International Journal of Information Security
https://doi.org/10.1007/s10207-023-00803-x

REGULAR CONTRIBUTION

A voting gray wolf optimizer-based ensemble learning models for intrusion detection in the Internet of Things

Yakub Kayode Saheed[1] · Sanjay Misra[2]

Accepted: 11 December 2023

Abstract
The Internet of Things (IoT) has garnered considerable attention from academic and industrial circles as a pivotal technology in recent years. The escalation of security risks is observed to be associated with the growing interest in IoT applications. Intrusion detection systems (IDS) have been devised as viable instruments for identifying and averting malicious actions in this context. Several techniques described in academic papers are thought to be very accurate, but they cannot be used in the real world because the datasets used to build and test the models do not accurately reflect and simulate the IoT network. Existing methods, on the other hand, deal with these issues, but they are not good enough for commercial use because of their lack of precision, low detection rate, receiver operating characteristic (ROC), and false acceptance rate (FAR). The effectiveness of these solutions is predominantly dependent on individual learners and is consequently influenced by the inherent limitations of each learning algorithm. This study introduces a new approach for detecting intrusion attacks in an IoT network, which involves the use of an ensemble learning technique based on gray wolf optimizer (GWO). The novelty of this study lies in the proposed voting gray wolf optimizer (GWO) ensemble model, which incorporates two crucial components: a traffic analyzer and a classification phase engine. The model employs a voting technique to combine the probability averages of the base learners. Secondly, the combination of feature selection and feature extraction techniques is to reduce dimensionality. Thirdly, the utilization of GWO is employed to optimize the parameters of ensemble models. Similarly, the approach employs the most authentic intrusion detection datasets that are accessible and amalgamates multiple learners to generate ensemble learners. The hybridization of information gain (IG) and principal component analysis (PCA) was employed to reduce dimensionality. The study utilized a novel GWO ensemble learning approach that incorporated a decision tree, random forest, K-nearest neighbor, and multilayer perceptron for classification. To evaluate the efficacy of the proposed model, two authentic datasets, namely, BoT-IoT and UNSW-NB15, were scrutinized. The GWO-optimized ensemble model demonstrates superior accuracy when compared to other machine learning-based and deep learning models. Specifically, the model achieves an accuracy rate of 99.98%, a DR of 99.97%, a precision rate of 99.94%, an ROC rate of 99.99%, and an FAR rate of 1.30 on the BoT-IoT dataset. According to the experimental results, the proposed ensemble model optimized by GWO achieved an accuracy of 100%, a DR of 99.9%, a precision of 99.59%, an ROC of 99.40%, and an FAR of 1.5 when tested on the UNSW-NB15 dataset.

Keywords Internet of Things · Gray wolf optimizer · Ensemble model · Intrusion detection system · K-nearest neighbor · Multilayer perceptron · Random forest · Decision tree · Average of probability

1 Introduction

The Internet of Things (IoT) is experiencing rapid growth and assuming an increasingly significant role in our everyday existence. IoT nodes can establish a connection with the Internet using an Internet Protocol (IP) address [1]. The past decade has witnessed a significant surge in the level of interconnectivity among individuals, machines, and services,

✉ Sanjay Misra
sanjay.misra@ife.no

Yakub Kayode Saheed
yakubu.saheed@aun.edu.ng

[1] Department of Computer Science, University of Alcala, Madrid, Spain

[2] Department of Applied Data Science, Institute for Energy Technology, Halden, Norway

Published online: 09 January 2024

ultimately leading to the emergence of a novel communication paradigm referred to as the IoT [2]. The proliferation of self-configured smart nodes is fueling the development of a wide range of innovative applications, including but not limited to home automation, process automation, smart automobiles, health-care systems, decision analytics, smart grids, industrial development, and autonomous cars [3]. It is predicted by analysts that in the future, the number of interconnected devices will surpass that of the human population on Earth. As per the International Data Corporation's projections, by the year 2025, a total of 41.6 billion interconnected IoT devices are expected to generate a staggering amount of 79.4 zettabytes of data, in contrast with the anticipated global population of 8.1 billion individuals [4].

The IoT is vulnerable to a range of security threats and presents significant security challenges for end-users, particularly as it continues to expand into various aspects of communal life, as shown in Fig. 1. The IoT is a complex system of various networks that include security measures for sensor data, Internet and mobile network connectivity, privacy protection, network authentication, access control, and information management, as noted in the source [5]. In recent years, the occurrence of anomalies and security breaches on IoT devices has become increasingly prevalent. The Internet of Things infrastructure framework is becoming increasingly complex, which is resulting in the introduction of undesired vulnerabilities into its systems. The IoT has the potential to facilitate the seamless integration of physical objects into networks, thereby providing advanced information services to individuals. A multitude of IoT services and applications that utilize ML have emerged across various domains such as security, surveillance, health care, transportation, control, and object monitoring. Preventative security measures are often limited by inadequate planning and implementation, and given the inevitability of attacks, machine learning systems can offer essential services and resilient security strategies for safeguarding IoT devices [6].

The attack detection system is classified as either a signature-based or an anomaly-based system. Signature-based system attacks compare certain patterns, such as bytes or harmful instruction sequences, in malware-infected network traffic to known attack types stored in a database [7]. Systems based on anomalies detect unknown threats or deviations from the typical flow. Unlike signature-based detection systems, machine learning-based solutions have the potential to detect unknown attacks. However, the ML models must be sufficiently precise to maximize high accuracy, increase the detection rate (DR), have a high ROC, and minimize false alarms [6]. They must be trained and assessed on genuine datasets to demonstrate their efficacy in real-world deployments. The basic strategy is to utilize ML to create a model of legitimate action and then analyze new behavioral attacks against the ML model.

As a result, numerous approaches and strategies such as data encryption, firewalls, and user verification via the fog computing model have been created and implemented to defend the IoT platform. These attack channels and risks continue to evolve, rendering traditional security solutions inefficient and ineffective at addressing the IoT safety challenge, paving the way for a new wave of IDS based on ML. A substantial amount of work and study has been undertaken to determine the optimum intelligent IDS for various types of applications in IoT-based environments [8]. As IDS is one of the key remedies used to ensure IoT security, there is a propensity to employ multiple techniques concurrently [9]. Alharbi et al. [10] proposed an IoT security proof-of-concept system built into the fog computing layer. Each unit defends against a specific type of attack. The IDS of traffic analyzer components was employed to spot DDoS and DoS attacks with a classification engine based on the decision tree ML technique. To authenticate the IDS's answer, the challenge-response component sends a challenge communication in the event of intrusion detection. As a result of the system's failure to respond to this message, the firewall unit disables the connection. Pajouh et al. [11] introduced a unique layered IDS for IoT mainstay networks that use a two-tier (2-tier) dimensionality reduction and classification phase. The dimension reduction engine is built of component analysis and LDA units, while the classification engine is composed of NB and a cascaded version of the CF-KNN units. The NB was utilized to classify attack records, which were further improved using the CF-KNN algorithm as a secondary filter layer. Using the NSLKDD [12] dataset, the suggested model demonstrated modest uncovering performance for difficult-to-catch attacks, specifically those belonging to the U2R and R2L classes. Zhang et colleagues [13] used the UNSW-NB [14] standard dataset to illustrate the efficacy of ML-based intrusion detection using a full depiction of modern IoT attack scenarios. They employed a new feature selection engine that applied DAE founded on a biased loss function, despite using a simple MLP as an algorithm. This unique feature selection technique resulted in an increased emphasis on attack-representative features. Koroniotis et al. [15] proposed an IoT network forensic framework consisting of C4.5, ARM, NB, and ANN ML approaches to recognize and spot novel and complex forms of present botnet attacks as another application of the UNSW-NB dataset.

Traditional ML techniques and approaches have been widely used due to their high accuracy for attack detection and low false alarms, but they have been disapproved for their inability to detect innovative threats. Traditional ML techniques are incapable of identifying composite and new attacks. The mainstream of mutation attacks is minor alterations known as cyberattacks in modern times. The prior logic and conceptions serve as the basis for the novel attacks. This means that typical ML models will fail to recognize

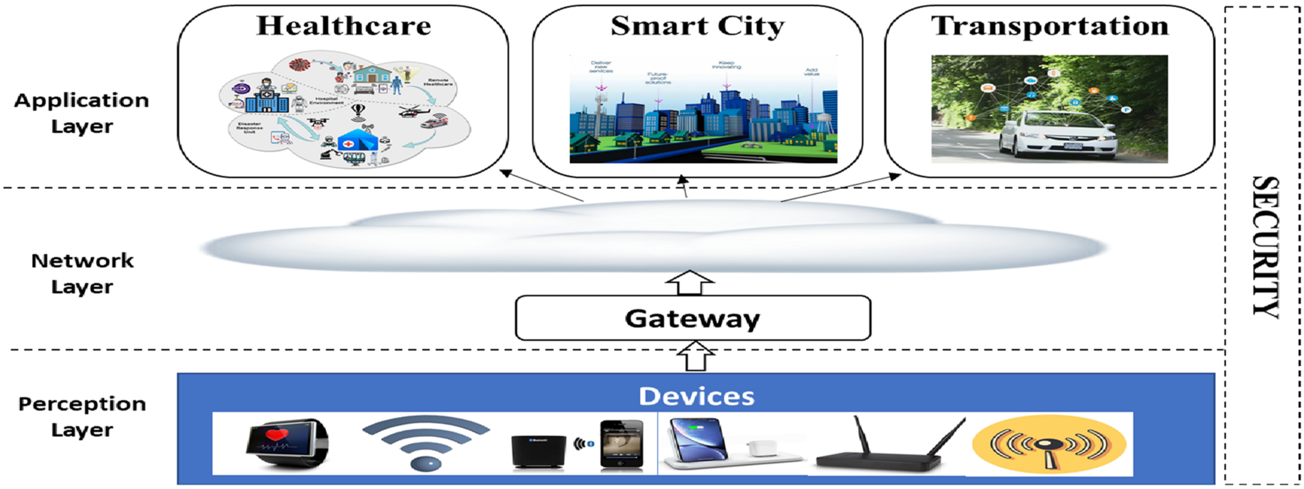

Fig. 1 The Internet of Things scenario

minute mutations because they are incapable of abstracting information to discern novel threats [16]. Hence, a more robust, intelligent method for IoT attack detection is needed. Therefore, this paper proposes an ensemble learning method. Ensemble learning for resilient IoT security is a strategy for solving a specific artificial intelligence-based challenge by combining different models or expertise. Ensemble learning enhances generalization, simplification, and voting among the various ensemble strategies in the intrusion detection problem, resulting in a higher detection performance than individual models [17]. The paper's primary contributions are as follows:

- Propose a new voting ensemble learning approach for IoT intrusion detection (To the best of our knowledge, this is the first voting GWO-optimized ensemble model for intrusion detection in the IoT).
- Analyze the model using feature extraction (principal component analysis) and feature selection (information gain) for dimensionality reduction. We created a hybrid IG + PCA technique for feature selection, feature extraction, and GWO-optimized ensemble models for classification tasks.
- Based on network traffic characteristics, low-cost and mountable cyber intrusion detection for IoT are proposed.
- Suggest several realistic datasets for IDS in the IoT environment.
- Develop a voting ensemble model based on the average of probability to increase the detection accuracy and decrease the false alarm rate to detect cyberattacks in the IoT.
- Leverage the realistic BoT-IoT and UNSW-NB15 datasets that reflect modern-day attacks and are representative of real-world attack scenarios in IoT which also satisfy IoT protocol requirements as against outdated and non-representative datasets used in some previous studies.

The paper is divided into seven sections. The literature and existing works are presented in Sect. 2. The proposed methodology is detailed in Sect. 3. Section 4 presents the GWO-optimized ensemble models, and Sect. 5 presents the experimental setup. The findings and discussions are given in Sect. 6. Section 7 contains the conclusion and recommendations for future work.

2 Related work

The study [18] employed bloom filtering for signature matching and offered a dynamic coding mechanism for constructing a decentralized signature-based IDS in IP-USN. The study [19] created a virtual test platform to mimic an actual network environment, installing a Snort IDS for traffic control and attack discovery by reflecting traffic to the server and constructing a stream-based IDS intelligent system using ML developed a specification-based IDS capable of identifying a novel sort of danger—the topology attack. They suggested an IDS architecture built on top of a network monitor and explained its monitoring techniques using an RPL FSM. Roy et al. [20] presented the use of a Bi-LSTM RNN for intrusion detection to spot a binary categorization of normal and malicious attacks. The model was trained on the UNSW-NB15 dataset and had a detection accuracy above 95% in IoT attacks. The work [21] devised an approach for detecting resource-constrained deep packet anomalies that distinguish between regular and anomalous payloads. Xu et al. [22] presented a unique IDS that examined the realization of several basic hybrid RNN models and MLP to protect against IoT threats. Both the NSL-KDD and KDD Cup 99 datasets are utilized for training and assessing the described models. The study [23] developed a several-layered RNN

Table 1 Summary of existing IoT attack detection using machine learning and deep learning

Authors	Methodology	Dataset	Results/strengths	Gaps
[39]	Single-layered ANN	N-BaIoT	Accuracy = 99	Building ten ANN models to identify an attack is a resource-intensive and time-consuming procedure
[40]	SMOTE and ANN	BoT-IoT	Accuracy = 100	Focus solely on detecting DDoS attacks. Additionally, a simple ANN with only one hidden layer was deployed
[41]	CNN	System call graph	Accuracy = 97% and F-measure = 98.33%	No experiment was conducted to determine the presence of other harmful lines on IoT devices
[20]	Bi-LSTM	UNSW-NB15	Accuracy = 95%	There was no optimization of parameters with long training time
[23]	RNN	NSLKDD	Probe = 97.35%; DoS: 98.27%; R2L: 77.25%; and U2R: 64.93%	It examined a single dataset without elucidating the tuning of the hyperparameters
[42]	K-NN, Gaussian Naive Bayes, and random forest	Capture live network	KNN: accuracy = 94.44, precision = 92.0, recall = 100, and F-measure = 96; Gaussian Naive Bayes: accuracy = 77.78, precision = 75, recall = 100, and F-measure = 86; and random forest: accuracy = 88.8, precision = 86, recall = 100, and F-measure = 92	The suggested approach is policy-based and relies on known attack signatures, signatures are upgraded
[24]	GRU, LSTM, BLS, and Bi-LSTM	NSLKDD	BLS performs better with an accuracy reaching 84.14% and F-measures = 84.68%	They considered only a single basic network data
[43]	GRU + MLP, BGRU-MLP, BLSTM + MLP, GRU, MLP, DLSTM + MLP, and LSTM	KDD Cup 99, NSLKDD	Accuracy = 99.24%	Several general attacks were discovered while examining a single dataset

[44]	SVM, RF, DT, and logistic regression	Capture live network	SVM = 98.06; RF accuracy = 99.17;
			DT = 98.34
			LR = 97.50
				It is difficult to replicate the research. The implementation details of the ML model are absent

Table 1 (continued)

Authors	Methodology	Dataset	Results/strengths	Gaps
[25]	SVM, J48, NB, MLP, NB, RF, RF, RNN-IDS, and ANN	NSLKDD	RNN-IDS accuracy = 95.2%	For performance comparisons, only machine learning models and outdated dataset were used for the experimental analysis
[27]	DJ, DF, DNN, LSTM, DBN, and GRU	NSLKDD, KDD Cup, and CICIDS	DBN gave an accuracy of 96.9% outperforming others	There are no realistic IoT datasets examined
Proposed GWO ensemble models	RF, DT, MLP, and KNN	BoT-IoT UNSW-NB15	Improved accuracy, F-measure, and ROC	We used multiple base classifiers, including RF, DT, MLP, and KNN, and designed a voting GWO ensemble model

model for IoT gadgets that might be deployed. The identification rates of attacks were determined to be DoS at 98.27 percent, the probe at 97.35 percent, U2R at 64.93 percent, and R2L at 77.25 percent, respectively, using the NSL-KDD dataset. Li et al. [24] used the NSLKDD dataset to build GRU, LSTM, BLS, and Bi-LSTM algorithms for several known intrusion classification tasks. According to the performance study, the BLS significantly reduces training time while maintaining an accuracy of 72.64% and 84.15 percent for the KDDTest-21 and KDDTest + data, respectively. The author [25] demonstrated an accuracy of 85.5 percent–95.25 percent for RNN-IDS using a heuristic technique for intrusion detection. The IDS is initially trained using the gradient descent approach and then retrained and tested using the KDD20 + and KDDTest + datasets. RNN-IDS outperforms various applied algorithms, including SVM, J48, NB, MLP NB tree, RF, ANN, and RF tree. In ref. [26], a DoS detecting design for 6LoWPAN was presented. This design incorporated an IDS into the ebbits framework created under the EUFP7 program. The paper [27] conducted an experimental investigation on intrusion detection utilizing DJ, DF, DNN, LSTM-RNN, DBN, GRU-RNN, and RNN of ML and deep learning models. Four datasets, namely, KDD Cup 99, NSLKDD, CICIDS2017, and CICIDS, were used to evaluate the algorithms' effectiveness in detecting and classifying anomalies using 22 distinct evaluation measures. However, the experiment results indicate that when DL models are combined with machine learning models, notably DBN, the detection accuracy rate increases from 5 to 10%. The study [26] set out to spot DoS attack protocols against CoAP and 6LoWPAN communication and to offer an IDS architecture for detecting and blocking attacks in an internet-connected environment. Jiang et al. [28] experimented with a mixed sampling-based intrusion detection method using the UNSW-NB15 and NSL-KDD datasets separately. The OSS and SMOTE are combined to create balanced data for training models built with CNN, AlexNet, BiLSTM, LeNet-5, and RF algorithms. According to the statistical result, CNN-BiLSTM surpassed other classifiers with an accuracy of 83.58%. Hasan et al. [29] addressed many paradigmatic machine learning strategies for spotting intrusions into IoT nets that result in system failure. On the DS2OS data, five-fold cross-validation was performed using LR, SVM, DT, RF, and ANN. Cheng et al. [30] developed an HS-TCN for detecting anomalous communication in the Internet of Things. The experiment was controlled using two variants of the unique dataset DS2OS: data collected over eleven (11) days and the DS2OS-UA. For both adjusted datasets, the HS-TCN model outperforms the LSTM and SVM models. The author [31] suggested an intrusion detection approach founded on node usage analysis in 6LowPAN. Sahu et al. [32] developed another machine learning-based method for detecting anomalies by combining LR and ANN classification methods. Both the ANN and LR achieve approximately 99.4 percent accuracy when the entire dataset is used and 99.99 percent accuracy when approximately 105,952 data points are omitted from the unique data. In both situations, the data are divided into 75 percent and 25% subsets. In reference [33], an event-processing IDS architecture based on CEP technology was described. Kalis [34], an adaptive expert IDS that can supervise several protocols without modifying existing IoT software, is a thorough approach for detecting IoT intrusions. Reddy et al. [35] described a DNN architecture for securing the apps of future smart cities. The findings demonstrate that this DNN technique achieves an accuracy of approximately 98.26 percent when compared to standard machine learning classifiers with a variable layer and neurons. The authors [36] developed a novel method for detecting network intrusions in IoT networks that are built on a conditional variational autoencoder with a specialized design that incorporates intrusion tags. To detect malicious activity, ref. [37] employed a single-class SVM equipped with characteristics such as memory utilization and CPU utilization. The study [38] examined the efficacy of many community detection methods for detecting P2P bots, particularly when only incomplete information is available. They demonstrated that the approach may be used with approximately half of the nodes, presenting their connection graphs with only a slight upsurge in detection mistakes. Table 1 summarizes the assessed studies on IoT security as per their datasets, models, best accuracy results, and gaps.

As seen from the review of the existing studies, the focus of some of the research is solely on detecting DDoS attacks. Other sizable attacks are not taken into account. Also, a simple ANN with only one hidden layer was deployed in one case with no optimization techniques applied. The majority of the work also lacks comparative analysis with other ML and DL models. In another study, it was difficult to replicate the research work. The implementation details of the machine learning model are absent, with obsolete datasets that do not reflect contemporary IoT attacks. Finally, the suggested approach is policy-based and relies on known attack signatures; hence, it will not be up-to-date with the most recent attack trends until signatures are upgraded.

Unlike the past efforts, we investigate intrusion detection for IoT resource-constrained devices in the network in this research. The difference is that our technique is divided into three stages. The first is hybrid dimensionality reduction, which involves using PCA and IG to choose the relevant attributes. The proposed GWO ensemble intrusion detection model includes two important engines in the second phase: a traffic analyzer and a classification phase engine. In the third phase, voting was utilized to merge the base learners' probability averages.

2.1 Motivation for the intelligent threat model on the Internet of Things

As IoT grows, so does the number of cybersecurity threats that investigators must address and examine to develop a reliable IDS. Numerous forms of malevolent action attempt to compromise the privacy and security of IoT gadgets, and all smart appliances connected to the Internet are potentially vulnerable. For a variety of reasons, the IoT is vulnerable to cyberattacks. For starters, IoT appliances are frequently unattended (for example, sensors located in remote places), making it relatively uncomplicated for an assailant to get admittance to them physically. Second, the vast majority of data transfers are wireless, making eavesdropping easier. Finally, most IoT devices have limited storage and computing capabilities [45]. Additional anti-virus protection, for example, cannot be deployed on IoT gadgets. Using numerous hacking tactics, hackers can disrupt or manipulate the functionality of smart gadgets [46]. In light of the physically insecure nature of a large number of IoT gadgets, some hacking approaches require active access to smart gadgets, making an attack more difficult but not impossible. Other attacks could be carried out remotely over the Internet. Table 2 shows the main kinds of attacks targeting smart devices.

The intrusion attacks can affect an IoT bot network comprised of unsecured IoT gadgets such as electrical gadgets, security systems, automobiles, thermostats, lights in-home or marketable locations, speaker systems, and wall timers. These attacks give a cybercriminal the ability to take control of the sensors. Unlike traditional botnets, compromised IoT devices actively seek to propagate their hateful behavior to a cumulative range of gadgets. While a traditional bot network may consist of hundreds of bots, IoT bot malware is far larger in scope, involving a large number of connected gadgets [51]. For instance, on October 21, 2016, cybercriminals targeted a prominent DNS firm named Dyn. This attack was initiated by a massive flood of DNS lookup queries from millions of IP addresses [52]. The bot network demands it infect a significant number of devices linked to the Internet, including printers, camcorders, and other gadgets. This IoT bot network attack was initiated by malevolent software known as Mirai. As a result of the Mirai contagion, computers continually search the Internet for susceptible gadgets and log in using the default username and password, attacking them with malicious programs. Researchers in the security field described how they targeted the Chrysler Jeep Cherokee at Black Hat 2015. While hacking the Jeep's IoT device and sensor network, one could remotely access the vehicle as it drove down the motorway [53]. The specific security challenges addressed in this research, which involves developing an IDS for the IoT using a hybrid approach of feature extraction via PCA, feature selection via IG, and parameter optimization using GWO for ensemble models, are related to the cybersecurity aspects of IoT environments. Firstly, about vulnerabilities in IoT devices, it is important to note that these devices frequently have limited resources and may lack comprehensive security measures. The primary objective of the IDS suggested in this study is to identify and address vulnerabilities present in these devices, hence thwarting unauthorized access and control. Furthermore, it is imperative to periodically upgrade the firmware and software of IoT devices to ensure their security. The suggested approach has the potential to facilitate monitoring and ensure the timely implementation of changes. Authentication and access control play a vital role in safeguarding IoT systems, as they are responsible for ensuring that solely authorized individuals or devices are granted access. The proposed IDS has the potential to effectively detect and identify unauthorized access attempts.

Table 2 Common types of attacks against smart IoT devices

References	IoT types of attack	Examples	Description
[47]	Attack on cloud infrastructure	Numerous cloud services contain a logical fault, which allows a cybercriminal to get delicate customer information as well as contact with the device without verification. These services also feature common management console susceptibilities	IoT devices connect to cloud services on the back end. Clients of IoT cloud services may be able to choose easy passwords
[48]	Attack on device	In the case of intelligent IoT devices such as surveillance cameras, a cybercriminal may gain direct knowledge of the equipment, allowing them to change the design settings	An attack is when someone exploits a defect or weakness in the IoT infrastructure to get access to it
[48]	Man-in-the-middle attack	Eavesdropping attacks such as man-in-the-middle are a sort of snooping attack. The attacker might use this approach to relay and possibly change interactions between two IoT devices invisibly	The attackers analyzed network traffic using a network packet analyzer, namely, Wireshark. IoT gadget interacts with additional IoT appliances. This link is neither encoded nor even authorized. This is the reason an attacker may easily target network access, allowing them to mount attacks such as ARP poisoning
[22]	Denial of service	An adversary can disable the sensors' capacity to transmit and receive data. Additionally, battery misuse, device disabling, or device botching are examples	A cybercriminal can disable or alter electronic equipment and its associated gadgets via physical or virtual access to the IoT sensors
[45]	IoT botnet attack	Mirai is regarded as a watershed moment in the latest threats because it leverages security flaws in IoT systems to launch attacks [49]	The term "IoT botnet" refers to a collection of compromised computers, smart gadgets, and utilities linked to the Web; these gadgets are the targets of attacks. They are mostly interested in attacking internet clients and devices, such as IP cameras and edge routers
[50]	Reconnaissance	This can be accomplished through the use of network port scanners and packet sniffers	The objective is to collect data on an IoT base, comprising network facilities and connected gadgets

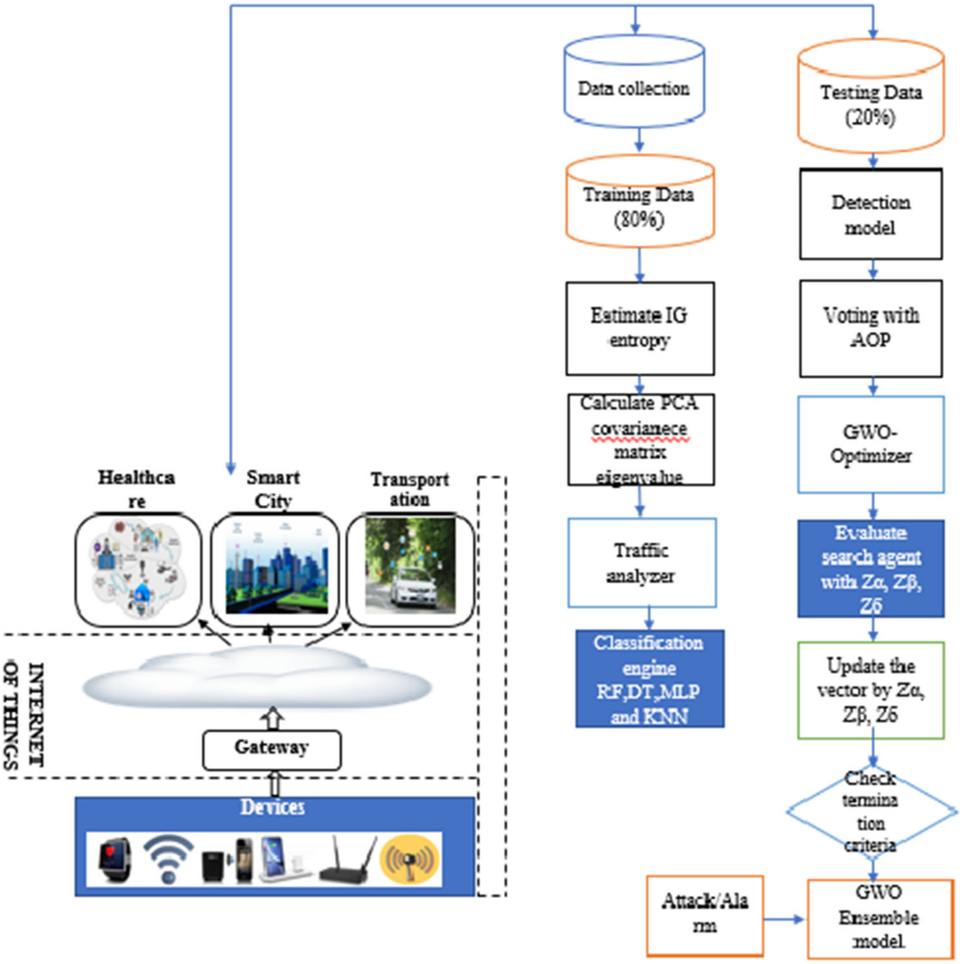

Fig. 2 The framework of the proposed GWO ensemble models for IoT

3 Methodology

This section discusses our proposed method's framework, philosophy, and design ideologies. In this research, a hybrid IG-PCA-based feature selection and extraction method employing optimized voting gray wolf optimizer-based ensemble learning models was proposed for intrusion detection in IoT. The general design of our suggested model is portrayed in Fig. 2, which is made up of three phases. The first phase is dimensionality reduction utilizing PCA and IG to control the relevant attributes. In the second phase, two key engines comprise the proposed ensemble intrusion detection model: a traffic analyzer and a classification (RF, DT, MLP, KNN, and voting ensemble) phase engine. The GWO evolutionary-based optimization was used for optimizing the parameters of the ensemble models. Preprocessing of traffic connection records in the circulation processing unit results in traffic data in a format appropriate for processing by the ensemble models of the classification phase, with these connections classed as normal or attacked by the GWO ensemble intrusion detection. In the third phase, voting was utilized to combine the average of the probability of the base classifiers. The new voting methodology employs GWO ensemble models to improve the legitimate/intrusion classification's prediction capacity. A probability average offers rapid reply and effective immediate safety management for the IoT system. Voting is a critical phase of the proposed classification-based traffic analysis; it analyzes network traffic that seeks to reach the IoT scheme and generates a security alert if an intrusion is identified. In the provided framework illustrated in Fig. 2, the data are trained using the IG approach, where the IG entropy is estimated. Following this, we proceed to calculate the eigenvalue of the PCA covariance matrix. During the testing phase, the voting process is conducted by calculating the average of probabilities obtained from the GWO-optimized ensembles, namely, RF, DT, MLP, and KNN. The voting mechanism is further enhanced by the utilization of vectors alpha, beta, and gamma, which are responsible for updating the voting process. In the context of an IoT setting, the process of data collecting encompasses not only the reception of data from IoT devices, but also the transmission of commands, updates, or responses back to these devices. The bidirectional flow of information is of utmost importance in facilitating real-time interactions and control inside IoT devices.

3.1 Data preprocessing

Normalization is a technique for scaling attributes in which the goal is to have all attribute values on the same scale normalization techniques include the standardized approach, min–max normalization, and *z*-score normalization [54, 55]. We selected the min–max normalizing technique since the majority of the features had a normal distribution to prevent information from leaking in the test data.

3.2 Normalization technique

The min–max approach [56] modifies a feature so that all of its values lie inside the interval [0,1]. Equation 1 depicts the fundamental formula for min–max normalization.

$$Y_{\text{new}} = \frac{y - \min\ (y)}{\max\ (y) - \min\ (y)} \tag{1}$$

where yi represents the value of a certain feature, *y* min represents its minimum value, and ymax represents its highest value.

3.3 Feature selection

The IoT ecosystem comprises intelligent devices with limited computing power, energy, communication range, and memory. Among the issues with IDSs are handling numerous irrelevant features, which might result in system overhead. Thus, the objective of feature evaluation is to discover key attributes that may be employed in the IDS to detect a variety of attacks efficiently. The characteristics are examined for both normal and pathological behaviors using the retrieved labels to select the most important features. We used an information gain (IG) strategy and principal component analysis (PCA) for feature extraction for feature selection.

3.4 Feature selection with IG

IG is a frequently used entropy-based feature evaluation approach in ML [57]. The information gain techniques were rapid to execute, and this strategy extracted the model's optimal feature set. IG was frequently used in the literature to determine how successfully each different attribute distinguished the assumed data. The first phase in this research is to use IG plus ranked as a filtering strategy to lower the datasets' dimensionality. The primary idea behind this method is to evaluate subgroups of features by estimating their IG entropy in decreasing order. From most relevant to least relevant, each feature receives a score. The attributes with the best scores are used as the input set of attributes for the next dimensionality reduction stage. The author [58] describes the overall entropy "*K*" of a given dataset "*D*" as follows:

$$K\ = (D)\ = -\sum_{i=1} pi\,Log2Pi \tag{2}$$

where "*e*" signifies the total class size, and "pi" denotes the percentage of cases belonging to class u. The reduction in entropy in information is estimated for each feature using the following formula:

$$\text{IG}\ (D,\ M)\ =\ K\ (D) - \sum_{w\varepsilon A} \frac{|DA,\ w|}{|D|} K(Dw) \tag{3}$$

3.5 Feature extraction with PCA

The IG method's specified attributes can be utilized directly for categorization. However, one of the most typical IG issues is a preference for traits with various possible numbers [59]. These features have a close-zero eigenvalue in this scenario, which improves their gain more than another attribute. As a result, the full importance of these attributes to the training examples may not be represented in their ranking. To overcome this constraint, features from the attribute selection phase will be presented for additional reduction using the PCA method to identify the best subgroup of features. This allows the PCA to narrow the search area from the whole subspace to the features that have been pre-selected [60]. The purpose of using PCA is to minimize dimensionality by retaining important attribute information in the data. It decreases the number of variables by employing orthogonal combinations with significant variance. Table 3 shows the proposed hybrid dimensionality reduction for our suggested models.

Two techniques are employed to reduce the dimensionality of features from m dimensions to j dimensions: preprocessing and dimensionality reduction. During the preprocessing phase, the mean and variance of the data are standardized using Eqs. (3) and (4) (steps 1 via 4 below). During the second phase (steps 5–8), the covariance matrix Cov_n, eigenvectors, and eigenvalues are constructed using Eqs. (5) and (6).

1 Standardize the initial input feature values by their mean and standard deviation using Eq. (4), where n is the number of cases, and $Y_{(i)}$ is the data points.

$$\mu = \frac{1}{n} \sum_{i=1}^{n} Y_{(i)} \tag{4}$$

2. Substitute $Y_{(i)}$ with $Y_{(i)} - \mu$.

Table 3 Hybrid feature dimensionality reduction

```
Algorithm 1   The proposed hybrid feature dimensionality reduction IG+PCA
Y is the input dataset, and (Y comprises m cases with its consistent S attributes)
procedure Calculate_IG (Y)
    Estimate the information needed to classify a specific case
while 1 ≤ j ≤ m do
    Calculate the entropy of an attribute $S_j$
    Calculate the IG for the feature   $S_j$
    The g qualities with the greatest scores are $\longrightarrow$X
    return X
procedure Calculate_PCA (X)
    Calculate X's covariance matrix
   Calculate the covariance matrix's eigenvector ($k_1,k_2,k_3...,k_j$) and eigenvalues ( $\rho 1,..., \rho j$ )
   The g eigenvectors with the highest eigenvalues are called R.
return R
End
```

3. Using Eq. (5), transform each vector $Y_{k(i)}$ to have unit variance.

$$\sigma_i^2 = \frac{1}{n}\sum_i \left(Y_{k(i)}\right)^2 \tag{5}$$

4. Substitute each $Y_{k(i)}$ with $\frac{Yk(i)}{\sigma}$.
5. Computation of the covariance matrix Cov_n:

$$\mathrm{Cov}_n = \frac{1}{n}\sum \left(Y_{(i)}\right) Y_{(i)})^T \tag{6}$$

6. Cov_n eigenvectors and eigenvalues are calculated.
7. Set eigenvectors by diminishing eigenvalues and select j eigenvectors with the greatest eigenvalues to produce S.
8. Using S and Eq. 7, convert the data to the novel subspace.

$$Y = S \times X \tag{7}$$

where Y is a $1 \times e$ vector on behalf of one sample, and y is the converted $j \times 1$ sample in the new subspace.

The computational difficulty of performing the specified PCA is proportional to the number of attributes F representing each point of data.

$$O\left(F^3\right) \tag{8}$$

In this study, PCA is utilized to reduce the dimensionality of the BoT-IoT and UNSW-NB15 datasets by compressing the attribute space with ten (10) selected features and nine (9) high-rank features, respectively. The ten (10) and nine (9) top-ranked features were considered for the BoT-IoT and UNSW-NB15 datasets. To identify the most effective features, we employed information gain, used in our feature selection process, which quantifies the importance of each feature based on its ability to discriminate between different classes (e.g., normal and intrusions). Features with higher information gain were considered more effective in distinguishing between classes. The design principle of PCA is given in Table 4.

Table 4 Design principles of PCA

Parameter	Values
Parameter ranking	True
Num to select	6
Threshold	0.5
Variance	1.832

Parameter ranking typically refers to the process of assessing and ranking the importance or influence of different parameters or hyperparameters on a machine learning model's performance. These parameters are settings or configurations that can be adjusted to influence how a model learns from data and makes predictions. In our research, the parameter ranking in the settings is set to true. The num to select parameter in PCA is set to the value 6. The threshold value is set to 0.5, and the variance is set to 1.832. The design principle revolves around finding a new set of orthogonal axes, called principal components, that capture the maximum variance in the data while reducing its dimensionality.

Ten (10) new features were selected from the BoT-IoT dataset, and nine (9) features were chosen from the UNSW-NB15 which are subsequently fed and passed to the GWO-optimized ensemble models (RF, DT, MLP, and KNN). The information gain efficiently identifies the most relevant features based on their contribution to the target variable, while PCA optimally captures the variance within the dataset to create a reduced set of orthogonal features. By combining these two methods, we achieve a balanced feature reduction approach that maximizes the preservation of informative features while minimizing computational overhead.

PCA aims to transform the original high-dimensional feature space into a lower-dimensional space while retaining as much of the variance in the IoT network traffic data as possible. This dimensionality reduction can lead to several benefits:

i. *Curse of Dimensionality* High-dimensional IoT network traffic data can suffer from the "curse of dimensionality," where the number of features greatly exceeds the number of samples. This can lead to increased computational complexity, overfitting, and difficulty in visualization. PCA helps mitigate these issues by reducing the dimensionality.
ii. *Noise Reduction* High-dimensional IoT network data often contain noise and irrelevant features. PCA helps remove and down-weight such noisy dimensions by identifying and emphasizing the dimensions with the most significant information.
iii. *Improved Model Performance* Reducing dimensionality leads to faster training and inference times for machine learning models, as well as potentially reducing overfitting.

3.6 Handling the class imbalance problem

Addressing class imbalance is a prevalent issue encountered in the field of machine learning, particularly in the context of intrusion detection systems. This challenge arises due to the substantial disparity between the abundance of normal instances and the scarcity of attack instances. In this research, we employed the synthetic minority oversampling technique (SMOTE) as a method to tackle the aforementioned concern. The SMOTE is a method that produces artificial cases for the underrepresented class by interpolating between the available data points. We ensure that the data are preprocessed properly, including removing irrelevant features, handling missing values, and encoding categorical variables. Subsequently, we divide the datasets into features (x) and corresponding labels (y) for both training and testing datasets. Thus, we create an instance of the SMOTE and apply it to the training data. The mathematical representation is given in Eq. (9).

$$x\_\text{ synthetic } = x\_\text{ minority } + \text{random}\_\text{ number} * \left(n - x\_\text{ minority}\right) \tag{9}$$

Assume there exists a dataset with features x and labels y. For each minority instance x_minority, there is a need to find its K-nearest neighbors from the minority class. The distance metric used for finding neighbors (such as Euclidean distance) can vary. Assume we denote the set of k-nearest neighbors as $N(x\text{\_minority})$. For each neighbor n in N (x_minority), a synthetic instance x_synthetic is generated as Eq. (9).

At this juncture, random_number is a random value between 0 and 1, controlling the interpolation between x_minority and n. The formula in Eq. (9) is applied to each feature of x_minority and n to generate the corresponding feature of x_synthetic.

3.7 Optimization of the ensemble learning models (ELM) with gray wolf optimizer

The GWO methodology is a metaheuristic algorithm that replicates the initiative chain of importance and pursues the method of dark posers [61]. In the numerical method for the GWO, the optimal configuration is denoted by the symbol alpha α. The beta (β) and delta (δ) are optimized according to the second- and the third-best configurations, respectively. It is believed that the remaining application setups are known as omega (ω). These three applicants are being pursued by β,δ, and ω using GWO tactics and α as a hunting guide.

For the pack to pursue prey, they immediately encircle it. The following Eqs. (10)–(13) are applied to mathematically model surrounding behavior.

$$\overrightarrow{Z}(r+1) = \overrightarrow{Z}_p(r) + \overrightarrow{B}.\overrightarrow{E} \tag{10}$$

$\overrightarrow{Z}_p$ is the position of the prey, $\overrightarrow{Z}$ is the gray wolf position, $\overrightarrow{B}$ and $\overrightarrow{D}$ are coefficient vectors, and r is the number of iteration number E as shown in Eq. (11)

$$\overrightarrow{E} = \left|\overrightarrow{D}.\overrightarrow{Z}_p(r) - \overrightarrow{Z}(r)\right| \tag{11}$$

$$\overrightarrow{D} = 2b.\ \overrightarrow{t}_1 - b \tag{12}$$

$$\overrightarrow{D} = 2\overrightarrow{t}_2 \tag{13}$$

b is lowered linearly from 2 to 0 throughout the emphasis span, while t_1 and t_2 are random vectors in the interval [0, 1]. Typically, the alpha leads the pursuit. Moreover, the beta and the delta may occasionally be interested in chasing. To scientifically emulate the chasing behavior of gray wolves, the alpha (the best candidate solution), beta (the second-best rival solution), and delta (the third-best optimistic solution) are accepted to obtain more information regarding the likely prey position. The initial three best application configurations have reached this stage, necessitating that the other hunt operators change their situations to match those of the best pursue experts. Therefore, the replenishment of the positions of the wolves is provided by Eq. (14):

$$\overrightarrow{Z} = (\mathrm{r} + 1) = \frac{\overrightarrow{Z}1 + \overrightarrow{Z}2 + \overrightarrow{Z}3}{3} \tag{14}$$

$$\overrightarrow{Z}_1 = \left|\overrightarrow{Z}\alpha - \overrightarrow{B}_1.\ \overrightarrow{E}_a\right| \tag{15}$$

$$\overrightarrow{Z}_2 = \left|\overrightarrow{Z}_\beta - \overrightarrow{B}_2.\ \overrightarrow{E}_\beta\right| \tag{16}$$

$$\overrightarrow{Z}_3 = \left|\overrightarrow{Z}\delta - \overrightarrow{B}_3.\ \overrightarrow{E}_\delta\right| \tag{17}$$

where $\overrightarrow{B}_1$, $\overrightarrow{B}_2$, and $\overrightarrow{B}_3$ are defined as Eq. (14) and $\overrightarrow{Z}\alpha, \overrightarrow{Z}_\beta$, and $\overrightarrow{Z}\delta$ are the leading three best solutions in the assumed iteration r, $\overrightarrow{B}_1$, $\overrightarrow{B}_2$, and $\overrightarrow{B}_3$ are expressed in Eqs. (15–17), and $\overrightarrow{E}\alpha$ and $\overrightarrow{E}\delta$are expressed as Eqs. 18–20, respectively.

$$\overrightarrow{E}_\alpha = \left|\overrightarrow{D}_1.\ \overrightarrow{Z}_1 - \overrightarrow{Z}\right| \tag{18}$$

$$\overrightarrow{E}_\beta = \left|\overrightarrow{D}_2 - \overrightarrow{Z}_\beta - \overrightarrow{Z}_1\right| \tag{19}$$

$$\overrightarrow{E}\delta = \left|\overrightarrow{D}_3.\overrightarrow{Z}\delta - \overrightarrow{Z}_1\right| \tag{20}$$

$\overrightarrow{D}_{\mathbf{1}}$, $\overrightarrow{D}_{\mathbf{2}}$, and $\overrightarrow{D}_{\mathbf{3}}$ are given as in Eq. (13)

A final observation regarding the GWO mediator is the updating of the parameter that regulates the investigation-abuse tradeoff. The stricture is continuously updated each cycle to range from 2 to 0 following Eq. (21).

$$b = 2 = r\frac{2}{\mathrm{Maxlter}} \tag{21}$$

where MaxIter is the full number of allowable optimization iterations, and r is the number of optimization iterations. The hunting and pursuit positions of gray wolves are required to be updated by binary {1, 0}. The gray wolf optimization pseudocode is described in Table 5.

We chose GWO to optimize the parameters of the ensemble algorithms because of three significant merits; exploration and exploitation, convergence speed, and handling constraints, which it has over other algorithms. GWO has gained a significant amount of prominence among other swarm intelligence methodologies due to its various characteristics such as fine-tuning parameters, simplicity and ease of use, scalability, and most notably its ability to just provide convergence speed by maintaining the right balance between exploitation and exploration during the search. GWO exhibits a better balance between exploration (searching the solution space) and exploitation (exploiting promising solutions). It uses the concept of alpha, beta, gamma, and delta wolves to strike a balance between exploration and exploitation which can lead to more efficient optimization compared to other algorithms. GWO tends to converge faster to a global optimum compared to several other algorithms in some cases. The nature-inspired hunting behavior of gray wolves, such as encircling prey, mimicked in GWO can lead to more efficient exploration and faster convergence. GWO promotes diverse solution exploration due to its hierarchical structure and the hunting behavior of gray wolves. This can help avoid getting stuck in local optima and facilitate a more comprehensive search of the solution space.

Table 5 Pseudocode of gray wolf optimization

1	Initialize values for the population size s, the Maxitrcoefficient parameter, and the D and B vectors
2	Create an initial population sample at random $Z_j(r)$
3	Using $f\ (z_j)$ to evaluate each search agent's fitness
4	$Z\ \alpha$, $Z\ \beta$, and $Z\ \delta$ to determine the values of the 1st, 2nd, and 3rd optimal solutions
5	Repeat
6	For ($j = 1$: $j \leq s$) do
7	Applying Eq. (21) to restore each population agent
8	End for
9	The vector has been updated by $Z\ \alpha$, $Z\ \beta$, and $Z\ \delta$ accordingly
10	Set $r = r + 1$
11	As soon as, the termination criteria are met till ($r \geq$ Maxitr)
12	Lastly to produce the optimal solution Z_a

In our research, the GWO is utilized to optimize the parameters of RF, DT, MLP, and n for KNN. Gray wolf optimizer (GWO) is a nature-inspired optimization algorithm that simulates the hunting behavior of gray wolves to find optimal solutions. We utilized the pseudocode of GWO to optimize the hyperparameters of ensemble learning models; random forest, decision tree, multilayer perceptron (MLP), and K-nearest neighbor (KNN) [62]. Here's a high-level overview of how we integrated GWO with ensemble models:

1. Initialize a population of gray wolves with random hyperparameter settings for the ensemble models.
2. Define a fitness function that evaluates the performance of the ensemble model with the given hyperparameters.

The fitness function used appropriate evaluation metrics.

3. In each iteration of the GWO loop, evaluate the fitness of each wolf (hyperparameter set) using the ensemble model. Update the positions of the alpha, beta, and delta wolves based on their fitness values. These wolves represent the best solutions found so far.
4. Update the positions of the other wolves using predefined formulas that simulate the hunting behavior of gray wolves. This step helps explore the search space efficiently.
5. Apply boundary constraints to ensure that hyperparameters remain within valid ranges for the ensemble models.
6. After a certain number of iterations or when a stopping criterion is met, select the best solution found so far based on fitness values.
7. Perform cross-validation to assess the performance of the ensemble model with the selected hyperparameters on a validation set.
8. If the new solution (hyperparameters) is better than the previous best solution, update the best solution.
9. Continue the optimization process until the stopping criterion is met.
10. Finally, return the best solution, which represents the optimal hyperparameters for the ensemble learning models.

By integrating GWO with ensemble models in this way, we effectively search for the best hyperparameters to maximize the ensemble's performance, improving its accuracy and effectiveness in real-world applications.

3.8 Mathematical formulation of the ensemble method for classification

Let $\{y(u)\}$ for $u = 1,\dots,m$ be a randomized data containing its associated examples and characteristics with a mean of zero. Equation (22) shows the covariance matrix of $y(u)$. Algorithm 1 summarizes the hybrid IG-PCA approach's selection procedure.

$$Z = \frac{1}{m-1} \sum_{u=1}^{m} \left[y(t) \times (u)^{U} \right] \tag{22}$$

In PCA, the transformation function from $y(u)$ to $x(v)$ is calculated as follows;

$$x(u) = N^{u} \times (u) \tag{23}$$

The jth column of the covariance sample matrix Z is equal to the jth eigenvector, and N denotes an $m \times m$ orthogonal matrix. The eigenvalue problem stated in Eq. (24) is initially fixed through PCA.

$$\beta_j k_j = Z k_j \tag{24}$$

where β_j signifies an eigenvalue of Z (say $\beta_1 > \beta_2 > \dots > \beta_m$), and k_j is the corresponding eigenvector. The PCA is obtained using Eq. (25) as follows:

$$x_j(u) = k_j \times (u),\ j = 1, 2, \dots, m. \tag{25}$$

The jth principle component is denoted by $x_j(v)$. The computation to project a fresh sample $y(u)$ onto the main space is given in Eq. (26). Let

$$y(u) = \sum\nolimits_{j=1}^{q} b_j U \times (u)_j^{a}, \tag{26}$$

where $A = \{e_j: e_j = k_j, j = 1,\dots, g\}$. Equation (27) calculates the distance f from $y(u)$ and (t) to determine the projection inaccuracy of $y(u)$ and $\acute{Y}(u)$:

$$b = f\left(y(u), \acute{Y}(u) \right) \tag{27}$$

3.9 Ensemble model

Ensemble methods are effective ways of improving the prediction outcome of the overall model by developing numerous self-reliant models and integrating them to provide results with improved, enhanced accuracy [63]. Ensemble learning approaches include boosting, bagging, Bayesian parameter averaging, and stacking [64]. This work proposes a unique ensemble classifier to improve intrusion detection accuracy in IoT that employs RF, DT, MLP, and KNN learners. These algorithms were utilized in a voting algorithm and were combined using the average of probabilities method. To accelerate the performance of each of the models, the GWO was used to optimize the parameters of each of the ensemble (RF, DT, MLP, and KNN) models.

Assume we have ϕ 'classifiers $A = \{A_1, A_{2,} \dots A\ \phi\}$ and l labels $= \{h_1, h_{2,} \dots, h_{\mathrm{l}}\}$. According to the classifiers given above, $\phi = 4$, and $\mathrm{l} = 2$ (that is, non-attack and attack) for the datasets analyzed in this work. $A_j: Z^m \rightarrow [1,0]^{\mathrm{l}}$ is a classifier. l takes an object y Z^M and returns a vector $[J_{Aj}\ (h_1|\mathrm{y}),\dots, JA_j\ (h|\mathrm{y})]$, where $J_A\ (h|\mathrm{y})$ represents the probability given by A_i to the assumption that entity y corresponds to class i. Where $\mathrm{n_i}$ becomes the average of the probabilities provided by the different classifiers for every class h_i,

$$n_i = \frac{1}{\phi} \sum\nolimits_{j=1}^{\phi} Jaj(h/y) \tag{28}$$

Let N denotes the collection of mean probability for each category ($n_{1,}$ n_2,…, n_c). Object y is classified correctly in N with the highest mean, i.e., y is allocated to class g if and only if

$$n_g = \max N \tag{29}$$

The proposed ensemble approach's performance is evaluated using two famous intrusion detection assessment data that are ideally suited for IoT, namely, BoT-IoT and UNSW-NB15.

3.10 Ensemble learning strategy

Ensemble learning is a powerful technique that combines multiple individual learning algorithms to create a stronger, more accurate predictive model. Voting-based ensembles are a popular approach within ensemble learning. In this research, we performed the average of probabilities from multiple models for intrusion detection in the IoT using the BoT-IoT and UNSW-NB15 datasets. Here's a step-by-step explanation of how we achieved this:

Step 1: Data Preparation We preprocess and split the datasets (BoT-IoT and UNSW-NB15) into training and testing subsets with the target labels (intrusion or non-intrusion) and the corresponding features for each dataset.

Step 2: Individual Learning Algorithms Choose a set of individual learning algorithms RF, DT, MLP, and KNN that we want to ensemble.

Step 3: Train Individual Models For each selected individual learning algorithm RF, DT, MLP, and KNN. We trained all these algorithms on training data from both datasets (BoT-IoT and UNSW-NB15). This gave us a set of trained models, each capable of making intrusion detection predictions.

Step 4: Probability Prediction For each trained model, we use it to make predictions on our testing data. Instead of just obtaining the final prediction label, we are interested in the predicted probabilities of intrusion (class attack) for each instance.

Step 5: Ensemble Voting For each instance in our testing data, we calculated the average of the predicted probabilities from all the individual models. This average can be computed for class 1 (intrusion).

Step 6: Evaluation We evaluated the performance of our voting ensemble models using standard metrics such as accuracy, DR, precision, ROC, and FAR on our testing data. We also compare these results with the performance of individual models to assess the effectiveness of the ensemble.

3.11 Benefits of the proposed voting-based ensembles model

- **Reduced Bias** Combining multiple models can help reduce bias present in any individual model.
- **Improved Generalization** Ensembles often perform better on unseen data compared to individual models.
- **Robustness** Ensemble methods are more robust against overfitting, especially if the individual models are diverse.
- **Model Diversity** Using different learning algorithms ensures that the ensemble captures different aspects of the data.

4 Experimental setup with the software and hardware requirements

The simulations are executed on a laptop with an Intel Core (TM) i5-8250U processor clocked at 1.60 GHz and 8 GB of RAM. To demonstrate the efficacy of the proposed approach, four GWO ensemble models (RF, DT, MLP, and KNN) with an average probability are chosen. The algorithms are used to classify and identify threats and anomalies across all the BoT-IoT and UNSW-NB15 datasets. Scikit learning was utilized in the implementation of the models.

4.1 Metrics used for performance evaluation

This study evaluated the performance of the proposed system using multiple performance measures, including precision, recall, dtection rate (DR), and accuracy (Acc), as well as the time required to create the model. These metrics' definitions are provided below. True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) determine the metrics (FN).

Detection rate (DR): The DR is the proportion of identified attacks relative to the total number of attack events in the dataset. Equation (30) can be utilized to estimate DR.

$$\text{DR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{30}$$

Accuracy is the measure of the classifier's ability to correctly classify an object as normal or as an attack. The accuracy is defined by Eq. (31).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{31}$$

Precision is the ratio of positive predictions to the total number of positive anticipated class values. It considered a measure of the classifier's precision. A low value represents a high number of FP. The precision is computed using Eq. (32).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{32}$$

The recall is calculated by dividing the number of TP by the number of TP and FN. The recall is regarded as a measure of a classifier's completeness, with a low recall value resulting in a large number of FN [65]. Using equation, recall is estimated (33).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{33}$$

4.2 Description of the dataset

One of the primary challenges encountered in the domain of anomaly detection research revolves around obtaining or generating a suitable dataset for experimental endeavors. In this study, we analyzed pre-existing datasets to identify the dataset that is most appropriate for further exploration. The authors delineated the dataset prerequisites identifying anomalies in the IoT by the following four criteria:

C1 The acquisition of the dataset ought to be conducted from the IoT;

C2 It is recommended that the dataset includes anomalies;

C3 The dataset must be appropriately labeled to distinguish between normal and abnormal data;

C4 It is recommended that the dataset utilized in the study closely approximates real-world data, specifically data derived from authentic or partially authentic systems.

C5 It is recommended that the datasets encompass a diverse range of attack scenarios and network conditions. A key criterion was the inclusion of a wide variety of attack types and patterns to ensure a comprehensive evaluation of our intrusion detection system.

C6 Took into account the accessibility and availability of the datasets to the research community. It was important to select datasets that are publicly accessible, well-documented, and readily available for replication and validation by other researchers.

The datasets that meet the specified criteria, namely, those that comprise labeled sensors, actuators, and network data, include the recently developed BoT-IoT and the UNSW-NB15 dataset. These datasets were subjected to a comprehensive analysis by the authors. The particulars of each dataset are delineated as follows;

4.2.1 BoT-IoT dataset

The BoT-IoT contains both typical IoT net traffic and a range of attacks. These data were utilized to test our system. It was chosen because it accurately depicts an IoT ecosystem context. DoS, DDoS, data exfiltration, keylogging, service scan, and OS attacks are included in the dataset. The BoT-IoT is available at https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/bot_iot.php. All of these data were preprocessed to establish network-level patterns for the varied types of traffic generated by devices and to use these similarities to spot attack behavior in the IoT architecture [51]. Table 6 summarizes the amount of benign and attack samples in the collection.

Table 6 Attack and normal behavior statistics from the BoT-IoT dataset

Attack and normal behavior	Values
DDoS	2766
Reconnaissance	298
Keylogging	73
Normal	8945

4.2.2 UNSW-NB15 dataset

The researchers [14] created the UNSW-NB15 dataset at UNSW Canberra. The researchers used the IXIA perfect storm to create a mix of benign and malicious traffic, yielding a 100 GB dataset in the form of PCAP files, including many novel attributes. The generated data were intended to be utilized for intrusion detection generation and validation. Nevertheless, the data were created using a simulated environment to generate attack activity. The UNSW-NB15 dataset record distribution is specified in Table 7.

Table 7 UNSW-NB15 data records

Feature type	Number of records
Fuzzers	24,246
Backdoors	2329
Analysis	2677
Exploits	44,525
DoS	16,353
Generic	215,481
Reconnaissance	13,987
Worms	174
Shellcode	1511
Normal	2,218,761

5 Results and discussion

We present the detailed findings of experiments conducted utilizing the proposed framework in this section. The suggested approach was tested on the datasets mentioned above.

Table 8 Confusion matrix

	Attack/intrusion	Non-attack/legitimate
Attack/intrusion	TP	FN
Non-attack/legitimate	FP	TN

The oversampling without replacement method was used to divide each dataset's selected samples into two distinct subgroups for training and testing. As a result, the training subset can accurately predict model performance on previously unrecognized data, and the testing sample is reserved for assessing the model's performance. In this instance, generating subgroups for cross-validation evaluation is not essential, which could be time-consuming with large datasets. Two tests were conducted to evaluate the efficiency of the presented technique. The following evaluation metrics were used according to the confusion matrix shown in Table 8: precision, accuracy, detection rate, ROC, and FAR. The authors [66] explain the mathematical computations for the measurement methods used.

Where TP is the number of current attacks recognized as attacks, TN is the number of frequent patterns identified as regular, FN is the series of attacks identified as frequent patterns, and FP is the number of frequent patterns identified as threats.

5.1 Experimental analysis based on BoT-IoT dataset

The BoT-IoT dataset was used in the first experiment. To begin, vital attributes were determined by computing the IG entropy for every feature in declining order. From the original thirty-one (31) potential features, ten (10) were chosen for the following step. The strategy was seen to create several FARs by deploying IG alone. To overcome this constraint, a second additional reduction phase founded on the selected attributes was done using the PCA as feature extraction. To evade bias, the PCA was created using only the training set, ensuring that no information from the test data was leaked into the training dataset. When genuine new unseen data are introduced into the model, the model will not function as well if the complete dataset is used to construct the PCAs. Similarly, calculating PCAs on the two sets independently will result in two mismatched sets of data. We cannot build a classifier in one domain and then apply it to another. The same characteristics from the training set were utilized to translate the testing dataset into the same feature space using the batch-filtering method. The new datasets were utilized to assess the efficiency of the presented method, so five separate classifiers were built utilizing the training data and classified using the testing dataset. On the BoT-IoT dataset, Table 9 compares the performance of standard ML models IG + PCA-RF, IG + PCA-DT, IG + PCA-MLP, IG + PCA-KNN, and the proposed voting GWO ensemble model. The results indicate that the voting GWO ensemble model performs the best, with an accuracy of 99.98% and DR of 99.97%, precision of 99.94%, ROC of 99.99%, and FAR of 1.30.

5.2 Experimental analysis based on UNSW-NB15 dataset

Additional tests on the UNSW-NB15 dataset were carried out to demonstrate the efficiency of the suggested feature dimensionality reduction (IG + PCA) GWO ensemble model. As in the first experiment, IG and PCAs were computed during the preprocessing step of these datasets. In this second experiment, nine (9) candidate features were chosen from UNSW-NB15 by computing the entropy of the IG and, subsequently, the PCA feature extraction. Table 10 shows the best results obtained using the reduction of dimension approaches on the dataset. Our proposed model produces promising classification results, as seen in the result. Table 10 compares the performance of the IG + PCA-RF, IG + PCA-DT, IG + PCA-MLP, IG + PCA-KNN, and the proposed GWO ensemble model on the UNSW-NB15 dataset. The voting GWO ensemble technique outperforms all other approaches, with an accuracy attaining 100%, DR of 99.99%, precision of 99.59%, ROC of 99.40%, and FAR of 1.15.

5.3 Multiclass experimental analysis on the BoT-IoT dataset

The initial step was the computation of the IG entropy for each characteristic, with the resulting values being arranged in descending order to identify the most significant qualities. Out of the initial set of thirty-one (31) possible features, a subset of ten (10) features was selected for the subsequent stage. The implementation of IG in isolation was observed to generate several FARs as part of the strategy. To address this limitation, a secondary reduction phase was implemented, utilizing the specified features and employing PCA as a feature extraction technique. To mitigate bias, the PCA was conducted exclusively on the training dataset, to preventing any potential leakage of information from the test data into the training set.

Table 11 shows the performance of the proposed voting GWO ensemble model on BoT-IoT in a multiclass scenario. The results indicate that the voting GWO ensemble model performed on DDoS HTTP achieved an accuracy of 99.87% and DR of 99.89%, precision of 99.60%, ROC of 99.56%, and FAR of 1.20.

Table 9 The performance of standard ML approaches and the proposed voting ensemble model on BoT-IoT

Classifier	Accuracy	DR	Precision	ROC	FAR
IG + PCA-RF	97.00	99.10	97.0	98.0	2.32
IG + PCA-DT	93.00	98.90	96.0	97.0	3.89
IG + PCA-MLP	95.00	98.0	97.0	98.0	4.83
IG + PCA-KNN	98.30	97.30	98.90	98.40	3.70
Proposed IG + PCA-Voting GWO ensemble $_{\text{Average of probability}}$	**99.98**	**99.97**	**99.94**	**99.99**	**1.30**

Values of our proposed model are in bold

Table 10 The performance of standard ML techniques and voting ensemble model on the UNSW-NB15

ML approaches	Accuracy	DR	Precision	ROC	FAR
IG + PCA-RF	98.14	99.20	99.20	98.10	3.40
IG + PCA-DT	97.00	99.12	98.40	97.81	5.20
IG + PCA-MLP	98.23	98.70	98.80	96.83	4.31
IG + PCA-KNN	97.80	99.70	98.80	98.30	3.79
Proposed IG + PCA-Voting GWO ensemble $_{\text{Average of probability}}$	**100**	**99.99**	**99.59**	**99.40**	**1.15**

Values of our proposed model are in bold

Table 11 Performance of the voting GWO ensemble model relative to the different attack types and benign in terms of DR, accuracy, and training time on the BoT-IoT dataset

Type of attack	Accuracy	DR	Precision	ROC	FAR
Benign	99.82	98.67	99.18	99.90	3.18
OS fingerprinting	98.41	99.86	99.28	99.18	4.28
Service scanning	98.67	98.87	99.68	99.68	3.89
DoS TCP	99.62	99.78	98.81	99.10	1.89
DoS HTTP	99.89	98.77	99.72	98.10	1.01
DoS UDP	98.84	98.89	99.83	98.53	1.10
Data theft	99.99	98.97	99.78	98.05	2.60
Keylogging	98.76	99.45	99.09	99.12	2.80
DDoS UDP	99.56	99.58	98.68	99.68	1.90
DDoS TCP	99.83	99.60	98.10	99.32	1.59
DDoS HTTP	**99.87**	**99.89**	**99.60**	**99.56**	**1.20**

Values of our proposed model are in bold

5.4 Multiclass experimental analysis on the UNSW-NB15

Further experiments were conducted on the UNSW-NB15 dataset to showcase the effectiveness of the proposed ensemble model, which combines feature dimensionality reduction techniques (IG + PCA) with the GWO. Similar to the initial experiment, the datasets underwent preprocessing in which IG and PCAs were generated. In the subsequent experiment, a total of nine (9) candidate features were selected from the UNSW-NB15 dataset by evaluating the entropy of the information gain (IG) and subsequently applying PCA for feature extraction. Table 12 shows the performance of the proposed voting GWO ensemble model on BoT-IoT in a multiclass scenario. The results indicate that the voting GWO ensemble model performed on reconnaissance achieved an accuracy of 99.91% and DR of 99.75%, precision of 97.08%, ROC of 98.80%, and FAR of 1.80.

5.5 Evaluation and comparison of current datasets suitability for IoT network

To determine the essential qualities of a valuable and realistic dataset for an IoT network, some of the current IDS datasets were evaluated in this part.

Table 12 Performance of the voting GWO ensemble model relative to the different attack types and benign in terms of accuracy, DR, precision, ROC, and FAR on the UNSW-NB15 dataset

Type of attack	Accuracy	DR	Precision	ROC	FAR
Benign	99.99	99.89	98.80	99.89	3.42
DoS	99.09	99.56	99.40	99.53	3.45
Backdoor	99.10	99.87	97.82	98.45	2.77
Worm	99.89	98.10	98.17	98.78	2.99
Shellcode	99.14	98.72	98.32	97.62	3.98
Probe	99.89	98.82	96.88	98.64	1.89
Exploits	99.90	98.67	99.08	99.80	3.79
Fuzzer	99.89	99.90	98.71	99.62	2.89
Analysis	99.78	99.10	98.83	99.84	2.89
Generic	99.69	99.59	98.67	99.89	1.99
Reconnaissance	**99.91**	**99.75**	**97.08**	**98.80**	**1.80**

Values of our proposed model are in bold

5.5.1 DARPA

For the goal of analyzing network security, this dataset was created. Due to problems with the fake injection of attacks as well as benign traffic, researchers chastised DARPA. DARPA covers tasks such as sending and receiving mail, surfing the web, sending and receiving files via FTP, using telnet to log into distant systems and carry out work, sending and receiving IRC messages, and remotely monitoring the router using SNMP. The aforementioned list comprises various types of attacks, including but not limited to denial of service (DOS), password guessing, buffer overflow, remote file transfer protocol (FTP), syn flood, network mapper (Nmap), and rootkit. Regrettably, the dataset under consideration does not accurately reflect network traffic in real-world scenarios in IoT and exhibits anomalies such as the lack of erroneous detections. Furthermore, it is no longer current enough to provide a comprehensive assessment of IDSs concerning contemporary network infrastructures and attack modalities. Furthermore, the absence of factual attack data records is evident [67].

5.5.2 KDD Cup 99

The dataset known as KDD Cup 1999 was derived by analyzing the tcpdump component of the 1998 DARPA dataset. However, it is important to note that the KDD Cup 1999 dataset is not immune to the same issues as its predecessor. The KDD99 dataset encompasses over twenty distinct types of attacks, including but not limited to neptune-dos, pod-dos, smurf-dos, buffer-overflow, rootkit, satan, and teardrop. The amalgamation of network traffic records of both normal and attack traffic within a simulated environment yields a dataset that contains a substantial amount of superfluous records, which are also tainted with data corruption. This, in turn, results in testing outcomes that are biased, as reported in reference [68]. NSL-KDD was developed as a means of addressing certain limitations of the KDD dataset [68], which had been identified in the previous research [67].

5.5.3 CDX

The utilization of network warfare competitions for the creation of contemporary labeled datasets is demonstrated by the CDX dataset. The dataset reveals that attackers have utilized widely recognized attack tools such as Nikto, Nessus, and WebScarab to conduct automated reconnaissance and attacks. Benign network traffic encompasses essential services such as web browsing, email communication, DNS queries, and other necessary functions. According to source [69], CDX has limitations in terms of traffic diversity and volume, although it can still serve as a tool for testing IDS alert rules.

5.5.4 Kyoto

The dataset in question has been generated through the utilization of honeypots, thereby precluding the possibility of manual labeling and anonymization. However, it is important to note that the dataset's scope is restricted to solely those attacks that were directed toward the honeypots. The current dataset offers ten additional features, including IDS detection, malware identification, and Ashula detection, compared to the previous datasets. These features are beneficial for conducting NIDS evaluation and analysis. As the attacks repeatedly simulate normal traffic, the resulting DNS and mail traffic information does not accurately reflect real-world normal traffic. Therefore, false positives are not present. The significance of false positives lies in their ability to reduce the frequency of alerts, as indicated by sources [70].

5.5.5 Twente

To generate the dataset, three distinct services, namely, OpenSSH, Apache web server, and Proftp utilizing auth/ident on port 113, were deployed to gather information from a honeypot network via netflow. Certain types of traffic, including auth/ident, ICMP, and irc traffic, may produce side effects that are neither entirely benign nor malicious. In addition, the dataset includes alert traffic that is both unidentified and lacking correlations. The labeled dataset under consideration is deemed more realistic; however, its deficiency in terms of the volume and variety of attacks is a conspicuous limitation as noted in reference [71].

5.5.6 ISCX2012

The authors have presented a valuable recommendation for producing realistic and useful IDS evaluation datasets through a dynamic approach. The dataset in question was generated using this approach. The methodology employed by the individuals involves a bifurcation into two distinct components, specifically denoted as the alpha and beta profiles. The alpha profile executes multiple stages of attack scenarios to filter the anomalous segment of the dataset. The beta profile, a benign traffic generator, produces authentic network traffic accompanied by ambient noise. Empirical data are utilized to construct profiles that simulate authentic traffic for various protocols such as HTTP, SMTP, SSH, IMAP, POP3, and FTP. The dataset produced by this methodology comprises network traces that include complete packet payloads and pertinent profiles. Nevertheless, it should be noted that the dataset in question does not pertain to novel network protocols, given that a significant proportion of contemporary network traffic, approximately 70%, is comprised of HTTPS, and no traces of HTTPS are present within the said dataset. Furthermore, the allocation of the simulated assaults is not grounded on empirical data [72]. Table 13 shows some popular realistic datasets for IoT networks.

As can be seen, only the proposed datasets used in this study meet all criteria. Tables 13 and 14 list and explain the dataset's flaws and strengths based on relevant documents and research, as well as their suitability for IoT networks. Some feature values are not presented as a result of inadequate documentation and a lack of metadata. Here, we evaluated the proposed model using two well-known datasets: UNSW-NB15 and BoT-IoT. In contrast with the datasets used in several existing models, which do not accurately reflect contemporary attacks on IoT networks and do not adhere to IoT protocol requirements, these chosen datasets are appropriate and realistic for IoT network traffic.

6 Discussion of findings

6.1 Comparison with the existing studies

In this section, we compared the performance of the proposed GWO ensemble model with the existing state-of-the-art models in Table 15. The majority of the state-of-the-art model concentrated on the NSLKDD and KDD Cup 99 datasets. These data are unrealistic intrusion detection datasets for the evaluation of IoT systems. They are unsuccessful in practical uses due to the dataset used to train and evaluate the underlying models being non-representative. On the other hand, several existing techniques address these issues but provide low accuracy, DR, precision, ROC, and FAR preventing them from being implemented in commercial systems. Also worthy of mentioning was that the existing state-of-the-art models paid no attention to feature dimensionality; this stage of dimensionality reduction is regarded as the most crucial stage. This phase is particularly time- and labor-intensive. This paper addressed the feature dimensionality phase by proposing a hybridized IG + PCA for dimensionality reduction and provides a novel GWO ensemble model for classification. Additionally, this proposed ensemble model was evaluated on realistic BoT-IoT and UNSW-NB15 datasets, which made it suitable for commercial and industrial applications. As shown in Fig. 3, the best state-of-the-art model provides 100% accuracy on the BoT-IoT data, while the ROC and F-measure were disregarded. On the comparable BoT-IoT data, the proposed innovative voting GWO ensemble model achieved an accuracy of 99.98%, DR of 99.97%, precision of 99.94%, ROC of 99.99%, and FAR of 1.30.

6.2 Computational compatibility across IoT devices

When designing a machine learning model for intrusion detection in IoT environments, it is important to consider the computational compatibility of the proposed model, especially given the heterogeneity in computational power among IoT devices. A model that works well on high-power devices might struggle or be impractical to implement on resource-constrained IoT devices. Imagine a scenario where our proposed model is deployed for real-time anomaly detection in a smart city environment, where various types of IoT devices are utilized, ranging from resource-constrained sensors to more powerful edge devices. In this scenario, the lightweight nature of our voting GWO ensemble model enables seamless integration across these devices. Resource-intensive tasks are offloaded to devise with higher computational power, while less resource-intensive tasks are managed by lower-powered devices. Our model's architecture is designed to dynamically adjust its computational requirements based on the available

Table 13 A comparative analysis of the datasets currently accessible for detecting attacks in IoT

		DARPA	LBNL	Kyoto	AWID	ISCX 2012	KDD'99	CDX	Twente
Traffic		No	Yes	No	No	No	No	No	Yes
Network		Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Label		Yes	No	Yes	Yes	Yes	Yes	No	Yes
Capture		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Interaction		Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Attacks	Brute-force	Yes	–	Yes	Yes	Yes	Yes	No	Yes
	Browser	Yes	–	Yes	Yes	Yes	Yes	No	No
	DoS	Yes	–	Yes	Yes	Yes	Yes	Yes	No
	DNS	No	–	Yes	No	No	No	Yes	No
	Backdoor	No	–	Yes	No	No	No	No	No
	Scan	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Others	Yes	–	Yes	Yes	Yes	Yes	–	Yes
Protocols	HTTP	No	No	Yes	No	No	No	No	No
	HTTP	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	FTP	Yes	No	Yes	Yes	Yes	Yes	Yes	No
	Email	Yes	No	Yes	Yes	Yes	Yes	Yes	No
	Ssh	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Heterogeneity		No	No	No	No	Yes	No	No	–
Anonymity		No	Yes	No	No	No	No	–	–
Metadata		Yes	No	Yes	Yes	Yes	Yes	No	Yes
Feature set		No	No	Yes	Yes	No	Yes	No	No

Table 14 Summary of representative (realistic) and non-representative (non-realistic) datasets for IoT

Dataset/authors	Traffic creation year	Public availability	Attack traffic	Normal traffic	Realistic network traffic for IoT
DARPA [73]	1999	Yes	Yes	Yes	No
LBNL [74]	2005	Yes	Yes	Yes	No
Kyoto 2006 + [70]	2011	Yes	Yes	Yes	No
NSL-KDD [68]	2009	Yes	Yes	Yes	No
SSENET-2011 [75]	2011	n.i.f	Yes	Yes	No
UNIBS [76]	2009	o.r	No	Yes	No
CDX [69]	2009	Yes	Yes	Yes	No
Twente [71]	2009	Yes	Yes	Yes	No
ISCX 2012 [72]	2012	Yes	Yes	Yes	No
Botnet [77]	2014	Yes	Yes	Yes	No
AWID [16]	2015	o.r	Yes	Yes	No
DDoS [78]	2016	Yes	Yes	Yes	No
CIDDS-001 [79]	2017	Yes	Yes	Yes	Yes
N-BaIoT [80]	2018	Yes	Yes	Yes	Yes
UNSW-NB15	2015	Yes	Yes	Yes	Yes
BoT-IoT	2019	Yes	Yes	Yes	Yes

o.r = on request and n.i.f = no information found

Table 15 Comparison with the state-of-the-art models

Authors	Methodology	Dataset used	Accuracy	DR	Precision	FAR
[44]	SVM, RF, DT, and LR	Capture live network	SVM = 98.06; RF = 99.17; DT = 98.34 LR = 97.50	X	x	x
[41]	CNN	System call graph	97	X	x	0.034
[20]	Bi-LSTM	UNSW-NB15	95	X	x	0
[42]	K-NN, Gaussian Naive Bayes, and random forest	Capture live network	K-NN = 94.44; Gaussian Naive Bayes = 77.78; RF = 88.8,	KNN = 100;GNB = 100; RF = 100	K-NN = 96; GNB = 86; RF = 92	x
[24]	GRU, LSTM, BLS, and Bi-LSTM	NSLKDD	84.14	X	x	x
[43]	GRU-MLP, BGRU-MLP, BLSTM + MLP, GRU, MLP, LSTM-MLP, LSTM	KDD Cup 99, NSLKDD	99.24	X	x	0.84
[39]	Single-layered ANN	N-BaIoT	99	X	x	x
[25]	SVM, J48, NB, MLP, NB tree, RF, RF tree, RNN-IDS, and ANN	NSLKDD	95.2	X	x	6.3
[40]	SMOTE and ANN	BoT-IoT	100	X	x	x
[27]	DJ, DF, DNN, LSTM, DBN, and GRU	NSLKDD, KDD Cup, and CICIDS	96.9	X	x	5.44
[81]	MTNN	ToN_IoT	87.79	90.69	77.95	x
[82]	CNN-CapSA	BoT-IoT	99.94	99.93	99.93	x
[83]	CNN-MGO	BoT-IoT	99.62	99.72	99.52	x
Our proposed model	Voting GWO ensemble model	BoT-IoT	**99.98**	**99.97**	**99.94**	**1.30**
Our proposed model	Voting GWO ensemble model	UNSW-NB15	**100**	**99.99**	**99.59**	**1.15**

resources, ensuring effective and efficient operation across the heterogeneous IoT landscape.

6.3 Transferable of the proposed research to real-world IoT applications

Our research is designed with a strong focus on practical applicability in real-world IoT environments. Here are key points highlighting the transferability of our research to real-world IoT applications:

a. **IoT-Centric Approach** We developed our intrusion detection system with a deep understanding of the unique characteristics and challenges of IoT networks. This approach ensures that our research is directly relevant to the specific requirements and constraints of IoT applications.
b. **Dataset Selection** We utilized datasets, such as BoT-IoT and UNSW-NB15, that are representative of real-world IoT network traffic and intrusions. This dataset selection ensures that our research is grounded in the realities of IoT security.
c. **Hybrid Approach** Our research combines feature extraction via principal component analysis (PCA), feature selection via IG, and GWO-based ensemble models.

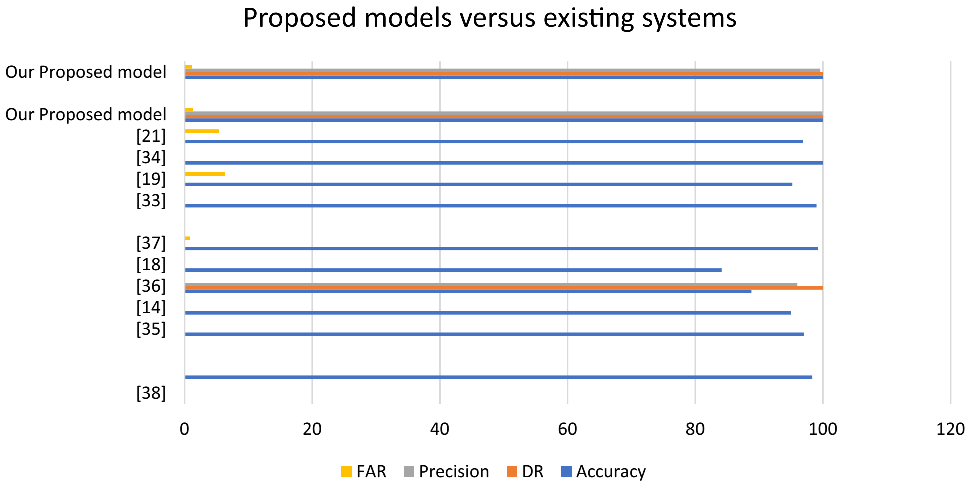

Fig. 3 Comparison of the proposed models with the existing models

This hybrid approach is designed to enhance the robustness and effectiveness of intrusion detection in real-world IoT scenarios.

d. **Generalization** We conducted experiments and evaluations on multiple datasets to ensure the generalizability of our proposed model to diverse IoT applications. Our research demonstrates the adaptability and transferability of our approach across various IoT contexts.
e. **Performance Metrics** We evaluated our intrusion detection system using well-established performance metrics, such as accuracy, DR, precision, and FAR. These metrics reflect the real-world effectiveness of our approach in identifying and mitigating security threats.
f. **Scalability** We addressed the scalability challenges often encountered in IoT environments, ensuring that our research can handle growing numbers of devices and data volumes while maintaining effectiveness.
g. **Practical Deployment Considerations** We discussed the practical considerations of deploying our intrusion detection system in real-world IoT applications, including the optimization of model parameters and the importance of network segmentation.
h. **Security Challenges** Our research explicitly addresses a range of security challenges and threats in IoT environments, making it directly applicable to scenarios where IoT security is a concern.

This research is built on a foundation that prioritizes real-world relevance and practicality. We have conducted experiments and evaluations that demonstrate the effectiveness and transferability of our IDS to various IoT applications. By addressing the unique challenges of IoT security and employing a hybrid approach that combines feature extraction, feature selection, and optimization techniques, we aim to provide a solution that can be readily applied in real-world IoT environments.

6.4 Threats to validity

The main danger to validity is random sampling, which makes it difficult to duplicate the exact experiment. To validate the suggested approach's reliability, the experiments were repeated on two separate realistic IoT sets of data with a substantial sample size. Finally, while the presented approach performed well in binary-class classification, it deserves additional investigation in the class of multiple classification issues.

7 Conclusion and future work

This paper proposes a novel voting GWO ensemble learning model for the detection of attacks in an IoT environment. The suggested system successfully detects various forms of IoT threats by leveraging the feature set retrieved from the IoT ecosystem. The strength of this paper concentrates on the voting GWO ensemble model, which is the first of its kind, the hybridization of IG + PCA for dimensionality reduction, and the leverage of realistic datasets that reflect real-time attacks in the IoT context. To construct a successful ensemble IDS for detecting IoT attacks, a collection of relevant features was selected. The experimental findings prove that the detection accuracy is increased in the voting GWO ensemble model in the suggested framework using the average probability technique. Our experimental results indicate that our proposed voting ensemble model outperforms other ML and DL approaches in terms of overall accuracy, attaining 100%, DR of 99.99%, precision of 99.59%, ROC of 99.40%, and FAR of 1.15 on the UNSW-NB15 compared to earlier studies. This indicates that our presented method will be extremely beneficial in designing contemporary IDS for the IoT environment. The suggested model will be extended in the future to incorporate multiple class classification problems. Also, the deep learning model to classify the additional forms of attacks may be considered in the future work.

Authors' contributions Authors contributed equally.

Funding Open access funding provided by Institute for Energy Technology.

Data availability The BoT-IoT is available at https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/bot_iot.php and N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive dataset for network intrusion detection systems (UNSW-NB15 network dataset)," 2015 Mil. Commun. Inf. Syst. Conf. MilCIS 2015—Proc., 2015, doi: https://doi.org/10.1109/MilCIS.2015.7348942. Also cited the same in the reference list [14].

Declarations

Conflict of interest Authors do not have any financial or non-financial interests that are directly or indirectly related to the work submitted for publication.

Ethical approval Authors comply with the highest level of ethical standards while preparing the manuscript.

References

1. Islam, N., et al.: Towards Machine learning based intrusion detection in IoT networks. Comput. Mater. Contin. **69**(2), 1801–1821 (2021). https://doi.org/10.32604/cmc.2021.018466
2. Rahman, M.A., Asyhari, A.T.: The emergence of Internet of things (IoT): connecting anything, anywhere. Computers **8**(2), 8–11 (2019). https://doi.org/10.3390/computers8020040
3. Lin, H., Hu, J., Wang, X., Alhamid, M.F., Piran, M.J.: Toward secure data fusion in industrial IoT using transfer learning. IEEE Trans. Ind. Inform. **17**(10), 7114–7122 (2021). https://doi.org/10.1109/TII.2020.3038780
4. Farsi, M., Daneshkhah, A., Hosseinian-Far, H., Jahankhani, A.: Digital Twin Technologies and Smart Cities. Springer, Berlin/Heidelberg, Germany (2020)
5. Zhao, K., Ge, L.: A survey on the Internet of things security. In: Proceedings—9th International Conference on Computational Intelligence and Security, CIS 2013, pp. 663–667 (2013). https://doi.org/10.1109/CIS.2013.145.
6. Yan, Z., Zhang, P., Vasilakos, A.V.: A survey on trust management for Internet of Things. J. Netw. Comput. Appl. **42**, 120–134 (2014). https://doi.org/10.1016/j.jnca.2014.01.014
7. Saheed, Y.K., Babatunde, A.O.: Genetic algorithm technique in program path coverage for improving software testing. Afr. J. Comput. ICT **7**(5), 151–158 (2014)
8. Kelton, A.P., Papa, J.P., Lisboa, C.O., Munoz, R., De, V.H.C.: Internet of Things: a survey on machine learning-based intrusion detection approaches. Comput. Netw. **151**, 147–157 (2019). https://doi.org/10.1016/j.comnet.2019.01.023
9. Saheed, Y.K., Misra, S., Chockalingam, S.: Autoencoder via DCNN and LSTM models for intrusion detection in industrial control systems of critical infrastructures. In: 2023 IEEE/ACM 4th Int. Work. Eng. Cybersecurity Crit. Syst. (EnCyCriS), Melbourne, Aust., pp. 9–16 (2023). https://doi.org/10.1109/EnCyCriS59249.2023.00006
10. Alharbi, S., Rodriguez, P., Maharaja, R., Iyer, P., Bose, N., Ye, Z.: FOCUS : a fog computing-based security system for the Internet of Things. (2018)
11. Pajouh, H.H., Javidan, R., Khayami, R., Dehghantanha, A., Choo, K.K.R.: A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks. IEEE Trans. Emerg. Top. Comput. **7**(2), 314–323 (2019). https://doi.org/10.1109/TETC.2016.2633228
12. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99 data set. no. Cisda, pp. 1–6 (2009).
13. Zhang, H., Wu, C.Q., Gao, S., Wang, Z., Xu, Y., Liu, Y.: An effective deep learning based scheme for network intrusion detection. In: 2018 24th Int. Conf. Pattern Recognit., pp. 682–687 (2018)
14. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Mil. Commun. Inf. Syst. Conf. MilCIS 2015—Proc. (2015). https://doi.org/10.1109/MilCIS.2015.7348942
15. Koroniotis, N., Moustafa, N., Sitnikova, E.: Towards Developing Network Forensic Mechanism for Botnet Activities in the IoT Based on Machine Learning Techniques. Springer International Publishing
16. Kolias, C., Kambourakis, G., Stavrou, A., Gritzalis, S.: Intrusion detection in 802. 11 Networks : Empirical Evaluation of Threats and a Public Dataset. no. c, pp. 1–24 (2015). https://doi.org/10.1109/COMST.2015.2402161
17. Saheed, Y.K., Usman, A.A., Sukat, F.D., Abdulrahman, M.: A novel hybrid autoencoder and modified particle swarm optimization feature selection for intrusion detection in the internet of things network. Front. Comput. Sci. **5**, 1–13 (2023). https://doi.org/10.3389/fcomp.2023.997159
18. Amin, S.O., Siddiqui, M.S., Hong, C.S., Choe, J.: A novel coding scheme to implement signature based IDS in IP based sensor networks. In: 2009 IFIP/IEEE Int. Symp. Integr. Netw. Manag. IM 2009, pp. 269–274 (2009).https://doi.org/10.1109/INMW.2009.5195973
19. Abubakar, A., Pranggono, B.: Machine learning based intrusion detection system for software defined networks. In: 2017 Seventh International Conference on Emerging Security Technologies, pp. 138–143 (2017)
20. Roy, B., Cheung, H.: A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network. In: 2018 28th Int. Telecommun. Networks Appl. Conf. ITNAC 2018, pp. 1–6 (2019).https://doi.org/10.1109/ATNAC.2018.8615294
21. Le, A., Loo, J., Luo, Y., Lasebae, A.: Specification-based IDS for securing RPL from topology attacks. IFIP Wirel. Days **1**(1), 4–6 (2011). https://doi.org/10.1109/WD.2011.6098218
22. Bertino, E.: Botnets and Internet of Things Security. Computer (Long. Beach. Calif)., pp. 76–79 (2017)
23. Almiani, M., AbuGhazleh, A., Al-Rahayfeh, A., Atiewi, S., Razaque, A.: Deep recurrent neural network for IoT intrusion detection system. Simul. Model. Pract. Theory **101**, 102031 (2020). https://doi.org/10.1016/j.simpat.2019.102031
24. Li, Z., Batta, P., Trajkovi, L.: Comparison of Machine Learning Algorithms for Detection of Network Intrusions. pp. 4248–4253 (2018). https://doi.org/10.1109/SMC.2018.00719

25. Ayyaz-ul-haq, Q., Larijani, H., Ahmad, J.: A heuristic intrusion detection system for Internet-of-Things (IoT). In: Arai, K., Bhatia, R., Kapoor, S. (eds.) Intelligent Computing. CompCom 2019. Advances in Intelligent Systems and Computing. Springer Cham, pp. 86–98 (2019)
26. Böhm, A., Jonsson, M., Uhlemann, E.: Performance comparison of a platooning application using the IEEE 802.11p MAC on the control channel and a centralized MAC on a service channel. Int. Conf. Wirel. Mob. Comput. Netw. Commun. 545–552 (2013).https://doi.org/10.1109/WiMOB.2013.6673411
27. Elmasry, W., Akbulut, A., Zaim, A.H.: Empirical study on multiclass classification-based network intrusion detection. Comput. Intell. **35**(4), 919–954 (2019). https://doi.org/10.1111/coin.12220
28. Jiang, K., Wang, W., Wang, A., Wu, H.: Network intrusion detection combined hybrid sampling with deep hierarchical network. IEEE Access **8**(3), 32464–32476 (2020). https://doi.org/10.1109/ACCESS.2020.2973730
29. Hasan, M., Islam, M., Zarif, I.I., Hashem, M.M.A.: Internet of things attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet Things **7**, 100059 (2019). https://doi.org/10.1016/j.iot.2019.100059
30. Cheng, Y., Xu, Y., Zhong, H., Liu, Y.: Leveraging Semi-supervised Hierarchical Stacking Temporal Convolutional Network for Anomaly Detection in IoT Communication, vol. 4662, no. c (2020). https://doi.org/10.1109/JIOT.2020.3000771.
31. Lee, T.H., Wen, C.H., Chang, L.H., Chiang, H.S., Hsieh, M.C.: A lightweight intrusion detection scheme based on energy consumption analysis in 6LowPAN. In: Advanced Technologies, Embedded and Multimedia for Human-centric Computing (2014). https://doi.org/10.1007/978-94-007-7262-5_137
32. Sahu, N.K., Mukherjee, I.: Machine learning based anomaly detection for IoT network:(Anomaly detection in IoT network). In: 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), no. Icoei, pp. 787–794 (2020). https://doi.org/10.1109/ICOEI48184.2020.9142921
33. Chen, J., Chen, C.: Design of complex event-processing IDS in internet of things. In: Proc. - 2014 6th Int. Conf. Meas. Technol. Mechatronics Autom. ICMTMA 2014, pp. 226–229 (2014). https://doi.org/10.1109/ICMTMA.2014.57
34. Midi, D., Rullo, A., Mudgerikar, A., Bertino, E.: Kalis—a system for knowledge-driven adaptable intrusion detection for the Internet of Things. In: Proc. - Int. Conf. Distrib. Comput. Syst., pp. 656–666 (2017). https://doi.org/10.1109/ICDCS.2017.104
35. Karunkumar, D., Himansu, R., Behera, S., Nayak, J.: Deep neural network based anomaly detection in Internet of Things network traffic tracking for the applications of future smart cities. no. July, pp. 1–26 (2020). https://doi.org/10.1002/ett.4121
36. Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., Lloret, J.: Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot. Sensors (Switzerland) (2017). https://doi.org/10.3390/s17091967
37. Guller, M.: Big data analytics with Spark: A practitioner's guide to using Spark for large scale data analysis. Apress (2015)
38. Joshi, H.P., Bennison, M., Dutta, R.: Collaborative botnet detection with partial communication graph information. In: 2017 IEEE 38th Sarnoff Symp. (2017). https://doi.org/10.1109/SARNOF.2017.8080397
39. Soe, Y.N., Feng, Y., Santosa, P.I., Hartanto, R., Sakurai, K.: A sequential scheme for detecting cyber attacks in IoT environment. In: Proc. - IEEE 17th Int. Conf. Dependable, Auton. Secur. Comput. IEEE 17th Int. Conf. Pervasive Intell. Comput. IEEE 5th Int. Conf. Cloud Big Data Comput. 4th Cyber Sci., vol. 324, pp. 238–244 (2019). https://doi.org/10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00051
40. Soe, Y.N., Santosa, P.I., Hartanto, R.: DDoS attack detection based on simple ANN with SMOTE for IoT environment. In: Proc. 2019 4th Int. Conf. Informatics Comput. ICIC 2019, pp. 0–4 (2019). https://doi.org/10.1109/ICIC47613.2019.8985853
41. Le, H.V., Ngo, Q.D., Le, V.H.: Iot Botnet detection using system call graphs and one-class CNN classification. Int. J. Innov. Technol. Explor. Eng. **8**(10) (2019).
42. Kumar, A., Lim, T.J.: EDIMA: early detection of IoT malware network activity using machine learning techniques. In: IEEE 5th World Forum Internet Things, WF-IoT 2019—Conf. Proc., pp. 289–294 (2019). https://doi.org/10.1109/WF-IoT.2019.8767194
43. Xu, C., Member, S., Shen, J., Du, X.I.N., Zhang, F.A.N.: An intrusion detection system using a deep neural network with gated recurrent units. IEEE Access **PP**(c), 1 (2018). https://doi.org/10.1109/ACCESS.2018.2867564
44. Chaudhary, P., Gupta, B.B.: DDoS detection framework in resource constrained internet of things domain. In: 2019 IEEE 8th Glob. Conf. Consum. Electron. GCCE 2019, pp. 675–678 (2019).https://doi.org/10.1109/GCCE46687.2019.9015465
45. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., Alazab, A.: A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. Electron (2019). https://doi.org/10.3390/electronics8111210
46. Alazab, A., Abawajy, J., Hobbs, M., Layton, R.: Crime Toolkits : The Productisation of Cybercrime (2013). https://doi.org/10.1109/TrustCom.2013.273
47. Singh, J., Pasquier, T., Bacon, J., Ko, H., Eyers, D.: Twenty security considerations for cloud-supported Internet of Things. vol. 4662, no. c, pp. 1–16 (2015). https://doi.org/10.1109/JIOT.2015.2460333
48. Adeyiola, A.Q., Saheed, Y.K., Misra, S., Chockalingam, S.: Metaheuristic firefly and C5 . 0 algorithms based intrusion detection for critical infrastructures. In: 2023 3rd International Conference on Applied Artificial Intelligence (ICAPAI), pp. 1–7 (2023). https://doi.org/10.1109/ICAPAI58366.2023.10193917
49. Kolias, C., Kambourakis, G., Stavrou, A., Voas, J.: DDoS in the IoT: Mirai and other botnets. Computer (Long Beach Calif.) **50**(7), 80–84 (2017). https://doi.org/10.1109/MC.2017.201
50. Abomhara, M., Køien, G.M.: Cyber security and the internet of things : vulnerabilities , threats , intruders.**4**, 65–88 (2015). https://doi.org/10.13052/jcsm2245-1439.414
51. Koroniotis, N., Moustafa, N., Sitnikova, E., Turnbull, B.: Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. Futur. Gener. Comput. Syst. **100**, 779–796 (2019). https://doi.org/10.1016/j.future.2019.05.041
52. Mansfield-devine, S., Security, N.: DDoS goes mainstream: attacks could make this threat an organisation ' s biggest nightmare. Netw. Secur. **2016**(11), 7–13 (2016). https://doi.org/10.1016/S1353-4858(16)30104-0
53. Greenberg, A.: Hackers remotely kill a jeep on the highway—with me in it. Wired, **7**(21) (2015)
54. Saheed, Y.K.: Data analytics for intrusion detection system based on recurrent neural network and supervised machine learning methods. In: Recurrent Neural Networks, pp. 167–179. CRC Press Taylor & Francis Group (2022)
55. Jain, S., Shukla, S., Wadhvani, R.: Dynamic selection of normalization techniques using data complexity measures. Expert Syst. Appl. **106**, 252–262 (2018). https://doi.org/10.1016/j.eswa.2018.04.008
56. Georganos, S., Lennert, M., Grippa, T., Vanhuysse, S., Johnson, B., Wolff, E.: Normalization in unsupervised segmentation parameter optimization: a solution based on local regression trend analysis. Remote Sens. (2018). https://doi.org/10.3390/rs10020222
57. Saheed, Y.K.: Performance improvement of intrusion detection system for detecting attacks on internet of things and edge of things. In: Misra, S., Kumar, T.A., Piuri, V., Garg, L. (eds.) Artificial Intelligence for Cloud and Edge Computing. Internet of Things

(Technology, Communications and Computing). Springer, Cham (2022)

58. Gray, R.M.: Entropy and Information Theory. Springer Science & Business Media (2011)
59. Adi, E., Baig, Z., Hingston, P.: Stealthy Denial of Service (DoS) attack modelling and detection for HTTP/2 services. J. Netw. Comput. Appl. **91**, 1–13 (2017). https://doi.org/10.1016/j.jnca.2017.04.015
60. Saheed, Y.K.: Machine learning-based blockchain technology for protection and privacy against intrusion attacks in intelligent transportation systems. In: Machine Learning, Blockchain Technologies and Big Data Analytics for IoTs: Methods, Technologies and Applications, p. 16 (2022)
61. ZorarpacI, E., Özel, S.A.: A hybrid approach of differential evolution and artificial bee colony for feature selection. Expert Syst. Appl. **62**, 91–103 (2016). https://doi.org/10.1016/j.eswa.2016.06.004
62. Jimoh, R.G., Ridwan, M.Y., Yusuf, O.O., Saheed, Y.K.: Application of dimensionality reduction on classification of colon cancer using Ica and K-Nn algorithm. Anale. Ser. Informatică, vol. 6, no. 10, pp. 55–59, 2018, [Online]. Available: http://anale-informatica.tibiscus.ro/download/lucrari/16-1-06-Olatunde.pdf.
63. Seni, G., Elder, J.F.: Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions, vol. 2, no. 1 (2010)
64. Hung, C., Chen, J.H.: A selective ensemble based on expected probabilities for bankruptcy prediction. Expert Syst. Appl. **36**(3 PART 1), 5297–5303 (2009). https://doi.org/10.1016/j.eswa.2008.06.068
65. Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., Abuzneid, A.: Features dimensionality reduction approaches for machine learning based network intrusion detection. Electron (2019). https://doi.org/10.3390/electronics8030322
66. Elhag, S., Fernández, A., Bawakid, A., Alshomrani, S., Herrera, F.: On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on Intrusion Detection Systems. Expert Syst. Appl. **42**(1), 193–202 (2015). https://doi.org/10.1016/j.eswa.2014.08.002
67. Mchugh, J.: Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM Trans. Inf. Syst. Secur. **3**(4), 262–294 (2000). https://doi.org/10.1145/382912.382923
68. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99 data set in computational intelligence for security and defense applications. Comput. Intell. Secur. Def. Appl., no. Cisda, 1–6 (2009)
69. Sangster, B. et al.: Toward instrumenting network warfare competitions to generate labeled datasets. In: 2nd Work. Cyber Secur. Exp. Test, CSET 2009 (2009)
70. Sato, M., Yamaki, H., Takakura, H.: Unknown attacks detection using feature extraction from anomaly-based IDS alerts. In: Proc.—2012 IEEE/IPSJ 12th Int. Symp. Appl. Internet, SAINT 2012, pp. 273–277 (2012). https://doi.org/10.1109/SAINT.2012.51
71. Sperotto, A., Sadre, R., Van Vliet, F., Pras, A.: A labeled data set for flow-based intrusion detection. In: IP Operations and Management: 9th IEEE International Workshop, IPOM, pp. 39–50 (2009). https://doi.org/10.1007/978-3-642-04968-2_4
72. Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput. Secur. **31**(3), 357–374 (2012). https://doi.org/10.1016/j.cose.2011.12.012
73. Lippmann, R.P. et al.: Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation. In: Proc. - DARPA Inf. Surviv. Conf. Expo. DISCEX 2000, vol. 2, pp. 12–26 (2000). https://doi.org/10.1109/DISCEX.2000.821506
74. Ruoming, P., Mark, A., Mike, B., Jason, L., Vern, P., Brian, T.: A first look at modern enterprise traffic. In: p. Proceedings of the 5th ACM SIGCOMM conference on I (2005)
75. Vasudevan, A.R., Harshini, E., Selvakumar, S.: SSENet-2011: a network intrusion detection system dataset and its comparison with KDD CUP 99 dataset. Asian Himalayas Int. Conf. Internet (2011). https://doi.org/10.1109/AHICI.2011.6113948
76. Gringoli, F., Salgarelli, L., Cascarano, N., Risso, F., Claffy, K.C., Rodriguez, P.: GT: picking up the truth from the ground in traffic classification. ACM SIGCOMM Comput. Commun. Rev. **39**(5), 12–18 (2009)
77. Beigi, E.B., Jazi, H.H., Stakhanova, N., Ghorbani, A.A.: Towards effective feature selection in machine learning-based botnet detection approaches. In: 2014 IEEE Conf. Commun. Netw. Secur. CNS 2014, pp. 247–255 (2014).https://doi.org/10.1109/CNS.2014.6997492
78. Alkasassbeh, M., Al-Naymat, G., B.A, A., Almseidin, M.: Detecting distributed denial of service attacks using data mining techniques. Int. J. Adv. Comput. Sci. Appl. **7**(1), 436–445 (2016). https://doi.org/10.14569/ijacsa.2016.070159
79. Sharafaldin, I., Gharib, A., Lashkari, A.H., Ghorbani, A.A.: Towards a reliable intrusion detection benchmark dataset. Softw. Netw. **2017**(1), 177–200 (2017). https://doi.org/10.13052/jsn2445-9739.2017.009
80. Meidan, Y., et al.: N-BaIoT-Network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Comput. **17**(3), 12–22 (2018). https://doi.org/10.1109/MPRV.2018.03367731
81. Ahmed, S.W., Kientz, F., Kashef, R.: A modified transformer neural network (MTNN) for robust intrusion detection in IoT networks. In: 2023 Int. Telecommun. Conf. ITC-Egypt 2023, pp. 663–668 (2023).https://doi.org/10.1109/ITC-Egypt58155.2023.10206134
82. Abd Elaziz, M., Al-qaness, M.A.A., Dahou, A., Ibrahim, R.A., El-Latif, A.A.A.: Intrusion detection approach for cloud and IoT environments using deep learning and Capuchin Search Algorithm. Adv. Eng. Softw. **176**(December 2022), 103402 (2023). https://doi.org/10.1016/j.advengsoft.2022.103402
83. Fatani, A., et al.: Enhancing intrusion detection systems for IoT and cloud environments using a growth optimizer algorithm and conventional neural networks. Sensors **23**(9), 1–14 (2023). https://doi.org/10.3390/s23094430

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