International Journal of Information Security
https://doi.org/10.1007/s10207-024-00844-w

REGULAR CONTRIBUTION

Adversarial attack detection framework based on optimized weighted conditional stepwise adversarial network

Kousik Barik[1] · Sanjay Misra[2,3] · Luis Fernandez-Sanz[1]

Accepted: 20 March 2024

Abstract

Artificial Intelligence (AI)-based IDS systems are susceptible to adversarial attacks and face challenges such as complex evaluation methods, elevated false positive rates, absence of effective validation, and time-intensive processes. This study proposes a WCSAN-PSO framework to detect adversarial attacks in IDS based on a weighted conditional stepwise adversarial network (WCSAN) with a particle swarm optimization (PSO) algorithm and SVC (support vector classifier) for classification. The Principal component analysis (PCA) and the least absolute shrinkage and selection operator (LASSO) are used for feature selection and extraction. The PSO algorithm optimizes the parameters of the generator and discriminator in WCSAN to improve the adversarial training of IDS. The study presented three distinct scenarios with quantitative evaluation, and the proposed framework is evaluated with adversarial training in balanced and imbalanced data. Compared with existing studies, the proposed framework accomplished an accuracy of 99.36% in normal and 98.55% in malicious traffic in adversarial attacks. This study presents a comprehensive overview for researchers interested in adversarial attacks and their significance in computer security.

Keywords Intrusion detection systems · Adversarial attack · Security · Weighted conditional stepwise adversarial network (WCSAN) · Particle swarm optimization (PSO)

1 Introduction

With the continuous evolution of the internet and widespread usage, the number of network users has increased exponentially. The quantity of internet-connected devices in finance and e-commerce is growing, and they are evolving targets of attacks, posing significant risk and driving significant damage. Hackers are individuals who pose a threat to information systems. Hackers use network and device flaws to conduct destructive operations, costing businesses and customers financially. The primary objective of intrusion detection is to differentiate between normal and abnormal information breaches [1]. Network security aims to protect systems, networks, programs, data, and user accounts from unauthorized access, modification, or disruption [2]. A single intrusion can instantly render the system unavailable and impact the organization. IDS can be categorized into host-based (HIDS) and network-based (NIDS) approaches. HIDS can observe and evaluate the network traffic passing through its interfaces [3]. NIDS analyzes the network traffic across the total network to detect known attacks [4]. DDoS attacks on a large scale, spoofing, Man-in-the-middle attacks, etc., can be used to conduct these malicious actions [5]. Practical procedures for detecting and defending against attacks and continuous monitoring are needed. Detecting different types of new attacks is challenging [6].

Detecting intrusions is a critical task in cyber security. Machine learning and deep learning techniques detect abnormal behavior in IDS [7]. Both linear and nonlinear ML/DL classifiers are exposed to adversarial attacks designed to mislead the classification model. IDSs are vulnerable to

✉ Sanjay Misra
sanjay.misra@ife.no

Kousik Barik
kousik.kousik@edu.uah.es

Luis Fernandez-Sanz
luis.fernandez.sanz@uah.es

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

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

[3] Department of Computer Science and Communication, Østfold University College, Halden, Norway

Published online: 12 April 2024

attacks, though they have been widely used commercially [8]. Conventional machine learning methods and strategies are commonly employed for their high precision in detecting attacks and low rate of false alarms. However, they have been criticized for their failure to identify emerging threats. Conventional machine learning methods need to improve in detecting complex and novel attacks. Typical machine learning models cannot detect slight modifications because they cannot generalize information and identify new attacks [9]. Adversarial attacks are a significant threat to modern AI applications, especially with the increasing use of data-oriented techniques and internet-based applications in critical areas such as biometric authentication and cybersecurity [10]. Adversarial attacks pose a risk when utilized to alter the categorization [11]. A minor disturbance can enable malware to bypass detection. An effective adversarial attack on an IDS can bypass detection, posing a direct threat to machine-learning-based intrusion detection systems [12].

An adversarial example is input to IDS that an attacker has deliberately designed to cause the model to make misclassifications. Different adversarial attacks on IDS, such as poisoning, model extraction, evasion, and inference attacks, have been observed [13]. During poisoning strikes, the attacker introduces false data points entering the practice facility to manipulate the trained classifier into making predictions favoring the adversary. In adversarial attacks, the attacker injects specially prepared data points into the testing set. In model extraction attacks, the attacker pilfers trained IDS; in inference attacks, the attacker infers sensitive data from the training set [14]. Figure 1 illustrates the different adversarial attacks on IDS. The attacker injects malicious code into the training data and attempts to gain sensitive information from the training data. The attacker attempts to steal the information from the trained IDS. The IDS predicts inaccurate classification.

From the attacker's perspective, adversarial attacks can possess changes to input data to enhance misclassification, thereby bypassing the IDS [15]. Consequently, malicious network packets are frequently incorrectly labeled benign due to the intrusion classifier's decision limits requiring clarification. Therefore, these disruptions restrict the performance of detectors based on ML and DL [16]. Defending IDS against adversarial attacks should be further assessed. Many investigations have been carried out to detect adversarial attacks, but the detection of adversarial attacks against IDS has yet to be explored more [17–19]. The motivation of this study is to design an adversarial attack mitigation strategy and analysis of IDS. The major contributions of the proposed work are as follows.

1. To propose a WCSAN-PSO framework for intrusion detection in adversarial attacks.
2. To analyze the framework by incorporating feature extraction (principal component analysis) and feature extraction (least absolute shrinkage and selection operator)
3. To employ labeling attacks to identify known attacks using a signature. The prediction can be made at the initial level, reducing bandwidth, computing resources, and attack detection efficiency in IDS.
4. To generate adversarial samples based on the IDS traffic characteristics. The IDS are trained with training datasets, including real and attack network traffic samples obtained from WCSAN.
5. To develop and evaluate the framework using an optimized PSO algorithm and SVC classifier with the CIC-IDS2017 dataset, which contains different types of contemporary attacks in IDS.

The remaining paper is formulated as follows. The theoretical background, related works, and problem statements are discussed in Sect. 2. Section 3 illustrates the proposed framework. Section 4 describes the performance analysis and comparative study. The discussion is presented in Sect. 5. The study limitations and future work are demonstrated in Sect. 6. Finally, the paper is concluded in Sect. 7.

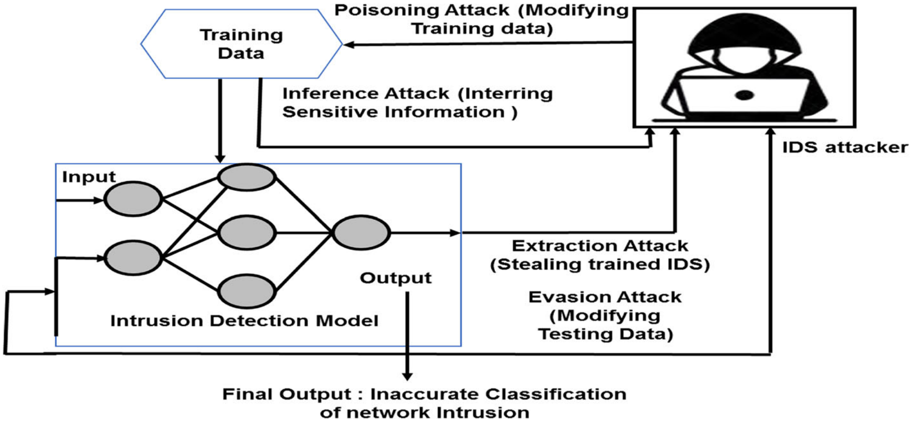
2 Literature review

This section outlines the background of the study, including the theoretical concepts of IDS and adversarial attacks. The existing studies on IDS and adversarial attack detection are highlighted with challenges. The problem statement is presented.

2.1 Theoretical background

Researchers have aimed to design more sophisticated algorithms since introducing artificial intelligence. Artificial intelligence has extended, and deep learning has emerged as a high-performing new approach [20]. This development was significant in machine learning due to the significantly superior performance results compared to those achieved using conventional methods [21]. DL has profited from utilizing large datasets during training in recent years and has seen hardware enhancements, particularly in GPUs [22]. Deep learning has simplified problem-solving by automating the fundamental stage of machine learning known as feature extraction. Convolution is the process of integrating two signals to create a new signal. The first signal is the data, while the subsequent is the filter [23]. DL's flexibility is another notable aspect. Deep learning requires extensive training with a larger number of samples. Due to advancements in

Fig. 1 Adversarial attacks in IDS


Training Data
Poisoning Attack (Modifying Training data)
Inference Attack (Interring Sensitive Information)
IDS attacker
Input
Output
Intrusion Detection Model
Extraction Attack (Stealing trained IDS)
Evasion Attack (Modifying Testing Data)
Final Output : Inaccurate Classification of network Intrusion

multicore PCs and GPUs, deep learning has accelerated significantly by dramatically reducing training time with large datasets [24].

Security measures like authentication and access control have been created to accomplish the goal of computer security, which is to prevent unauthorized individuals from accessing and altering information. These prevention mechanisms function as the primary line of defense [25]. The Internet's benefits, such as easy access to vast information, also present the greatest risk to information security. An intrusion detection system (IDS) is a secondary defense measure [26]. An IDS is a combination of two phrases: intrusion and detection systems. Intrusion is the unauthorized access to computer or network information intending to compromise its CIA triad, i.e., integrity, confidentiality, or availability. A detection system is a security measure designed to identify illegal action. IDS is a security tool that monitors the CIA triad [27].

From the perspective of deployment-based IDS, it can be further categorized as Host-based IDS (HIDS) or Network-based IDS (NIDS) [28]. HIDS is installed on a single information host. The task is to monitor all activities on a single host, scanning for security policy violations and suspicious activities [29]. The primary disadvantage is the need to deploy it on all hosts that need intrusion protection, leading to additional processing overhead for each node and ultimately reducing the performance of the IDS. On the contrary, NIDS is installed on the network to safeguard all devices and the entire network from intrusions. The NIDS continually observes network traffic to detect security breaches and violations [30]. IDS can be grouped into two categories depending on the model used: signature-based IDS and anomaly-based IDS. Signature-based IDS stores pre-defined attack signatures in a database and monitors the network for any matches against these signatures. Anomaly-based IDS monitors network traffic and compares it to the standard usage patterns of the network [31]. Adversarial attacks create samples using a natural sample and the victim instance. Generative adversarial networks (GANs) are a potent category of generative models that employ two networks trained concurrently in a zero-sum game, with one network dedicated to data generation and the other to discrimination [32]. A GAN consists of two elements: a generator and a discriminator. The generator simulates the data distribution to create adversarial examples and deceive the discriminator, which attempts to differentiate between fake and real examples [33]. Adversarial attacks pose evolving difficulties, requiring ML models to enhance their protection and resilience. Many studies in cybersecurity and IDS have explored the risk of adversarial examples and proposed potential strategies to counter them [34].

2.2 Related works

Machine learning is a subset of artificial intelligence focusing on algorithms and scientific models computer systems utilize. ML involves constructing a mathematical model using training data to make predictions or decisions [35]. ML techniques are commonly utilized in IDS research because they classify new data based on patterns from historical data. With the advancement of deep learning methods, they began to be extensively utilized in intrusion detection system research [36]. Ferdowsi et al. [37] proposed a study on distributed adversarial networks on IDS systems, and 2365 samples were considered. The authors reached both higher 20% accuracy and 25% precision than standalone IDS. Caminero et al. [38] conducted a study introducing adversarial reinforcement learning for IDS and developed a new technique that integrates the environment's behavior into the learning process. The Random Forest, Random Tree, MLP, J48, and Naive Bayes classifiers are evaluated for performance analysis. The Radom Tree classifier achieved an accuracy of 96.23%, precision of 95.90%, f1-score of 94.80%, and recall of 95.80%. Qiu et al. [39] presented a study using adversarial attacks on network intrusion detection systems. The authors employed two methods, i.e., reproduction of the black box model with training data and feature extraction of packets. The FGSM technique was used for iteration and achieved a 94.31% attack success rate. Alhajjar et al. [40] presented a study using particle swarm optimization, genetic algorithm, and generative adversarial networks to detect attacks in NIDS. The proposed method is applied to two datasets, i.e., NSL-KDD and UNSW-NB15, and achieves an accuracy of 98.06% using the PSO algorithm. The study [41] explored targeting supervised techniques by creating adversarial instances utilizing the Jacobian-based Saliency Map attack and analyzing classification behaviors in IDS. The authors used two methods, i.e., RF and J48, and achieved a precision of 94%, recall of 94%, and f1-score of 94% using RF.

Chatzoglou et al. [42] presented a study on attack detection in the IEEE 802.11 network using the AWID3 dataset. It significantly enhances and expands examining evidence of an extensive array of attacks launched within the IEEE 802.1 X extensible authentication protocol frameworks. Smiliotopoulos et al. [43] presented a comprehensive approach to identifying lateral movement, which is the tactic of an advanced persistent threat group using supervised machine learning methods. The authors achieved an f-score of 99.41% and an AUC of 0.998 while considering an unbalanced dataset. Yu et al. [44] proposed an intrusion detection system based on multi-scale convolutional neural networks for network security communication. The proposed deep learning based on the MSCNN model is tested on five different types of attacks and achieves an enhanced accuracy of 4.27% reached to others. Chatzoglou et al. [45] studied machine learning-driven IDS to identify Wi-Fi threats behind schedule. The authors used the 802.11 security-based AWID dataset. The study achieved an f1-score of 99.55% and 97.55% using shallow and deep learning techniques repetitively without optimization. Khan et al. [46] explored an in-depth study of IDS based on deep learning methods with various IDS. The public IDS datasets are comprehensively analyzed and discussed in the research. The study demonstrated various performance criteria used objectively to assess deep learning approaches for IDS. The authors further highlighted the challenges and solutions while implementing IDS. Chatzoglou et al. [47] studied detecting application layer attacks on Wi-Fi networks and used the AWID3 dataset. The study considered 802.11 and non-802.11 network protocol features. The different classifiers are DT, LightGBM, and Bagging. MLP and AE were used to evaluate the performance and presented an attack detection performance of 96.7%. Usmani et al. [48] examined distributed DOS and detected DOS. It's difficult to stop these attacks early. The authors used deep learning based on the long short-term memory technique and decision tree to classify ARP Spoofing attacks. They presented an accuracy of 99% and 100% utilizing LSTM and DT, respectively. Ramachandran et al. [49] designed an active method for detecting ARP spoofing. It can accurately identify the true correspondence between MAC and IP addresses during an attack.

Pawlicki et al. [50] proposed an artificial neural network using an IDS to identify adversarial attacks. The false positive rate of adversarial evasion attack prediction based on ANN is higher. Taheri et al. [51] presented a study on malware detection on adversarial mobile networks. They used a two-stage, real-time adversarial deep learning approach. The authors presented an accuracy of 96.03% using the C4N technique in normal conditions, but with adversarial attacks, the accuracy was reduced to 40%. Yang et al. [52] presented network-based intrusion detection with adversarial autoencoders with DNN (SAVAER-DNN). The NSL-KDD and UNSW-NB15 are used to evaluate the model. The proposed model yielded an accuracy of 93.01% and an f-score of 93.54%. Quresh et al. [53] proposed a study on adversarial attack detection on IDS using the Jacobian Saliency Map Attacks technique. They proposed an RNN-ADV model based on a radon neural network and used the NSL-KDD dataset for training. The proposed model achieved an accuracy of 95.6% in a normal scenario, but in the adversarial scenario, the accuracy falls by 47.58%.

Debicha et al. [54] presented a study using multi-adversarial networks against NIDS. The authors developed and executed transfer learning-based adversarial detectors, individually obtaining a subset of the data handed via the IDS. The proposed model is evaluated using the CIS-IDS2017 and NSL-KDD datasets. The proposed DNN-IDS model yielded an attack detection rate of 71.69% and 74.05% using the NSL-KDD and CIS-IDS2017 datasets in the adversarial scenarios. Roshan et al. [55] presented a study generating adversarial methods using the Fast Gradient Sign Method, Jacobian Saliency Map Attack, Carlini & Wagner, and Projected Gradient Descent in NIDS. The CIS-IDS2017 dataset was used. The authors demonstrated an accuracy of 98.7% using the FSGM method in adversarial conditions. Alotaibi et al. [56] presented a study on the sustainability of deep learning-based techniques on IDS using adversarial attacks. The study proposed a CNN-based IDS model, and the CIS-IDS2017 dataset has been used. Different techniques are used to generate adversarial attacks. The proposed model yielded an accuracy of 89.40% in adversarial attack detection. Paya et al. [57] proposed a method of detecting adversarial attacks against machine learning in IDS. The proposed model uses various classifiers to determine intrusions and utilizes Multi-Armed Bandits with Thompson sampling to choose the optimal classifier for each input dynamically. The authors demonstrated an accuracy of 93.04%. The existing IDS attack detection studies are summarized in Table 1.

Based on the review of existing studies, some research specifically concentrates on identifying DDoS attacks. Other

Table 1 Summary of existing IDS attack detection studies

References no	Method	Dataset	Outcome	Gaps
Ferdowsi et al. [37]	GAN-based IDS, ANN	IoTD	Accuracy 89%,	Focused ANN to identify attacks and time-consuming process
Caminero et al. [38]	RF, RT, MLP, J48, and NB	NSL-KDD	Accuracy 80%, F1 score 79%	Focused on detecting IDS attacks but not considered optimization techniques
Qiu et al. [39]	DNN, FGSM	Mirai	Attack success rate 94.31%	Focused attack detection is DL-based NIDS but has not been considered an optimization method
Alhajjar et al. [40]	DT, PSO, GA, LDA, KNN	NSL-KDD, UNSW-NB15	Accuracy 98.06%	Parameter optimization is not considered
Anthi et al. [41]	J48, RF, JSMA	Power plant	F-score 80% in adversarial conditions	Feature selection and optimization are not considered
Pawlicki et al. [50]	ANN, RF, AdaBoost, SVM	CIS-IDS2017,	Precision 11%, recall 99%, f1-score 20%	Not focused on feature extraction and optimization
Taheri et al. [51]	Robust-NN, C4N	Drebin, Contagio, Genome	F1-score 69.29%, Recall 69.73%, Precision 68.86%	Not focused on data preprocessing and feature selection
Yang et al. [52]	SAVAER-DNN	NSL-KDD, UNSW-NB15	Accuracy 93.01%, F-score 93.54%	Two datasets are combined, but preprocessing, feature extraction and optimization are not considered
Qureshi et al. [53]	RNN-ADV, JSMA,MLP	NSL-KDD	Accuracy in normal conditions of 63.41%. Accuracy 71.38%, Precision 47.23% in adversarial attack using RNN-ADV IDS model	Not focused on the implementation model
Debicha et al. [54]	DNN-IDS, FSGM,PGD,CQ,DF,	NSL-KDD, CIC-IDS2017	Accuracy 74.05% using the DNN-IDS model in adversarial conditions	The study did not focus on preprocessing, feature extraction, and optimization techniques
Roshan et al. [55]	FGSM, JSMA, PGD, C&W,	CIC-IDS2017	Accuracy 98.7% in adversarial attack scenario using FGSM method	The study did not focus on feature extraction or explore the impact of balanced and unbalanced data in adversarial scenarios
Alotaibi et al. [56]	CNN, FGSM,BIM,PGD, Auto-PGD	CIC-IDS2017	Accuracy 89.40% using the CNN-IDS model in adversarial conditions	Not focused on processing, bias in the dataset
Paya et al. [57]	Apollon-IDS, MLP, RF,LR,NB	CIC-IDS2017, CSE-CIC-IDS-2018, CIC-DDoS-2019	Accuracy 93.04%, F1-score 88.35%, using Apollon-IDS	The model takes more training time and computation resources

Table 1 (continued)

References no	Method	Dataset	Outcome	Gaps
Proposed WCSAN-PSO framework	PCA, LASCO,WCSAN, PSO, SVC	CIC-IDS2017	Improved accuracy, precision, and AUC value in adversarial attack detection	The study uses the PCA, LASCO, WCSAN, PSO, and SVC techniques to design the WCSAN-PSO framework

significant attacks are not considered. Likewise, a straightforward ANN was deployed in one case, processing without feature selection, and no optimization techniques were applied. Similarly, a fundamental artificial neural network was used in one case, operating without feature selection and without applying any optimization techniques. Also, in a few studies, the proposed IDS model with machine and deep learning performed well in normal scenarios. However, the accuracy and other evaluation parameters are decreased in an adversarial attack scenario. Most existing approaches demonstrated in this study for detecting machine and deep learning are the main targets of adversarial attacks. Still, they are complex evaluation processes with high false positive rates, no effective validations, time-consuming processes, require higher bandwidth and high computing resources for processing, challenge in maintenance, and larger memory consumption. Further, ML and DL-based IDS are vulnerable to adversarial attacks. Unknown adversarial attacks can still bypass machine and deep learning-based IDS because they are trained on known adversarial attacks, which is a shortfall in the adversarial training process.

To overcome the existing research gap, the proposed framework is designed with a unique attack leveling pattern while maintaining and updating the signature database so that in case any known attack is detected. The prediction can be made at the initial level, reducing bandwidth, computing resources, and attack detection efficiency in IDS. The proposed framework utilizes a WCSAN to construct a corrected training data set with correct labels. PCA has adopted feature extraction and LASSO for feature selection. The PSO algorithm optimizes the parameters of the generator and discriminator in WCSAN to enhance the adversarial training of IDS.

2.3 Problem statement

IDS is used to automate a variety of cybersecurity responsibilities. Most of these techniques employ supervised learning algorithms, which rely on data from the specific field to train the method to classify arriving information into clusters. Let i denote the clean malicious traffic data from a given dataset, and o denote the predicted class of network traffic sample by IDS. The processing of the IDS model is defined by 0 $g : i \rightarrow o$. These algorithms are vulnerable to malicious attacks, in which a malicious attacker known as an adversary deliberately alters the input data to mislead the learning algorithm into misclassification. The adversarial sample is defined using Eq. 1.

$$i^* = i + \delta \tag{1}$$

where i^* means the adversarial example generated from i and δ means the magnitude of the adversarial perturbation. Adversarial sample generation and training of IDS for classifying training samples into true and adversarial instances are required. The loss associated with adversarial sample generation can be minimized using Eq. 2.

$$\text{arg}min\|\delta\|,\ i^* \neq i \tag{2}$$

The probability (P_{adv}) of training data belonging to a specific class m (m = true or adversarial) misclassified by the discriminator module is determined using Eq. 3.

$$P_{adv} = \frac{N_{misclassified}}{T_{train}} \tag{3}$$

$N_{misclassified}$ indicates the number of training instances misclassified by the discriminator. T_{train} indicates the total training instances.

The objective function for optimizing the adversarial training dataset for IDS is defined by Eq. 4.

$$G_{adv} = w_1.\delta + w_2.P_{adv} \tag{4}$$

Objective function minimization is the optimization problem for developing a corrected adversarial training dataset for IDS. Table 2 depicts the notations of the problem definition.

3 Methodology

This section describes the proposed WCSAN-PSO-based framework. The proposed framework is illustrated in Fig. 2. First, the publicly available CIC-IDS2017 dataset [58]

Table 2 Notations of problem definition

Symbols	Description	Problem definition
i^*	Adversarial sample for data ‘i’	Injection of i^* into input data forces IDS to larger misclassifications
δ	The magnitude of the adversarial perturbation	The magnitude of adversarial perturbation influences adversarial training against unknown adversarial attacks
P_{adv}	Probability of misclassifications by the discriminator	P_{adv} affects the adversarial training of IDS
G_{adv}	The objective function for optimizing the adversarial training dataset for IDS	G_{adv} must be minimized for efficient adversarial training of IDS

(https://www.unb.ca/cic/datasets/ids-2017.html) is collected and normalized using preprocessing. Network traffic PCA extracts features and selects the feature using LASSO. These methods are further complemented by the subsequent steps involving labeling attacks and managing signature lists, resulting in reduced system bandwidth usage and streamlined computing processes. Then, WCSAN is employed to create a corrected training dataset with correct labels of true and adversarial network traffic instances for IDS adversarial training. PSO optimizes the parameters of WCSAN to enhance the adversarial training process. The primary focus of the proposed framework is leveraging signatures to identify destructive patterns. Signatures are distinct traits or patterns connected to particular sorts of attacks. The system can effectively identify well-known attack patterns by employing and updating signatures based on the known attacks. High bandwidth utilization and computing processes for device connection could be drawbacks of existing approaches. This system alleviates the problem by effectively managing signatures and minimizes the data that must be sent over the network. The IDS is trained on a corrected adversarial training dataset to classify true and adversarial samples. Finally, IDS is trained on true network traffic data to classify the true samples into benign and malicious instances. The efficiency of the IDS is validated with the proposed WCSAN-PSO-based adversary training by comparing without adversary training and classification with the SVC classifier.

3.1 Data collection

This study uses the publicly available Canadian Institute for Cybersecurity CIS-IDS2017 dataset (https://www.unb.ca/cic/datasets/ids-2017.html). The dataset is available in both CSV and PCAPs format. It includes most updated attacks like Bot, PortScan, Infiltration, Web Attack Brute Force, Web Attack Sql Injection, Heartbleed, SSH-Patator, DoS Hulk, FTP-Patator, DoSGoldenEye, Web Attack XSS and DoSslowloris, and normal records. The CIC Flow Meter analyzes the network traffic features of this dataset. Table 3 shows the description of the dataset.

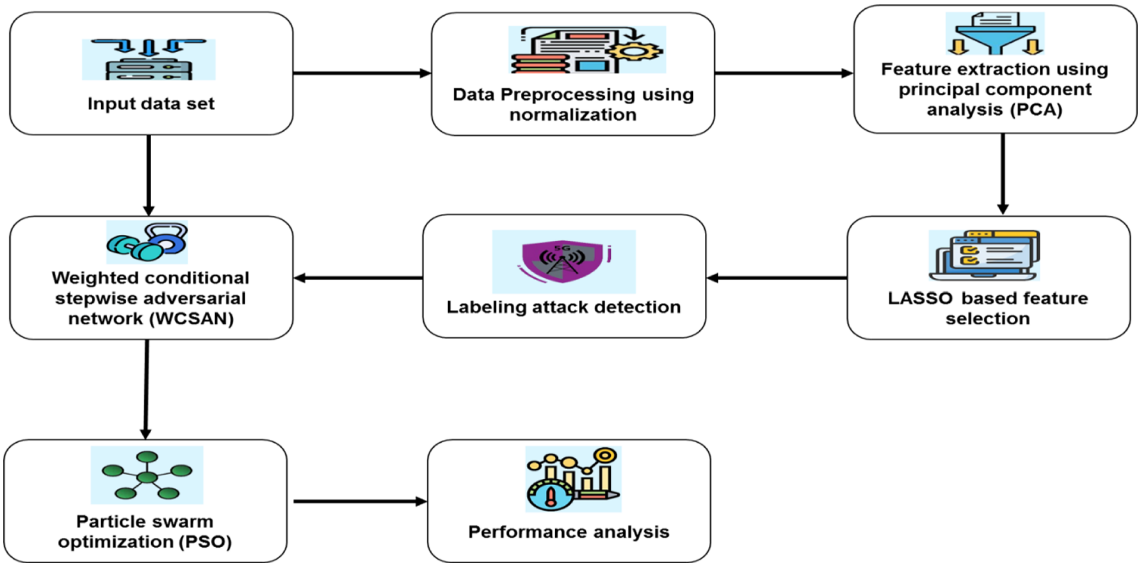

Input data set
Data Preprocessing using normalization
Feature extraction using principal component analysis (PCA)
LASSO based feature selection
Labeling attack detection
Weighted conditional stepwise adversarial network (WCSAN)
Particle swarm optimization (PSO)
Performance analysis

Fig. 2 The proposed framework

Table 3 Description of the dataset

Dataset	Normal	Attack	Total
Training	49105	–	49,104
Testing	59,415	1966	12,277
Total	108,520	1966	61,381

3.2 Data preprocessing using normalization

Each network traffic feature sample is preprocessed to remove the irrelevant network traffic features. Noisy data can insignificantly influence the forecast of any influential data. The missing values and noises are moved from the dataset in data cleaning [59]. The labels in the dataset have string values encoded into numerical values corresponding to each label. Before feeding the dataset to IDS, the features are correctly scaled to 0 and 1 to avoid some features overlooking others [60]. The maximum normalization approach is employed. Assume the variables as $a^{x} = a_{y}^{x}, \ldots\ldots.., a_{m}^{x}$, where x ∈ n, y ∈ m. The number of variables is defined by 'n,' and the number of data corresponding to each variable is defined by 'm'. The normalization for each network traffic variable is performed using Eq. 5.

$$G_{y}^{x} = \frac{a_{y}^{x} - \min(a^{x})}{\max(a^{x}) - \min(a^{x})} \tag{5}$$

where G_{y}^{x} defines the standardized value of a specific variable, and a_{y}^{x} denotes the actual value of a specific variable. Min (a^{x}) and max (a^{x}) refer to the minimum and maximum value of a variable a^{x} correspondingly.

3.3 Principle component analysis using feature extraction

Using PCA, essential features that contribute to the PCA intrusion detection process are extracted from the preprocessed feature set. PCA has been widely used because of its simplicity, ease of understanding, and lack of constraining parameters. Employing PCA, m-dimensional network traffic variables can be l-dimensional reduction network traffic features [61]. To fulfill its dimension reduction objectives, the PCA eliminates data duplication, compromising the smallest quantity of information. These steps of PCA are as follows.

Step 1: The stages are grouped into PCA using Eq. 6 among the following groups: h = $h_1, h_2, \ldots.., h_J$.

$$\propto = \frac{1}{j}\sum_{n=1}^{j} h_a \tag{6}$$

where j shows the decision made in the example n = 1, and…, j

Step 2: Employing the sample mean, the covariance matrix for the test set is computed using Eq. 7.

$$P = \frac{1}{j}\sum_{a=1}^{j}(h_a - \propto)(h_a - \propto)^{o} \tag{7}$$

where P is the sample set's correlation matrix.

Step 3: The feature values and vectors of the samples' covariance matrix may be identified using Eqs. 8, 9, and 10.

$$P = K.\Sigma.K^{T} \tag{8}$$

$$\Sigma = \mathrm{diag}\left(\lambda_1, \lambda_{2,\ldots,}\lambda_s\right)\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_s \geq 0 \tag{9}$$

$$K = \left[\mathrm{k}_1, \mathrm{k}_{2,\ldots,}\mathrm{k}_s\right] \tag{10}$$

P is the quality values of m covariance matrices that have been organized diagonally and are down-ordered; attribute values of covariance matrices λ_j are shown below, together with the property vector. k_j Of feature value λ_j is used to create a quality matrix. K, i = 1…, s.

Step 4: For the first l-row main items, use Eq. 11 to calculate the cumulative deviations pension contribution using feature vectors and feature ratings produced from the first l-row primary components.

$$\theta = \sum_{j=1}^{l}\lambda_j \Big/ \sum_{i=1}^{n}\lambda_j \tag{11}$$

where θ shows the cumulative variations contribution level of the past l-row fundamental modules and is typically equal to or more than 0.9, the component should, in theory, be as high. The component θ of has to be properly chosen for a problem to be resolved from a realistic viewpoint. Particulars of an originally restated selection: If the value is properly selected, the main components for k-row collection may be determined.

Step 5: Utilize and reduce the collected vector size with q-row features using Eqs. 12 and 13.

$$A = K_l \tag{12}$$

$$X = A.Y \tag{13}$$

The relevance of quality for the first k-row (l ≤ n), P is a matching quality vector, was used to create the characteristic matrix. A feature's first k rows matrix Q_l, should be filled. Unbent information may then be converted from m-dimensional (Y) into linear (X), the dimensions needed for linearization.

3.4 LASSO-based feature selection and labeling attack detection

A safe and effective method for selecting a small number of significant network traffic characteristics from the above-obtained feature set is feature selection. These methods usually remove superfluous or inconsequential functionalities or characteristics deeply correlated in the information without causing significant data loss [62]. It is a popular model for simplifying translation and ramping up supposition by lowering variance. The estimated LASSO function can be calculated using Eqs. 14, 15, and 16.

$$\beta^{lasso} = arg_{\beta}min\left\{ \frac{1}{2}\sum_{x=1}^{M}\left(j_y - \beta_0 - \sum_{y=1}^{t} i_{xy}\beta_y \right)^2 + \lambda\sum\nolimits_{y=1}^{t}\left|\beta_y\right| \right\} \tag{14}$$

$$\beta^{lasso} = arg_{\beta}min\sum_{x=1}^{M}\left(j_y - \beta_0 - \sum_{y=1}^{t} i_{xy}\beta_y \right)^2 \tag{15}$$

$$\sum_{y=1}^{t}\left|\beta_y\right| \le p \tag{16}$$

LASSO replaces each correlation value with a continuous component that shortens at zero. Anticipating the feature selection technique is advantageous. It reduces the unutilized sum of squares forced to submit to a total of the entire correlation coefficient estimation to less than full conformity. The LASSO improves the direct learning model, precision, and accuracy by combining the benefits of perimeter depressive episodes and subset shortlisting.

A data instance's label indicates whether the instance is normal or suspicious. The labeled data set for training is obtained. Anomaly behaviors are often dynamic; for example, new anomalies can develop without labeled training information. This work used four classification levels, presented in Table 4: 0 for begin network traffic as non-attack, 1 for attacks. If any attack is an attack, the types of attacks are maintained and updated in the dataset so that similar attacks can be predicted earlier while consuming bandwidth and computing resources. The flow diagram for maintaining the attack dataset and attack labeling is demonstrated in Fig. 3. Initially, information about network traffic behavior is gathered for system analysis. After data gathering, the information is labeled to differentiate between known and unknown behavior. The system uses the suggested framework to identify and categorize unknown or novel assaults when it detects one different from known signatures. The proposed framework quickly recognizes the attack and does not need further processing if the acquired data sample matches known attack signatures. The IDS decides whether to generate alerts, take appropriate action in response, or do additional analysis based on the labeled data. This approach reduces the total amount of data sent over the network, which assists in preserving bandwidth resources while maintaining the accuracy of threat detection through signatures.

Table 4 Log entry of labeled data

Classification	Log	Label	Predicted label
0	BENIGN	Non-attack	Non-attack
1	Bot	Attack	Attack
	PortScan	Attack	Attack
	Infiltration	Attack	Attack
	Web Attack Brute Force	Attack	Attack
	Web Attack Sql Injection	Attack	Attack
	Heartbleed	Attack	Attack
	SSH-Patator	Attack	Attack
	DDoS	Attack	Attack
	DoS Hulk	Attack	Attack
	FTP-Patator	Attack	Attack
	DoSGoldenEye	Attack	Attack
	Web Attack XSS	Attack	Attack
	DoSslowloris	Attack	Attack

3.5 Handling the class imbalance problem

The class imbalance is a common problem in IDS. The substantial difference between the number of typical scenarios and the low frequency of attack cases is the root cause of this problem. The synthetic minority oversampling technique (SMOTE) is used in this study to address the issue. The SMOTE technique interpolates between the given data points to generate fictional cases for the underrepresented class. The preprocessed data are correctly handled, which includes encoding class variables, deleting unnecessary features, and handling missing values [7]. The datasets are then split into training and testing datasets associated with characteristics (a) and labels (b). The instances are built using the SMOTE training set of data using Eq. 17.

$$asynthetic = aminority + randomnumber * (n - aminority) \tag{17}$$

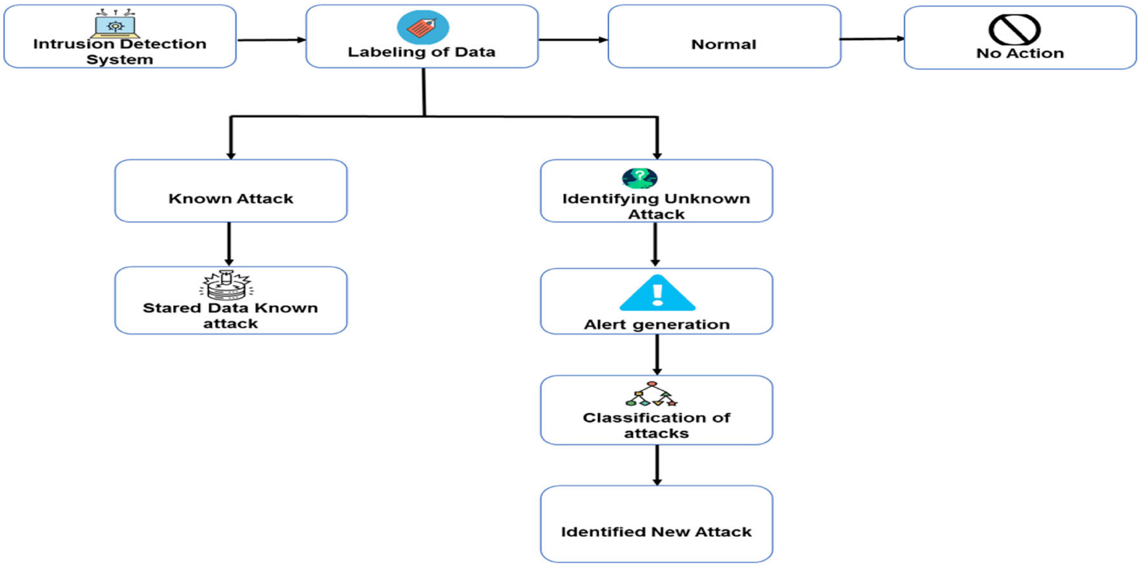


Fig. 3 Flow diagram of labeling

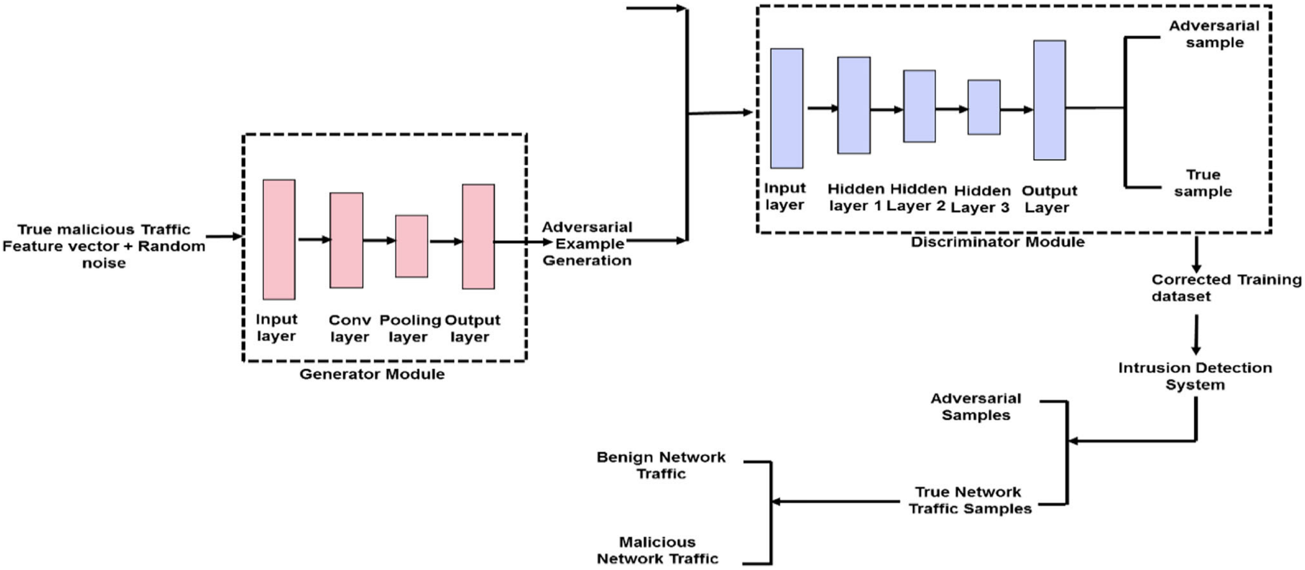


Fig. 4 Architecture of the WCSAN-based IDS

Let's assume there is a dataset with labels b and features a. The K-nearest neighbors of each minority instance,$a\_minority$, from the minority class must be located. In ($a\_minority$), a synthetic instance $a\_synthetic$ is created for every neighbor n. The random number that controls the interpolation between $a\_minority$ and n is a random number between 0 and 1.

3.6 Weighted conditional stepwise adversarial network particle swarm optimization (WCSAN-PSO)

3.6.1 Weighted conditional stepwise adversarial network (WCSAN)

The generator (G) generates adversarial network traffic feature values from the network traffic records. The generator is based on a convolutional neural network. It includes the input, convolutional, pooling, and output layers [63]. Network traffic data, which usually includes features, is received by the input layer of WCSAN-based IDS. This data forms the basis for further analysis. G takes true network traffic features and Gaussian noises δ as input and generates an adversarial network traffic feature vector using Eq. 18. This generated feature vector is labeled as an adversarial traffic sample.

$$\left[I^{*}\right] = \left\{i_1^{*}, i_2^{*}, \ldots\ldots, i_h^{*}\right\}, i^{*} \neq i \tag{18}$$

where $\left[I^{*}\right]$ means the adversarial network traffic feature vector, i indicates the clean network traffic features, and i^{*} means the adversarial network traffic features. The adversarial training dataset combines true (clean) and adversarial network traffic features. This adversarial training dataset is sent as input to the discriminator module of WCSAN. The discriminator module of WCSAN is designed based on a neural network. Discriminators are trained on an adversarial training dataset (I') to distinguish between true and adversarial network traffic samples. Game training is used to modify model feature weights between the network entities to update the model's generalization capacity. The output of the discriminator can be defined using 19.

$$O_D = \begin{cases} 1, \ I_{j'} isadversarial \\ \quad 0, \ I_{j'} istrue \end{cases} \tag{19}$$

where $I_{j'}$ means the j^{th} sample of the adversarial training dataset (I'), and O_D means the adversarial classification result of the discriminator (D). The adversarial classification result of the discriminator is that one of the samples is predicted as adversarial and zero if the sample is true. The corrected training dataset containing correct labels of true and fake network traffic records obtained from the discriminator is provided to the IDS. The proposed architecture is shown in Fig. 4. The discriminator module's corrected training dataset is useful to identify and resist adversarial attempts to IDS. First, the IDS is trained to discriminate between samples that are categorized as adversarial samples and samples that are true instances. The IDS acquires the capacity to distinguish between efforts at subversion by adversaries and normal network traffic during the training phase. Then, the IDS continues a further training program to distinguish between two types of network data: malicious and benign. The IDS can distinguish between malicious activity that could be an attack and regular network traffic, which does not affect the system's performance due to its dual classification capacity.

The corrected training dataset obtained from the discriminator module is used to train the IDS on adversarial attacks. The IDS is initially trained to classify samples into true and adversarial instances. Then, the IDS are trained to categorize the true network activity samples into benign and malicious network data. The proposed algorithm for WCSAN-based adversarial classification is presented in Algorithm 1.

Algorithm 1 WCSAN-based adversarial classification

```
Phase 1: Adversarial Network Traffic Generation by Generator module

    Input: Clean Network Traffic Feature Set [I]
    Output: Adversarial Network Traffic Training Dataset [I']
    [I]= {i_i,., i_h}, where h is selected true network traffic features, [I*] as an adversarial network feature vector
        For (j=1, j ≤ h, j=j+1)
        {
                $i_j^* = i_j + \delta$, where i* means the adversarial example of j^th feature, append $i_j^*$ to [I*]
        }
        Create $[I']$ by combining [I] and [I*].
Phase 2: Adversarial Detection by Discriminator module
    Input: Adversarial Network Traffic Training Dataset [I'']
    Output: Corrected Network Traffic Training Dataset $[I'']$
    Let [I']= {$i'_1, i'_2, \ldots\ldots, i'_n$)} where n → number of samples in [I']
    for sample in [I']:  $O_D$result = $O_D$ (sample)
      if $O_D$result = 0:
    For (j=1, j ≤ n, j=j+1)
                Compute $O_D(i'_j)$ as classification result of sample$(i'_n)$
                if ($O_D(i'_j)$=0)
                        Label $i'_j$ as true sample
                Else
                        Label $i'_j$ as adversarial sample and construct corrected training dataset $[I'']$
    Phase 3: Intrusion detection by IDS module
    Let [I'']= {$i''_1, i''_2, \ldots\ldots, i''_n$)} where n → number of samples in [I'']
    For (j=1, j ≤ n, j=j+1)
                {
            Compute $O_{IDS_1}(i''_j)$ as classification result of sample$(i''_j)$
            ($O_{IDS_1}(i''_j)$=0),append  $i''_j$ to $[I''_{IDS}]$
                }
    Let $[I''_{IDS}]$]= {$i''_1, i''_2, \ldots\ldots, i''_h$)} where h → number of true samples in $[I''_{IDS}]$]

    For (j=1, j ≤ h, j=j+1)
        Compute $O_{IDS_2}(i''_j)$ as classification result of sample$(i''_j)$ according to $O_{IDS_2}(i''_j) =$
$\begin{cases} 1, i''_j \text{ is malicious} \\ 0, \ i''_j \text{ is benign} \end{cases}$
    End
```

The flow diagram of the WCSAN-based adversarial classification is presented in Fig. 5.

3.6.2 Particle swarm optimization (PSO)

PSO optimizes the parameters of the generator and discriminator modules of WCSAN to enhance the performance of the adversarial training of IDS. The PSO algorithm is associated with the social behavior of birds flocking and fish schooling [64]. When an independent fish or bird (quantum-state) decides on where to keep moving, three components are recognized at the same time: (a) its prevailing movable strategy (rate of change) based upon that inertia of the movement, (b) it is ideal position so far with, and (c) the most robust option that its neighbor particles have accomplished thus far using Eqs. 20 and 21. In the automated system, the particles form a swarm, and each material can represent an effective solution to the issue.

$$B_x^{p+1} = e^* B_x^p + f_1 * Rand() * (t_x^p - I_x^p) + f_2^* Rand() * (t_k^p - I_x^p) \tag{20}$$

$$I_x^{p+1} = B_x^p + B_x^{p+1} \tag{21}$$

$$I_x = (I_{x1}, I_{x2}, .., I_{xM}) \tag{22}$$

$$B_x = (B_{x1}, B_{x2}, .., B_{xM}) \tag{23}$$

where x represents the number of active nodes, p is the number of points, and B and I are the granules' kinetic energy and placement matrices. Equations 22 and 23 show that M particle dimensions can represent B and I in an N-dimensional problem (22).

Algorithm 2 PSO algorithm

```
Step 1: Initialization
For every material x = 1, . . . ,$M_t$, do
  (a) $T_x(0) \sim Q$ (LB, UB), where LB and UB symbolize the threshold values of the solution space, is used to initialize the quantum state location with a density.
  (b) Set $t^{best}$ to its starting stance: t best(x, 0) = $t_x(0)$.
  (c) Set g to the swarm's smallest possible significance: $k^{best}$ (0) = argmin c[$t_x(0)$].
  (d) Transform the original acceleration: $b_x \sim Q$ ($-|UB - LB|$, $|UB - LB|$).
Step 2: Continue till the conclusion is reached.
For every material x = 1, . . . ,$M_t$, do
  (a) Choose different numbers: $o_1$, $o_2 \sim Q$ (0, 1).
  (b) Modify the quantum state speed
  (c) Hadron's position is updated.
  (d) If c[$T_x$ (p)] < c[$t^{best}$ (x, p)]
    (i) Modify the molecule's strongest value x: $t^{best}$ (x, p) = $T_x$ (p).
    (ii) Modify the swarm's better-known direction if c [$T_x$ (p)] < c[$k^{best}$ (p)]:$k^{best}$ (p) = $T_x$ (p).
  (e) P ← (p + 1);
Step 3: Return $k^{best}$(p) which contains the perfect option identified.
```

The inertia weight e adjusts the predisposition to enhance global adventure (smaller e). The natural inclination to accommodate local adventure (larger e) to fine-tune this same current search agent (larger e), Rand (), comes back with a spontaneous ranging between [0, 1], and f_1 and f_2 are constant operating numbers used to control the influence of t_x and t_k. After each particle's velocity has been updated, the locations of the particles are updated using Eq. 23. Equations 24 and 25 construct the particles' initial position and velocity vectors.

$$I_{x,g} = I_{min} + (I_{max} - I_{min}) \times q_1 \tag{24}$$

$$B_{x,g} = B_{min} + (B_{max} - B_{min}) \times q_2 \tag{25}$$

The PSO algorithm is presented in Algorithm 2.

4 Result and analysis

This section presents the analysis of IDS with the WCSAN-PSO framework in classifying network traffic into benign and malicious samples. The evaluations are employed in the Python environment. The experimental setup was carried out on a single PC with 64-bit Windows 11 and an Intel Pentium CPU with 32 GB RAM and 500 GB SSD. The study uses an SVC classifier for classification [65]. The performance indicators for the analysis of the proposed framework are precision, accuracy, F1-score, recall, ROC, and AUC value, which are explained below.

Accuracy is the proportion of correct classifications of network traffic instances out of total samples made by the IDS, using Eq. 26.

$$Accuracy = \frac{l + m}{l + m + n + 0} \tag{26}$$

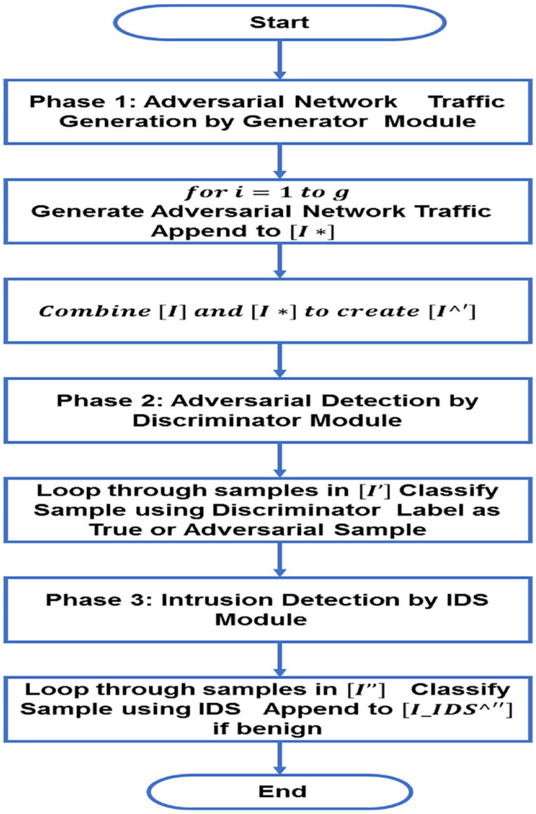

Start
Phase 1: Adversarial Network Traffic Generation by Generator Module
for i = 1 to g
Generate Adversarial Network Traffic
Append to [I *]
Combine [I] and [I *] to create [I^']
Phase 2: Adversarial Detection by Discriminator Module
Loop through samples in [I'] Classify Sample using Discriminator Label as True or Adversarial Sample
Phase 3: Intrusion Detection by IDS Module
Loop through samples in [I''] Classify Sample using IDS Append to [I_IDS^''] if benign
End

Fig. 5 Flow diagram of WCSAN

where l (known as true positive) denotes the quantity of true malicious network traffic instances correctly classified as malicious network traffic instances, m (known as true negative) indicates the amount of true benign network traffic instances accurately categorized as benign network traffic instances, n (false positive) represents the number of true benign network traffic instances misclassified as malicious network traffic instances, and o (false negative) denotes the number of true malicious network traffic instances misclassified as benign network traffic.

Precision is determined as the proportion of network traffic samples correctly identified as malicious out of samples identified as malicious instances, using Eq. 27.

$$Precision = \frac{l}{l + n} \tag{27}$$

The recall is defined as the proportion of network traffic samples correctly identified as malicious out of total malicious network traffic samples, using Eq. 28.

$$Recall = \frac{l}{l + o} \tag{28}$$

The weighted ratio is the F1-score of recall and precision, using Eq. 29.

$$F1 - score = \frac{2 * precision * recall}{precision + recall} \tag{29}$$

The Detection Rate (DR) can be defined using Eq. 30.

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

where TP stands for True Positive and FN for False Negative.

The Area Under the ROC Curve (AUC) is a commonly utilized performance measure in classification assignments. The metric quantifies the ability of a classification model to differentiate between positive and negative instances by calculating the probability that a randomly selected positive instance will be ranked higher than a randomly selected negative instance. The ROC curve illustrates the relationship between the true positive rate (DR) and the false positive rate (1-specificity) across different classification points, with specificity calculated using Eq. 31.

$$\text{specificity} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{31}$$

where TN stands for True Negative and FP for False Positive.

The AUC is the area under the curve, ranging from 0 to 1. A value of 1 signifies an ideal classifier, while a 0.5 value indicates an ineffective classifier. Greater AUC values signify superior model performance in differentiating between positive and negative samples.

To evaluate the effectiveness of the proposed IDS with the WCSAN-PSO defense framework in adversarial attacks, we have chosen the attack leveling, as illustrated in Fig. 3. Three scenarios are presented in this section, demonstrated in Fig. 6. In the first scenario, the IDS is trained with the original network traffic dataset and generates network traffic samples with no defence mechanism and without an adversarial attack dataset. In the second scenario, the IDS is trained with the original network traffic dataset and adversarial samples generated from WSCAN with no defence mechanism. The classification is based on an imbalanced dataset for the first and second scenarios. In the third scenario, IDS is trained with the original network traffic dataset, adversarial samples generated from WSCAN, and a corrected training dataset with a defence mechanism. The proposed framework is evaluated in both balanced and imbalanced datasets.

4.1 Scenario 1

The original network traffic dataset is pre-processed and normalized, features are extracted using PCA, and features are selected using LASCO. The attacks are leveled. Network

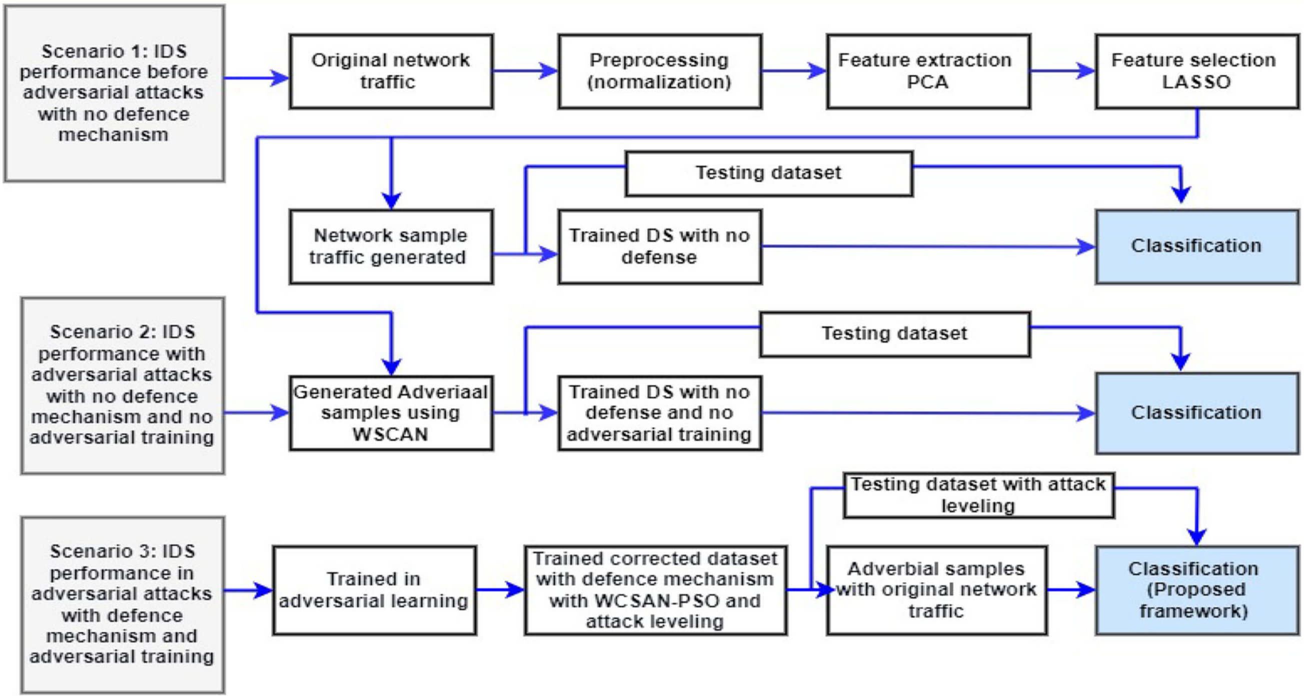

Scenario 1: IDS performance before adversarial attacks with no defence mechanism
Original network traffic
Preprocessing (normalization)
Feature extraction PCA
Feature selection LASSO
Testing dataset
Network sample traffic generated
Trained DS with no defense
Classification
Scenario 2: IDS performance with adversarial attacks with no defence mechanism and no adversarial training
Testing dataset
Generated Adveriaal samples using WSCAN
Trained DS with no defense and no adversarial training
Classification
Testing dataset with attack leveling
Scenario 3: IDS performance in adversarial attacks with defence mechanism and adversarial training
Trained in adversarial learning
Trained corrected dataset with defence mechanism with WCSAN-PSO and attack leveling
Adverbial samples with original network traffic
Classification (Proposed framework)

Fig. 6 Three evaluation scenarios for the analysis of IDS in adversarial attacks

Table 5 Transformed extracted features with generated network samples and original network dataset

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
– 2.3874579	– 0.0520089	0.4039036	– 0.4212386	– 0.4444382	– 1.0774039	0.2511127	0.183508	– 0.2085869	– 0.0637675
– 2.800229	– 0.298451	0.5778567	1.9874548	– 0.1287246	– 0.8104782	– 1.700661	1.3027706	0.1607661	– 0.1771233
0.7000989	1.4046537	– 4.1112846	– 0.889538	– 0.4381019	0.0350623	1.0478822	– 1.0805345	0.0869719	0.0903951
0.685761	1.2945483	– 3.7652078	– 0.9346478	– 0.1077882	– 0.0921702	1.039185	– 0.8611829	0.0214558	0.0956127
0.3151581	2.0196977	– 1.5316126	– 1.2665882	0.5241469	– 0.7385545	0.8348879	– 0.3901929	– 0.3537512	0.0472945

samples are generated and combined with the original traffic to the dataset to train the IDS with no adversarial attack samples and without a defense mechanism. The imbalanced dataset is used, and the transformed extracted features with the combination of generated network samples and the original network dataset are illustrated in Table 5. The outcomes are tested with the testing dataset.

The four performance evaluation parameters considered are accuracy, recall, F1-score, and precision. The outcomes are presented in Table 6, and it achieved an accuracy of 93.58% in detecting normal traffic and 90.74% in detecting malicious traffic without an adversarial scenario and no defense mechanism.

Figure 7 demonstrates the Receiver Operating Characteristic (ROC) curve with the Area under the ROC Curve (AUC) value and shows an AUC value of 0.92 in the imbalanced dataset in scenario 1.

Table 6 The IDS before adversarial attacks on the dataset

Performance indicators	Category	Outcome (%)
Accuracy	Benign	93.58
	Malicious	90.74
Precision	Benign	70.67
	Malicious	68.35
Recall	Benign	78.58
	Malicious	77.67
F1-score	Benign	75.12
	Malicious	74.89

4.2 Scenario 2

In scenario 2, the adversarial samples are generated with WCSAN. The IDS is trained with the original network

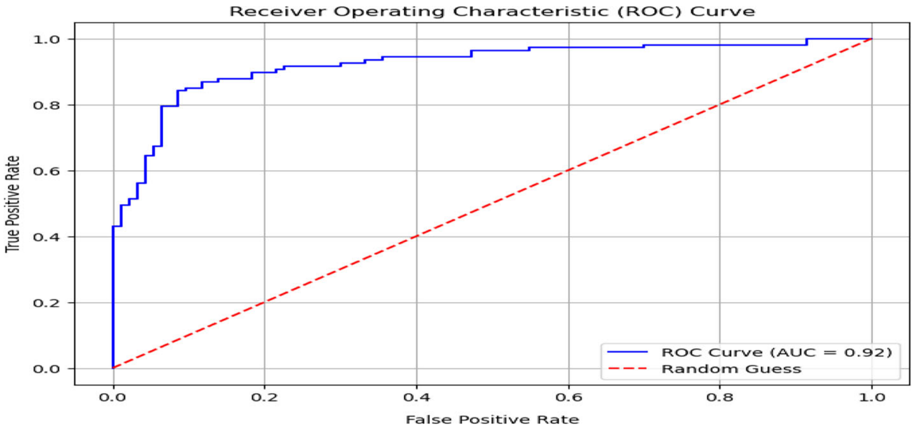


Fig. 7 ROC Curve with the AUC value

dataset, and the adversarial samples are generated using the WCSAN with no defence mechanism and without adversarial training. The imbalanced dataset is used in scenario 2, and the transformed extracted features with the combined adversarial sample and the original training dataset are illustrated in Table 7. The outcomes are tested with the testing dataset.

The four performance evaluation parameters considered are accuracy, recall, F1-score, and precision. The outcomes are presented in Table 8. The IDS yielded in the detection of normal packets an accuracy of 92.78%, precision of 74.67%, recall of 77.58%, and f1-score of 75.12% in an adversarial attack scenario. In detecting attacks, IDS achieved an accuracy of 85.72%, precision of 69.35%, recall of 73.67%, and f1-score of 75.89% in adversarial attack scenarios. However, the accuracy, precision, recall, and F1-score of the IDS with no defense mechanism, tested on a network traffic dataset with adversarial samples, was lower than the one without adversarial examples. This signifies that the adversarial attacks generated by the WCSAN compromise the performance of the IDS compared to scenario 1. Adversarial samples increase the number of false positives and force the IDS to learn erroneous decision limits, as seen by the decrease in IDS performance in an adversarial environment. This signifies that the outcome is impacted by detecting adversarial attacks in scenario 2.

Table 8 IDS performance after adversarial attacks with no defense

Performance indicators	Category	Outcome (%)
Accuracy	Benign	92.78
	Malicious	85.72
Precision	Benign	74.67
	Malicious	69.35
Recall	Benign	77.58
	Malicious	73.67
F1-score	Benign	75.12
	Malicious	75.89

The performance of the IDS with WCSAN-PSO-based adversarial training is further tested. The WCSAN is trained for 1000 iterations to check the performance in determining adversarial samples from each 200 iterations. The scatter plot of true versus adversarial samples for the WCSAN method is illustrated in Fig. 8. The orange distinguishes true network traffic samples, and the blue indicates adversarial samples. Table 9 depicts the classification accuracy of the discriminator of WCSAN for real and adversarial sample discrimination. PSO significantly enhances the WCSAN method's accurate and adversarial sample discrimination performance.

Figure 9 demonstrates the ROC curve with the AUC value and shows an AUC value of 0.84 in the imbalanced dataset in scenario 2.

Table 7 Transformed extracted features with the combined adversarial sample and the original training dataset

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
0.8488575	0.7560821	0.8146192	0.6425675	0.3576311	0.4183124	0.8347974	0.8718762	0.8269289	0.8409152
0.9046618	0.8666568	0.7539278	0.8956943	0.7569068	0.7451112	0.9503375	0.642829	0.9042664	0.9279852
0.4679091	0.3894138	0.7861135	0.6702121	0.6995128	0.7555856	0.9398418	0.8276241	0.6017654	0.936344
0.7794187	0.8210003	0.8543053	0.7655664	0.8767144	0.5711125	0.5910316	0.6405485	0.6951845	– 0.1163267
0.9054413	0.8010286	0.8662278	0.7714617	0.7346173	0.9112899	0.9548118	0.9651832	0.8967986	0.3505101

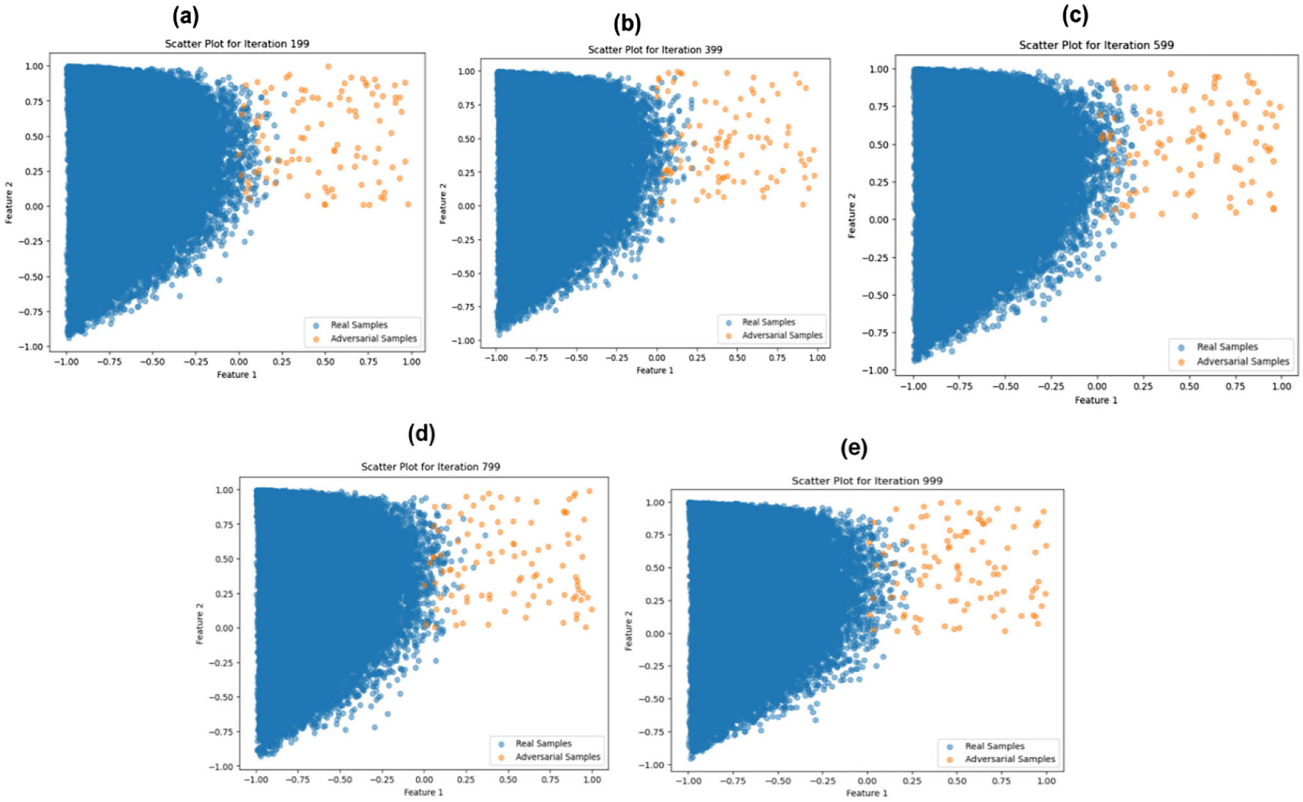


Fig. 8 Scatter plot of true versus adversarial samples in WCSAN **a** after 200 iterations. **b** After 400 iterations. **c** After 600 iterations. **d** After 800 iterations. **e** After 1000 iterations

Table 9 Classification accuracy of discriminator of WCSAN

Iteration	Real sample	Adversarial sample
199	0.98317855	1
399	0.98317855	1
599	0.98319892	1
799	0.98319892	1
999	0.98315819	1

4.3 Scenario 3

In scenario 3, The IDS is further trained on the combined dataset, i.e., the normal original traffic and adversarial samples generated from scenario 2. The IDS is trained with a corrected adversarial training dataset generated using the proposed WCSAN-PSO defense. The common problem in machine learning is addressing class imbalance, especially in IDS. The SMOTE is used in this study to address the data transformation issue from unbalanced to balanced. The proposed framework is evaluated on both balanced and imbalanced datasets. The third evaluation scenario with the WCSAN-PSO defense mechanism with adversarial training in the adversarial scenario is depicted in Fig. 6 and evaluated using a balanced and imbalanced dataset. The value counts for each data class in imbalanced and balanced datasets are shown in Fig. 10 (a) and (b), respectively.

It demonstrates that the value counts are not equal in an imbalanced dataset, and the value counts for all classes are equal when the data are balanced. The extracted transformed combined features for the corrected training and adversarial samples dataset generated by WCSAN-PSO for the imbalanced dataset are demonstrated in Table 10 and 11.

Figure 11 illustrates the confusion matrix for classifying network traffic samples into benign and attack samples by IDS with WCSAN-PSO-based adversarial training in the balanced dataset.

Table 12 exhibits the proposed framework's accuracy, precision, recall, and f1-score in detecting adversarial attacks with defense mechanisms in normal and malicious scenarios with adversarial training in the balanced dataset. Further, using a signature database is maintained for the known attack; it predicts initially without using bandwidth and computing resources. Once an unknown attack is detected, the proposed framework updates the signature database so that a similar attack can be predicted at the initial stage next time. This significantly enhanced the robustness and performance of the framework. The proposed framework achieved an accuracy of 99.36%, a precision of 98.96%, a recall of 97.56%, and

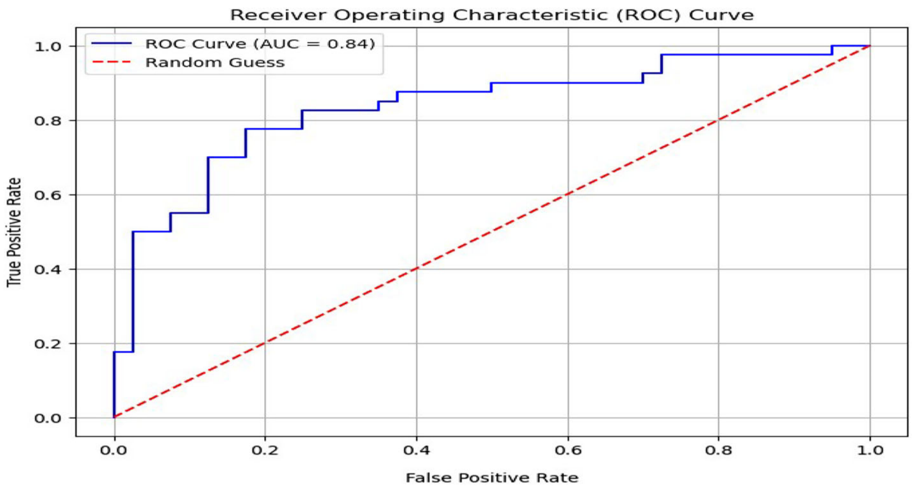
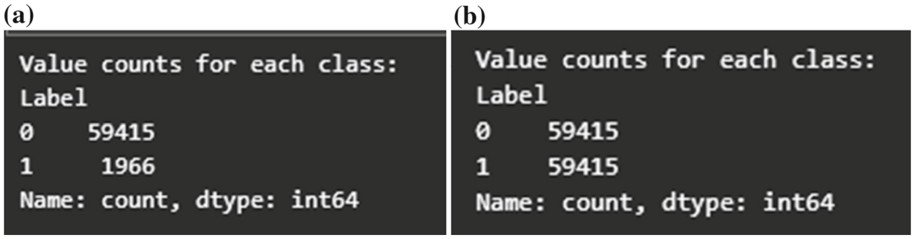


Fig. 9 ROC curve with the AUC value



(a)

```
Value counts for each class:
Label
0    59415
1     1966
Name: count, dtype: int64
```

(b)

```
Value counts for each class:
Label
0    59415
1    59415
Name: count, dtype: int64
```

Fig. 10 Value counts for each class **a** imbalanced dataset. **b** balanced dataset

Table 10 Transformed extracted features for the corrected training dataset and adversarial sample in the imbalanced dataset

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
0.1635971	0.2862763	− 0.3474317	− 0.9001193	0.9156828	− 0.2313149	− 0.790592	0.4767821	0.4169529	− 0.3781796
0.6447493	0.4485475	− 0.8793359	− 0.8039202	0.4526681	− 0.6978667	0.2069586	0.8472676	0.5431126	0.3478841
−0.3650534	0.6004493	− 0.3187721	− 0.6268917	0.4436878	− 0.5397366	− 0.1966288	0.3021472	0.3040808	0.206059
0.0454293	0.2957718	− 0.5242466	− 0.9403765	0.9371705	0.0972641	− 0.8363273	0.4402154	0.2895089	− 0.2385539
0.3529987	0.7379372	0.3213875	− 0.5913125	0.5550004	0.6555107	− 0.8531961	0.3473344	0.0643052	− 0.3071494

an f1-score of 95.54% in identifying normal samples. Meanwhile, detecting attacks yielded an accuracy of 98.55%, a precision of 97.33%, a recall of 94.96%, and an f1-score of 93.81%. This symbolizes that the proposed framework enhances the performance of detecting malicious attacks in adversarial scenarios after applying the defense mechanism compared to scenario 2.

Figure 12 displays the ROC curve with the AUC value for classifying network traffic samples into benign and attack samples by the proposed framework using the balanced dataset and achieving an AUC value of 0.99.

Table 13 displays the proposed framework's accuracy, precision, recall, f1-score, and AUC value in detecting adversarial attacks with adversarial training with the imbalanced dataset. The proposed framework performed an accuracy of

Table 11 Transformed extracted features for the corrected training dataset and adversarial sample in the balanced dataset

PC1	PC2	PC3	PC4	PC5	Pc6	PC7	PC8	PC9	PC10
0.7412561	0.3977996	0.6727271	0.7125944	0.5600536	0.7012566	0.7008215	0.8453897	0.6375687	0.7290632
0.9150863	0.9467107	0.5309644	0.8906117	0.9469074	0.572613	0.954946	0.975059	0.6177086	0.946415
0.9024052	0.8110246	0.0832738	0.860105	0.4311961	0.9057025	0.8233756	0.9682738	0.742722	0.8514427
0.7307583	0.7950293	0.1272996	0.6692681	0.5939809	0.7015559	0.7232251	0.8969641	0.5855946	0.7693119
0.9715712	0.9447051	0.279698	0.9250448	0.6990635	0.5575316	0.9361724	0.8332815	0.8031395	0.9500042

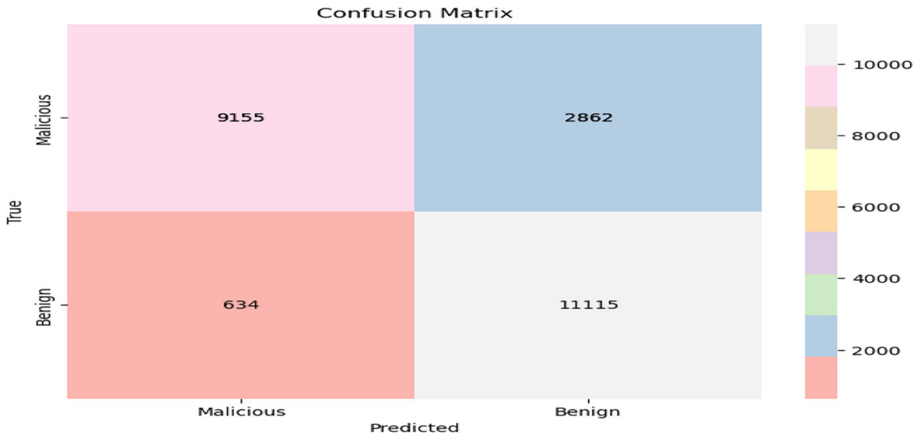


Fig. 11 Confusion matrix in the balanced dataset

Table 12 Performance analysis of the proposed framework in the balanced dataset

Performance indicators	Category	Proposed
Accuracy	Benign	99.36
	Malicious	98.55
Precision	Benign	98.96
	Malicious	97.33
Recall	Benign	97.56
	Malicious	94.96
F1-score	Benign	95.54
	Malicious	93.81

Table 13 Performance analysis of the proposed framework in the imbalanced dataset

Performance indicators	Category	Proposed
Accuracy	Benign	98.92
	Malicious	95.55
Precision	Benign	97.95
	Malicious	92.53
Recall	Benign	96.58
	Malicious	91.54
F1-score	Benign	92.64
	Malicious	92.35

98.92%, a precision of 97.95%, a recall of 96.58%, and an f1-score of 92.64% in identifying normal samples. However, detecting attacks achieved an accuracy of 95.55%, a precision of 92.53%, a recall of 91.54%, and an f1-score of 92.35%.

Figure 13 illustrates the ROC curve with the AUC value using an imbalanced dataset, which yielded an AUC value of 0.97.

The summary of the comparative performance analysis of the proposed framework using a balanced and imbalanced dataset is depicted in Fig. 14. The AUC value in the balanced

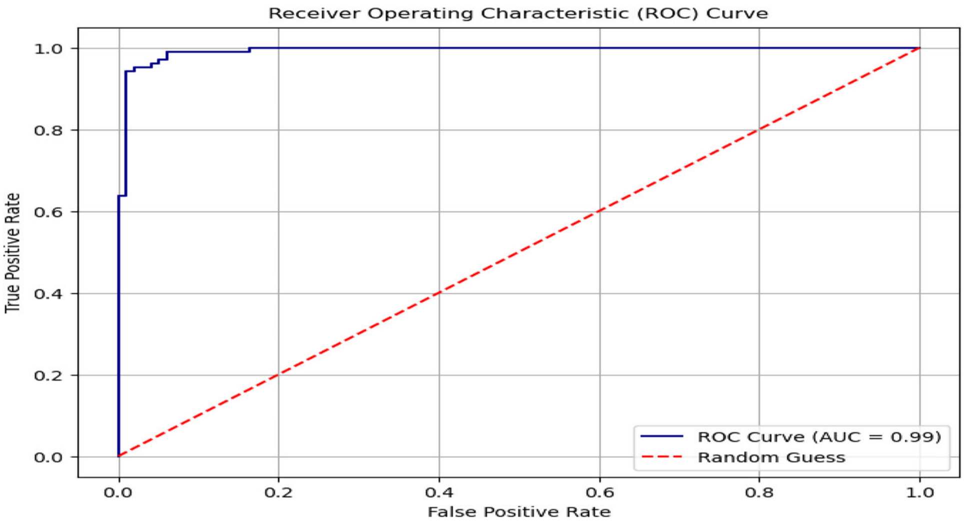


Fig. 12 ROC curve with AUC score in the balanced dataset

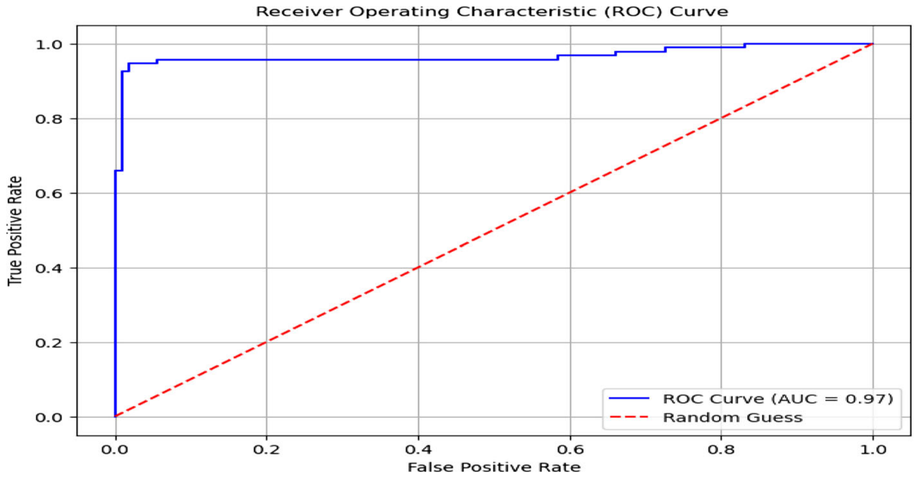


Fig. 13 ROC curve with AUC value in the imbalanced dataset

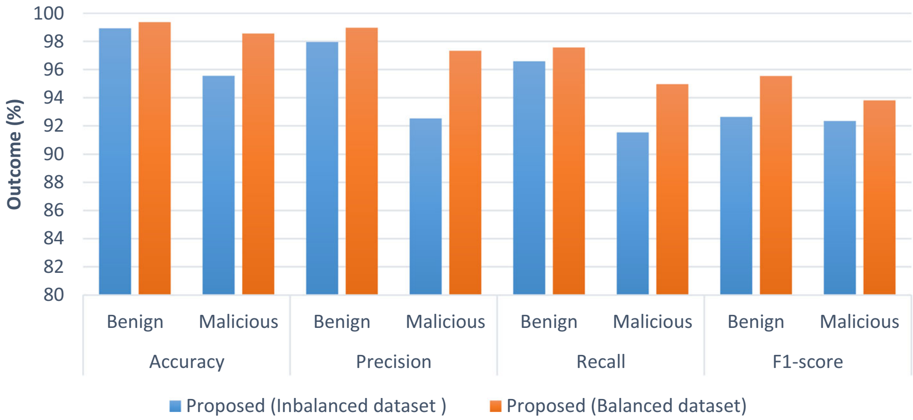


Fig. 14 Outcome comparison with the imbalanced and balanced dataset

dataset is 0.99, as demonstrated in Fig. 12, whereas using the imbalanced dataset is 0.97, as presented in Fig. 13. It indicates that the performance of the proposed framework is consistent but slightly better in the balanced dataset.

The outcome of the proposed framework is compared based on adversarial attack detection on IDS with the existing studies, namely IDS-ANN [50], C4N [51], RNN-ADV [53], DNN-IDS [54], JSMA [55], CNN-IDS [56] and Apollon [57]. The comparative analysis with the existing studies is presented in Table 14 and Fig. 15. The proposed framework achieved an accuracy of 98.55%, followed by IDS-ANN with an accuracy of 60%, C4N of 76.93%, RNN-ADV of 71.38%, DNN-IDS of 74.05%, JSMA of 97.3%, CNN-IDS of 89.4% and Apollon of 93.04%. The proposed framework yielded a precision of 97.33%, and JSMA demonstrates a precision of 97.3%.

5 Discussion

The identification and mitigation of malicious behavior and breaches of security is the preliminary function of IDS, which is essential to safeguarding computer networks and systems. Traditional IDS, however, are susceptible to adversarial attacks, in which hackers modify or obscure network traffic to avoid detection. Inadequate capacity for identifying known network attacks at the beginning stage, high false alarm rates, and inadequate feature engineering and selection increase the usage of high bandwidth and compute resources. IDS should successfully classify large-scale intrusion data in the complex network application environment. The proposed approach addresses the issues by incorporating adequate feature selection. extraction and maintaining updated signature-based systems, identifying the known attack at the initial stage and thus reducing computing resources.

Three scenarios are presented in this study, demonstrated in Fig. 6. In the first scenario, the IDS is trained with the original imbalanced dataset, and network samples are generated and tested with no defense technique. The details of the outcome with the SVC classifier are demonstrated in Table 6, and an accuracy of 93.58% in normal and 90.74% in attack detection is achieved. In the second scenario, the IDS model with no defense mechanism is trained using the original network traffic dataset and generated adversarial samples from the WCSAN, as demonstrated in Algorithm

Table 14 Comparative analysis with the existing studies

Performance indicators	IDS-ANN	C4N	RNN-ADV	DNN-IDS	JSMA	CNN-IDS	Apollon	Proposed
Accuracy	60	76.93	71.38	74.05	97.3	89.4	93.04	98.55
Precision	60	68.86	47.23	76.25	97.3	89.1	93.1	97.33
Recall	91	69.73	31.22	75.2	97.3	88.3	92.24	94.96
F1-score	11	69.29	37.59	74.22	97.3	89.21	88.35	93.81

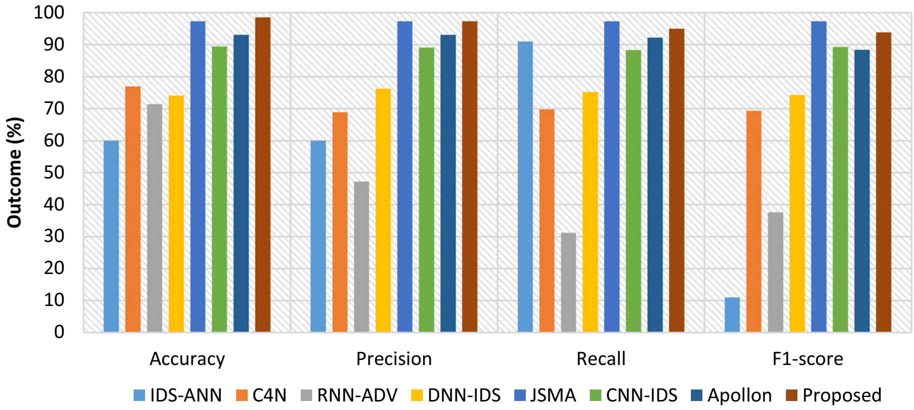


Fig. 15 Performance evaluations with the existing studies

1. The performance is evaluated on the test imbalanced dataset, and an accuracy of 92.78% in normal packets and 85.72% in attack detection is achieved, as demonstrated in Table 8 with the SVC classifier. This symbolizes that the adversarial attacks generated by the WCSAN reduce the performance of the IDS. The IDS is further trained with a corrected adversarial training dataset generated using the proposed WCSAN-PSO defense in scenario 3. It is tested on a dataset with an updated signature-based mechanism, as demonstrated in Fig. 3. The PSO optimization is demonstrated in Algorithm 2. The proposed framework is evaluated in balanced and unbalanced datasets to validate its effectiveness. The proposed framework in adversarial attacks with a defense mechanism achieved an accuracy of 99.36% in normal and 98.55% in detecting malicious attacks, as depicted in Table 12. The ROC curve with AUC value is demonstrated in Figs. 12 and 13 for balanced and imbalanced datasets, which signifies the performance is consistent but slightly better in the balanced dataset.

The comparative analysis with the existing studies in adversarial attack scenarios is presented in Table 14 and Fig. 15. However, it should be noted that existing studies are performed in different environments. The proposed framework accomplished an accuracy of 98.55%, whereas IDS-ANN of 60%, C4N of 76.93%, RNN-ADV of 71.38%, DNN-IDS of 74.05%, JSMA of 97.3%, CNN-IDS of 89.4% and Apollon of 93.04% in adversarial attack detection. The process is similar to adversarial sample generation. However, the proposed framework is distinct since it uses WCSAN-PSO to make IDS more resistant to adversarial concerns of known and unknown types while maintaining attack signature datasets. An increase in the intrusion detection performance of IDS with WCSAN-PSO-based adversarial training in adversarial conditions demonstrates that it pushed the IDS to learn and train efficiently between benign and malicious network traffic. The framework can be adapted to emerging adversarial techniques and attack patterns. Also, the proposed framework can be scaled to manage large datasets and high-throughput environments, making them suitable for real-time and high-performance applications in adversarial environments.

6 Limitations and future work

6.1 Limitations

This analysis of the study is based on one publicly available dataset. The study mainly concentrated on the attacks present in the dataset. The adversarial environment is extensive and constantly changing. Focusing solely on these particular attacks may cover a partial range of threats faced in real-world situations. The experiment used static datasets, which may not fully represent network traffic's dynamic and evolving nature and adversarial behaviors. Real-world IDS function in dynamic settings, and the research results may not completely correspond with these functional complications. The study examined different adversarial defense methods, but it was necessary to analyze all potential defense tools comprehensively. Various defense strategies could produce varying outcomes, necessitating further research. The study predominantly utilized traditional evaluation metrics such as accuracy, precision, recall, f1-score, and AUC. Although informative, these metrics must fully encompass the impact of adversarial attacks on IDS systems. Further metrics and practical testing could offer a more thorough evaluation.

6.2 Future work

Future research can explore the impact of emerging adversarial attack techniques on NIDS systems. It is paramount to stay updated on developing attack strategies to improve the resilience of NIDS. There is a tremendous opportunity to create a strong new framework to resist adversarial attacks for IDS. This framework should surpass existing known attacks and adjust to new threats, enhancing NIDS systems against adversarial attacks. Incorporating comprehensibility and model interpretation into NIDS models indicates significant potential. Explicit model predictions help analysts quickly detect adversarial attacks and develop efficient responses. Heuristic-based solutions are proficient at identifying new and unfamiliar threats, whereas verified countermeasures efficiently combat recognized threats. Combining the two achieves a thorough threat range, minimizing the chances of missing threats and triggering false alarms. Therefore, it would be a useful direction for research. The proposed framework can be extended by using different attacks and live datasets.

7 Conclusion

This study presented a proposed WCSAN-PSO-based framework on a weighted conditional stepwise adversarial network with particle swarm optimization and support vector classifier for classification to effectively detect adversarial attacks in IDS. The framework uses updated signature-based attack detection to predict known attacks in the first stage, which reduces computing resources. The study analyzed adversarial attacks and defense mechanisms through three comprehensive scenarios with practical and quantitative evaluation. The proposed framework achieved an accuracy of 99.36% in determining normal traffic and 98.55% in identifying malicious traffic in an adversarial attack scenario. The proposed framework yielded an AUC value of 0.99 in the balanced dataset and 0.97 using an imbalanced dataset, which signifies consistency. Adversaries may modify many network traffic features without affecting network behavior, making it difficult to detect intrusions. The future goal is to study the impact of the proposed framework on various ML and DL techniques. This approach can be expanded to explore the transferability concept in adversarial machine learning with advanced techniques. The proposed framework can be extended by considering different types of attacks, datasets, and optimization techniques to enhance attack detection, accuracy, and efficiency in reducing high false positive rates.

Author contributions Kousik Barik(KB), Sanjay Misra(SM), Luis Fernandez Sanz(LFS), KB and SM conceptualize the topic. KB, SM, and LFS are involved in Methodology, investigation, and validation. SM and LFS supervised the whole work. All authors reviewed the manuscript.

Funding Open access funding provided by Institute for Energy Technology.

Data availability The data is available at https://www.unb.ca/cic/datasets/ids-2017.html.

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 No ethical approval is required based on the following a. This article does not contain any studies with animals performed by any of the authors. b. This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Ince, K.: A novel approach for intrusion detection systems: V-IDS. Turk. J. Electr. Eng. Comput. Sci. **29**(4), 1929–1943 (2021)
2. Chimuco, F.T., Sequeiros, J.B., Lopes, C.G., Simões, T.M., Freire, M.M., Inácio, P.R.: Secure cloud-based mobile apps: attack taxonomy, requirements, mechanisms, tests and automation. Int. J. Inf. Secur. **22**(4), 833–867 (2023)
3. Amro, A., Gkioulos, V.: Cyber risk management for autonomous passenger ships using threat-informed defense-in-depth. Int. J. Inf. Secur. **22**(1), 249–288 (2023)
4. He, K., Kim, D.D., Asghar, M.R.: Adversarial machine learning for network intrusion detection systems: a comprehensive survey. IEEE Commun. Surv. Tutor. **25**(1), 538–566 (2023)
5. Javaheri, D., Gorgin, S., Lee, J.A., Masdari, M.: Fuzzy logic-based DDoS attacks and network traffic anomaly detection methods: classification, overview, and future perspectives. Inf. Sci. **626**, 315–338 (2023)
6. Park, N.E., Lee, Y.R., Joo, S., Kim, S.Y., Kim, S.H., Park, J.Y., Lee, I.G.: Performance evaluation of a fast and efficient intrusion detection framework for advanced persistent threat-based cyberattacks. Comput. Electr. Eng. **105**, 108548 (2023)

7. Xu, H., Sun, Z., Cao, Y., Bilal, H.: A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things. Soft. Comput. **27**(19), 14469–14481 (2023)
8. Lampe, B., Meng, W.: Intrusion detection in the automotive domain: A comprehensive review. IEEE Commun. Surv. Tutor. (2023). https://doi.org/10.1109/COMST.2023.3309864
9. Saheed, Y.K., Misra, S.: A voting gray wolf optimizer-based ensemble learning models for intrusion detection in the Internet of Things. Int. J. Inf. Secur. (2024). https://doi.org/10.1007/s10207-023-00803-x
10. Goyal, S., Doddapaneni, S., Khapra, M.M., Ravindran, B.: A survey of adversarial defenses and robustness in nlp. ACM Comput. Surv. **55**(14s), 1–39 (2023)
11. Rosenberg, I., Shabtai, A., Elovici, Y., Rokach, L.: Adversarial machine learning attacks and defense methods in the cyber security domain. ACM Comput. Surv. (CSUR) **54**(5), 1–36 (2021)
12. Apruzzese, G., Andreolini, M., Ferretti, L., Marchetti, M., Colajanni, M.: Modeling realistic adversarial attacks against network intrusion detection systems. Digital Threats **3**(3), 1–19 (2022)
13. Catillo, M., Del Vecchio, A., Pecchia, A., & Villano, U. (2023). A case study with CICIDS2017 on the robustness of machine learning against adversarial attacks in intrusion detection. In Proceedings of the 18th international conference on availability, reliability and security (pp. 1–8).
14. Lansky, J., Ali, S., Mohammadi, M., Majeed, M.K., Karim, S.H.T., Rashidi, S., Rahmani, A.M.: Deep learning-based intrusion detection systems: a systematic review. IEEE Access **9**, 101574–101599 (2021)
15. Kuzlu, M., Catak, F.O., Cali, U., Catak, E., Guler, O.: Adversarial security mitigations of mmWave beamforming prediction models using defensive distillation and adversarial retraining. Int. J. Inf. Secur. **22**(2), 319–332 (2022)
16. Vitorino, J., Praça, I., Maia, E.: SoK: Realistic adversarial attacks and defenses for intelligent network intrusion detection. Comput. Secur. **134**, 103433 (2023)
17. Alhussien, N., Aleroud, A., Melhem, A., Khamaiseh, S.Y.: Constraining adversarial attacks on network intrusion detection systems: transferability and defense analysis. IEEE Trans. Netw. Serv. Manag. (2024). https://doi.org/10.1109/TNSM.2024.3357316
18. Liu, Y., Xu, L., Yang, S., Zhao, D., Li, X.: Adversarial sample attacks and defenses based on LSTM-ED in industrial control systems. Comput. Secur. **140**, 103750 (2024)
19. Sarker, I.H.: Machine learning: algorithms, real-world applications and research directions. SN computer science **2**(3), 160 (2021)
20. Darwish, A., Hassanien, A.E., Das, S.: A survey of swarm and evolutionary computing approaches for deep learning. Artif. Intell. Rev. **53**, 1767–1812 (2020)
21. Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. SIAM Rev. **60**(2), 223–311 (2018)
22. Mayer, R., Jacobsen, H.A.: Scalable deep learning on distributed infrastructures: Challenges, techniques, and tools. ACM Computing Surveys (CSUR) **53**(1), 1–37 (2020)
23. Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., Gao, R.X.: Deep learning and its applications to machine health monitoring. Mech. Syst. Signal Process. **115**, 213–237 (2019)
24. Thompson, N. C., Greenewald, K., Lee, K., & Manso, G. F. (2020). The computational limits of deep learning. arXiv preprint arXiv:2007.05558.
25. Biermann, E., Cloete, E., Venter, L.M.: A comparison of intrusion detection systems. Comput. Secur. **20**(8), 676–683 (2001)
26. Depren, O., Topallar, M., Anarim, E., Ciliz, M.K.: An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Syst. Appl. **29**(4), 713–722 (2005)
27. Molina-Coronado, B., Mori, U., Mendiburu, A., Miguel-Alonso, J.: Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process. IEEE Trans. Netw. Serv. Manage. **17**(4), 2451–2479 (2020)
28. Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., Ahmad, F.: Network intrusion detection system: a systematic study of machine learning and deep learning approaches. Trans. Emerg. Telecommun. Technol. **32**(1), e4150 (2021)
29. Martins, I., Resende, J.S., Sousa, P.R., Silva, S., Antunes, L., Gama, J.: Host-based IDS: a review and open issues of an anomaly detection system in IoT. Futur. Gener. Comput. Syst. **133**, 95–113 (2022)
30. Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., Faruki, P.: Network intrusion detection for IoT security based on learning techniques. IEEE Commun. Surv. Tutor. **21**(3), 2671–2701 (2019)
31. Dutt, I., Borah, S., Maitra, I.K.: Immune system based intrusion detection system (IS-IDS): A proposed model. IEEE Access **8**, 34929–34941 (2020)
32. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.
33. Deldjoo, Y., Noia, T.D., Merra, F.A.: A survey on adversarial recommender systems: from attack defense strategies to generative adversarial networks. ACM Comput. Surv. (CSUR) **54**(2), 1–38 (2021)
34. Alatwi, H. A., & Morisset, C. (2021). Adversarial machine learning in network intrusion detection domain: A systematic review. arXiv preprint arXiv:2112.03315
35. Hernandez-Ramos, J. L., Karopoulos, G., Chatzoglou, E., Kouliaridis, V., Marmol, E., Gonzalez-Vidal, A., & Kambourakis, G. (2023). Intrusion Detection based on Federated Learning: a systematic review. arXiv preprint arXiv:2308.09522.
36. Papamartzivanos, D., Mármol, F.G., Kambourakis, G.: Introducing deep learning self-adaptive misuse network intrusion detection systems. IEEE Access **7**, 13546–13560 (2019)
37. Ferdowsi, A., & Saad, W. (2019, December). Generative adversarial networks for distributed intrusion detection in the internet of things. In 2019 IEEE global communications conference (GLOBECOM) (pp. 1–6). IEEE.
38. Caminero, G., Lopez-Martin, M., Carro, B.: Adversarial environment reinforcement learning algorithm for intrusion detection. Comput. Netw. **159**, 96–109 (2019)
39. Qiu, H., Dong, T., Zhang, T., Lu, J., Memmi, G., Qiu, M.: Adversarial attacks against network intrusion detection in IoT systems. IEEE Internet Things J. **8**(13), 10327–10335 (2020)
40. Alhajjar, E., Maxwell, P., Bastian, N.: Adversarial machine learning in network intrusion detection systems. Expert Syst. Appl. **186**, 115782 (2021)
41. Anthi, E., Williams, L., Rhode, M., Burnap, P., Wedgbury, A.: Adversarial attacks on machine learning cybersecurity defences in industrial control systems. J. Inf. Secur. Appl. **58**, 102717 (2021)
42. Chatzoglou, E., Kambourakis, G., Kolias, C.: Empirical evaluation of attacks against IEEE 802.11 enterprise networks: the AWID3 dataset. IEEE Access **9**, 34188–34205 (2021)
43. Smiliotopoulos, C., Kambourakis, G., Barbatsalou, K.: On the detection of lateral movement through supervised machine learning and an open-source tool to create turnkey datasets from Sysmon logs. Int. J. Inf. Secur. **22**, 1893–1919 (2023)
44. Yu, J., Ye, X., Li, H.: A high precision intrusion detection system for network security communication based on multi-scale convolutional neural network. Futur. Gener. Comput. Syst. **129**, 399–406 (2022)
45. Chatzoglou, E., Kambourakis, G., Kolias, C., Smiliotopoulos, C.: Pick quality over quantity: expert feature selection and data preprocessing for 802.11 intrusion detection systems. IEEE Access **10**, 64761–64784 (2022)
46. Khan, A.R., Kashif, M., Jhaveri, R.H., Raut, R., Saba, T., Bahaj, S.A.: Deep learning for intrusion detection and security of Internet

of things (IoT): current analysis, challenges, and possible solutions. Secur. Commun. Netw. (2022). https://doi.org/10.1155/2022/4016073

47. Chatzoglou, E., Kambourakis, G., Smiliotopoulos, C., Kolias, C.: Best of both worlds: Detecting application layer attacks through 802.11 and non-802.11 features. Sensors **22**(15), 5633 (2022)
48. Usmani, M., Anwar, M., Farooq, K., Ahmed, G., & Siddiqui, S. (2022). Predicting ARP spoofing with machine learning. In 2022 international conference on emerging trends in smart technologies (ICETST) (pp. 1–6). IEEE.
49. Ramachandran, V., & Nandi, S. (2005). Detecting ARP spoofing: an active technique. In: Information systems security: first international conference, ICISS 2005, Kolkata, India, December 19–21, 2005. Proceedings 1 (pp. 239-250). Springer Berlin Heidelberg
50. Pawlicki, M., Choraś, M., Kozik, R.: Defending network intrusion detection systems against adversarial evasion attacks. Futur. Gener. Comput. Syst. **110**, 148–154 (2020)
51. Taheri, R., Javidan, R., Pooranian, Z.: Adversarial android malware detection for mobile multimedia applications in IoT environments. Multimed. Tools Appl. **80**, 16713–16729 (2021)
52. Yang, Y., Zheng, K., Wu, B., Yang, Y., Wang, X.: Network intrusion detection based on supervised adversarial variational auto-encoder with regularization. IEEE access **8**, 42169–42184 (2020)
53. Qureshi, A.U.H., Larijani, H., Yousefi, M., Adeel, A., Mtetwa, N.: An adversarial approach for intrusion detection systems using jacobian saliency map attacks (jsma) algorithm. Computers **9**(3), 58 (2020)
54. Debicha, I., Bauwens, R., Debatty, T., Dricot, J.M., Kenaza, T., Mees, W.: TAD: Transfer learning-based multi-adversarial detection of evasion attacks against network intrusion detection systems. Futur. Gener. Comput. Syst. **138**, 185–197 (2023)
55. Roshan, K., Zafar, A., Haque, S.B.U.: Untargeted white-box adversarial attack with heuristic defence methods in real-time deep learning based network intrusion detection system. Comput. Commun. **218**, 97–113 (2023)
56. Alotaibi, A., Rassam, M.A.: Enhancing the sustainability of deep-learning-based network intrusion detection classifiers against adversarial attacks. Sustainability **15**(12), 9801 (2023)
57. Paya, A., Arroni, S., García-Díaz, V., Gómez, A.: Apollon: a robust defense system against adversarial machine learning attacks in intrusion detection systems. Comput. Secur. **136**, 103546 (2024)
58. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp **1**, 108–116 (2018)
59. Gudivada, V., Apon, A., Ding, J.: Data quality considerations for big data and machine learning: going beyond data cleaning and transformations. Int. J. Adv. Softw. **10**(1), 1–20 (2017)
60. Elmasry, W., Akbulut, A., Zaim, A.H.: Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic. Comput. Netw. **168**, 107042 (2020)
61. Rm, S.P., Maddikunta, P.K.R., Parimala, M., Koppu, S., Gadekallu, T.R., Chowdhary, C.L., Alazab, M.: An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture. Comput. Commun. **160**, 139–149 (2020)
62. Li, F., Lai, L., Cui, S.: On the adversarial robustness of LASSO based feature selection. IEEE Trans. Signal Process. **69**, 5555–5567 (2021)
63. Zhong, G., Liu, F., Jiang, J., Chen, C.P.: CauseFormer: interpretable anomaly detection with stepwise attention for cloud service. IEEE Trans. Netw. Serv. Manag. (2023). https://doi.org/10.1109/TNSM.2023.3299846
64. Donkol, A.A.E.B., Hafez, A.G., Hussein, A.I., Mabrook, M.M.: Optimization of intrusion detection using likely point PSO and enhanced LSTM-RNN hybrid technique in communication networks. IEEE Access **11**, 9469–9482 (2023)
65. Alsarhan, A., Alauthman, M., Alshdaifat, E.A., Al-Ghuwairi, A.R., Al-Dubai, A.: Machine Learning-driven optimization for SVM-based intrusion detection system in vehicular ad hoc networks. J. Ambient. Intell. Humaniz. Comput. **14**(5), 6113–6122 (2023)