimizes the operational parameters and performance of the IDS, minimizing the overhead caused. Figure 6 compares the TLBO-IDS approach with the bat algorithm [7] and GA [8] for device overhead. As seen in Fig. 6, TLBO-IDS has a low overhead in comparison to state-of-the-art research. In particular, the TLBO-IDS approach has 16.6% and 52.7% less overhead than the bat algorithm [7] and GA [8] respectively.

Communication overhead is generally caused due to excessive number of transmission packets in an IoT network. Communication overhead impacts the overall network performance. This research compares the performance of the proposed approach with the state-of-the-art algorithms in terms of communication overhead. Figure 7 shows a comparison of TLBO-IDS with the bat algorithm [7] and GA [8] for communication overhead. It could be seen from Fig. 7 that TLBO-IDS has low communication overhead as compared to the state-of-the-art approaches. In particular, TLBO-IDS performs better than the bat algorithm [7] and GA [8] by 22.2% and 40% respectively in terms of communication overhead.

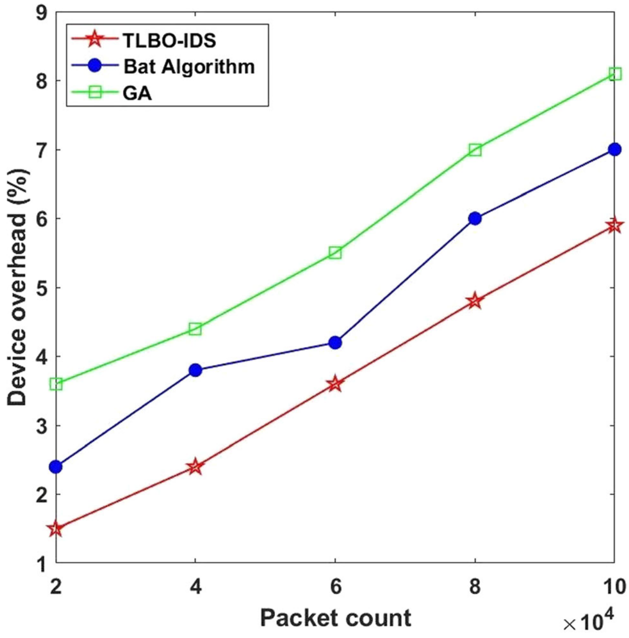


**Fig. 6** Comparison of TLBO-IDS with state-of-the-art approaches for device overhead

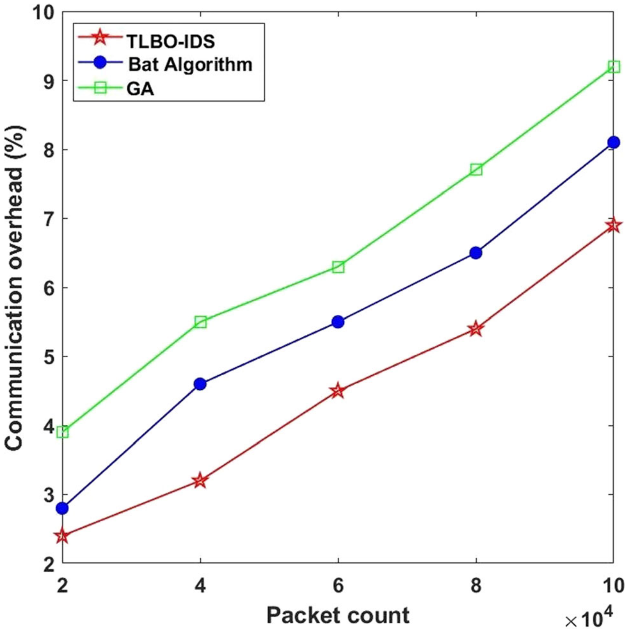


**Fig. 7** Comparison of TLBO-IDS with state-of-the-art approaches for communication overhead

## 6 Conclusion

This research proposed a teaching–learning-based optimization enabled intrusion detection system to detect intrusion attacks on IoT networks, ensuring low overhead at the same time. Data from the UNSW-NB15 dataset was pre-processed, and features were extracted using TLBO. The machine learning classification algorithm RF was used to train the model. The proposed framework could detect analysis, fuzzing, shellcode, worms, DoS, exploits, and backdoor intrusion attacks. Extensive experiments were carried out, and the performance of TLBO-IDS was tested in terms of detection rate, accuracy, throughput, device overhead, and communication overhead. TLBO-IDS was compared with state-of-the-art algorithms, namely the bat algorithm and GA. The proposed approach had excellent accuracy and detection rate. In particular, TLBO-IDS outperformed the bat algorithm and GA by 22.2% and 40%, respectively. TLBO-IDS can distinguish between normal and intrusive network traffic and has applications in both domestic and industrial sectors where the security of IoT data is paramount. Future works include enhancing the applicability and utility of the proposed approach by incorporating various encryption standards.

**Author contributions** Dr. AK prepared the algorithm framework, implemented the algorithm, obtained the results, and documented the research paper. Prof. HA-R helped in revising the paper by contributing to the output evaluation and documentation of the research paper.

**Funding** The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

**Data availability** Data is available on reasonable request.

## Declarations

**Conflict of interest** The authors have no relevant financial or non-financial interests to disclose.

## References

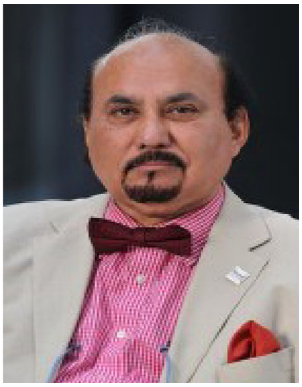
1. Mahamat, M., Jaber, G., & Bouabdallah, A. (2023). Achieving efficient energy-aware security in IoT networks: A survey of recent solutions and research challenges. *Wireless Networks, 29*(2), 787–808.
2. Janabi, S. M. A., & Kurnaz, S. (2023). A new localization mechanism in IoT using grasshopper optimization algorithm and DVHOP algorithm. *Wireless Networks*. https://doi.org/10.1007/s11276-023-03247-2
3. Kaushik, A., Goswami, M., Manuja, M., Indu, S., & Gupta, D. (2020). A binary PSO approach for improving the performance of wireless sensor networks. *Wireless Personal Communications, 113*, 263–297.
4. Cvitić, I., Peraković, D., Periša, M., & Botica, M. (2021). Novel approach for detection of IoT generated DDoS traffic. *Wireless Networks, 27*(3), 1573–1586.
5. Bodkhe, U., & Tanwar, S. (2021). Secure data dissemination techniques for IoT applications: Research challenges and opportunities. *Software: Practice and Experience, 51*(12), 2469–2491.
6. Gill, H. S., Khehra, B. S., Singh, A., & Kaur, L. (2019). Teaching-learning-based optimization algorithm to minimize cross entropy for Selecting multilevel threshold values. *Egyptian Informatics Journal, 20*(1), 11–25.
7. Gaber, T., Awotunde, J. B., Folorunso, S. O., Ajagbe, S. A., & Eldesouky, E. (2023). Industrial internet of things intrusion detection method using machine learning and optimization techniques. *Wireless Communications and Mobile Computing*. https://doi.org/10.1155/2023/3939895
8. Kasongo, S. M. (2021). An advanced intrusion detection system for IIoT based on GA and tree based algorithms. *IEEE Access, 9*, 113199–113212.
9. Ziegeldorf, J. H., Morchon, O. G., & Wehrle, K. (2014). Privacy in the Internet of Things: Threats and challenges. *Security and Communication Networks, 7*(12), 2728–2742.
10. Thierer, A. D. (2015). The internet of things and wearable technology: Addressing privacy and security concerns without derailing innovation. In *Adam Thierer, The Internet of Things and Wearable Technology: Addressing Privacy and Security Concerns without Derailing Innovation, 21*.
11. Atziori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey computer networks. *Computer Networks, 54*(28), 2787–2805.
12. Babar, S., Mahalle, P., Stango, A., Prasad, N., & Prasad, R. (2010). Proposed security model and threat taxonomy for the Internet of Things (IoT). In *Recent Trends in Network Security and Applications: Third International Conference, CNSA 2010, Chennai, India, July 23-25, 2010. Proceedings 3* (pp. 420-429). Springer.
13. Kocher, P., Jaffe, J., & Jun, B. (1999). Differential power analysis. In *Advances in Cryptology—CRYPTO'99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings 19* (pp. 388-397). Springer.
14. Mulligan, G. (2007). The 6LoWPAN architecture. In *Proceedings of the 4th workshop on Embedded networked sensors* (pp. 78–82).
15. Hummen, R., Ziegeldorf, J. H., Shafagh, H., Raza, S., & Wehrle, K. (2013). Towards viable certificate-based authentication for the internet of things. In *Proceedings of the 2nd ACM workshop on Hot topics on wireless network security and privacy* (pp. 37–42).
16. Rescorla, E., & Modadugu, N. (2006). *Datagram transport layer security* (No. rfc4347).
17. Kent, S., & Seo, K. (2005). *Security architecture for the internet protocol* (No. rfc4301).
18. Brachmann, M., Keoh, S. L., Morchon, O. G., & Kumar, S. S. (2012). End-to-end transport security in the IP-based internet of things. In *2012 21st International conference on computer communications and networks (ICCCN)* (pp. 1–5). IEEE.
19. Seggelmann, R. (2013). *SCTP: Strategies to secure end-to-end communication (Doctoral dissertation* (p. 2012). Universität Duisburg-Essen.
20. Kim, H. J. (2012). Online social media networking and assessing its security risks. *International journal of security and its applications, 6*(3), 11–18.
21. McKeen, F., Alexandrovich, I., Berenzon, A., Rozas, C. V., Shafi, H., Shanbhogue, V., & Savagaonkar, U. R. (2013). Innovative instructions and software model for isolated execution. *Hasp@isca, 10*(1).
22. Anati, I., Gueron, S., Johnson, S., & Scarlata, V. (2013). Innovative technology for CPU based attestation and sealing. In *Proceedings of the 2nd international workshop on hardware and architectural support for security and privacy* (Vol. 13, No. 7).
23. Krishnan, M. (2015). Survey on security risks in Android OS and an introduction to Samsung KNOX. *International Journal of Computer Science and Information Technologies, 6*(4), 3965–3967.
24. Hosseinpour, F., Vahdani Amoli, P., Plosila, J., Hämäläinen, T., & Tenhunen, H. (2016). An intrusion detection system for fog computing and IoT based logistic systems using a smart data approach. *International Journal of Digital Content Technology and its Applications*, *10*(5).
25. Nobakht, M., Sivaraman, V., & Boreli, R. (2016). A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow. In *2016 11th International conference on availability, reliability and security (ARES)* (pp. 147–156). IEEE.
26. Alotaibi, B., & Elleithy, K. (2016). A majority voting technique for wireless intrusion detection systems. In *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)* (pp. 1–6). IEEE.
27. Pajouh, H. H., Javidan, R., Khayami, R., Dehghantanha, A., & Choo, K. K. R. (2016). A two-layer dimension reduction and two-

tier classification model for anomaly-based intrusion detection in IoT backbone networks. *IEEE Transactions on Emerging Topics in Computing, 7*(2), 314–323.

28. Moustafa, N., Turnbull, B., & Choo, K. K. R. (2018). An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. *IEEE Internet of Things Journal, 6*(3), 4815–4830.
29. Roux, J., Alata, E., Auriol, G., Nicomette, V., & Kaâniche, M. (2017). Toward an intrusion detection approach for IoT based on radio communications profiling. In *2017 13th European dependable computing conference (EDCC)* (pp. 147–150). IEEE.
30. Kanimozhi, V., & Jacob, T. P. (2019). Calibration of various optimized machine learning classifiers in network intrusion detection system on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. *International Journal of Engineering Applied Sciences and Technology, 4*(6), 2455–2143.
31. Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications, 50*, 102419.
32. Al-Kasassbeh, M., Almseidin, M., Alrfou, K., & Kovacs, S. (2020). Detection of IoT-botnet attacks using fuzzy rule interpolation. *Journal of Intelligent & Fuzzy Systems, 39*(1), 421–431.
33. Mishra, A., Gupta, B. B., Peraković, D., Peñalvo, F. J. G., & Hsu, C. H. (2021). Classification based machine learning for detection of ddos attack in cloud computing. In *2021 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1–4). IEEE.
34. Sahu, A. K., Sharma, S., Tanveer, M., & Raja, R. (2021). Internet of things attack detection using hybrid deep learning model. *Computer Communications, 176*, 146–154.
35. Ahmad, F. B., Nawaz, A., Ali, T., Kiani, A. A. & Mustafa, G. (2022) Securing cloud data: A machine learning based data categorization approach for cloud computing, https://doi.org/10.21203/rs.3.rs-1315357/v1.
36. Abdeldayem, M. M. (2022). Intrusion detection system based on pattern recognition. *Arabian Journal for Science and Engineering*. https://doi.org/10.1007/s13369-022-07421-0
37. Kaushik, A., Vadlamani, L. S. S., Hussain, M. M., Sahay, M., Singh, R., Singh, A. K., & Kousik, N. G. V. (2023). Post quantum public and private key cryptography optimized for IoT security. *Wireless Personal Communications, 129*(2), 893–909.
38. Singh, R., Hussain, M. M., Sahay, M., Indu, S., Kaushik, A., & Kumar Singh, A. (2021). Loki: A lightweight LWE method with rogue bits for quantum security in IoT devices. In *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2020, Volume 2* (pp. 543–553). Springer Singapore.
39. Attou, H., Guezzaz, A., Benkirane, S., Azrour, M., & Farhaoui, Y. (2023). Cloud-based intrusion detection approach using machine learning techniques. *Big Data Mining and Analytics, 6*(3), 311–320.
40. Elnakib, O., Shaaban, E., Mahmoud, M., & Emara, K. (2023). EIDM: Deep learning model for IoT intrusion detection systems. *The Journal of Supercomputing, 79*, 13241–13261.
41. FreeRTOS reference manual: API functions and configuration options. *Real Time Engineers Limited*, 2009.
42. Barbalace, A., Luchetta, A., Manduchi, G., Moro, M., Soppelsa, A., & Taliercio, C. (2008). Performance comparison of VxWorks, Linux, RTAI, and Xenomai in a hard real-time application. *IEEE Transactions on Nuclear Science, 55*(1), 435–439.
43. Qnx operating systems. (1982–2014), *Available online.* https://blackberry.qnx.com/en/products/foundation-software/qnx-rtos
44. Common Vulnerabilities and Exposures CVE-2014–0160, *Available online.* https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Ajay Kaushik** is a postdoctoral visiting researcher (Computer Science), Department of Electronic and Electrical Engineering, Brunel University London, United Kingdom. He obtained a Ph.D. from the Department of Computer Science and Engineering, Delhi Technological University, Delhi, India. His areas of interest are the internet of things, edge computing, 5G, 6G, wireless sensor networks, machine learning, neural networks, and nature-inspired intelligence. He has published many journal and conference research papers in field of wireless sensor networks. He received the Research Excellence Award for outstanding research during his Ph.D. He completed a Master of Technology following a B.Tech. at Kurukshetra University, India. He passed the International English Language Testing System with a Band 7 score in 2021. He is a fellow of Advanced Higher Education, United Kingdom.



**Hamed Al-Raweshidy** is a Professor of Communications Engineering who received his B.Eng. and M.Sc. degrees in 1977 and 1980, respectively, from the University of Technology in Baghdad. In 1987, he received his Postgraduate Diploma from Glasgow University in Glasgow, Scotland. Strathclyde University in Glasgow, Scotland, granted him a Ph.D. in 1991. He has worked for the Space and Astronomy Research Centre in Iraq, United States, Germany and Kent University. Professor Al-Raweshidy is the Director of the Wireless Networks and Communications Centre as well as the Director of Postgraduate Studies (EEE) at Brunel University in London, UK.