Soft Computing (2022) 26:10651–10663
https://doi.org/10.1007/s00500-022-06798-2

FOCUS

Intrusion detection in networks using cuckoo search optimization

Muhammad Imran[1] · Sangeen Khan[1] · Helmut Hlavacs[2] · Fakhri Alam Khan[3] · Sajid Anwar[4]

Accepted: 11 January 2022 / Published online: 3 February 2022

Abstract
One of the key problems for researchers and network managers is anomaly detection in network traffic. Anomalies in network traffic might signal a network intrusion, requiring the use of a quick and dependable network intrusion detection system. Intrusion detection systems (IDS) based on artificial intelligence (AI) techniques are gaining the interest of the research community as AI techniques have evolved in recent years. This research proposes a novel method for anomaly detection using artificial neural networks (ANNs) optimized using cuckoo search algorithm. For simulation purposes, the NSL-KDD dataset has been utilized with a 70:30 ratio where 70% of data is used for training and the remaining 30% is used for testing. The proposed model is then evaluated in terms of mean absolute error, mean square error, root-mean-square error, and accuracy. The results of the proposed work are compared with standard methods available in the literature including fuzzy clustering artificial neural network (FC-ANN), intrusion detection with artificial bee colony (ABC), neural network intrusion detection (NNID) system, and selection of relevant feature (SRF). The results clearly show that the proposed method outperforms the listed standard methods.

Keywords Intrusion detection · Artificial neural networks · Cuckoo search

1 Introduction

An intrusion detection system is a hardware device or software application that monitors a system or network of systems for any unusual behaviour such as security attacks. Such systems are utilized in Internet of things (IoT), mobile networks, and big data environments. In case of an attack on the network, an IDS analyses the traffic passing through the network and sends an alert if any abnormal traffic is found. Intrusion detection systems are generally classified according to the architecture and detection model (Debar et al. 1999; Debar 2000). The architecture-based classification refers to the underlying environment where the IDS is utilized. If the target environment of IDS is a single machine, then host-based IDS are utilized. Similarly, if the target environment of IDS is a network of computers such as distributed systems, then network-based IDS is utilized (Singh and Singh 2014). Host-based IDS generally utilizes log files for detection of intrusion, whereas network-based IDS uses network traffic, e.g. network packets. With the shift of computing from hosted environments to distributed computing, network-based IDS is more common in the literature and industry. Depending on the detection model, i.e. how intrusion detection is per-

Communicated by Tiancheng Yang.

S. Khan, H. Hlavacs, F. Alam Khan, S. Anwar contributed equally to this work.

✉ Muhammad Imran
mimran@aup.edu.pk

Sangeen Khan
sangeen.khan94@gmail.com

Helmut Hlavacs
helmut.hlavacs@univie.ac.at

Fakhri Alam Khan
fakhri.khan@kfupm.edu.sa

Sajid Anwar
sajid.anwar@imsciences.edu.pk

[1] Institute of Computer Sciences and Information Technology (ICS/IT), The University of Agriculture, Peshawar, Pakistan

[2] Entertainment Computing, Faculty of Computer Science, University of Vienna, Vienna, Austria

[3] Department of Information and Computer Science, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

[4] Computer Science, IMSciences, Peshawar, Pakistan

formed, there are 2 subtypes of IDS, namely signature-based intrusion detection and anomaly-based intrusion detection.

Signature-based intrusion detection detects intrusions using an existing database. The knowledge base is made up of records of previously recognized incursions. These records contain common patterns found in previously detected intruders. This idea comes from antivirus software, which refers to these sorts of discovered patterns as signatures. Signature-based intrusion detection systems are well suited for scenarios where previously identified intruders need to be detected. For new intruders whose patterns do not match the database, a signature-based intruder detection system cannot achieve results in this scenario and fails. Moreover, with the shift of computing to a distributed environment, the efficiency of such systems suffers in real time (Kumar and Sangwan 2012).

Anomaly-based systems are suitable for the detection of unknown attacks (Garcia-Teodoro et al. 2009). This is achieved by creating profiles that differentiate between normal and abnormal behaviour. The behaviour of the network is trained to either accept a transition or not. If the transition is not accepted, it triggers warnings. The network administrator defines the specifications or conditions for the performance of an acceptable network. Although signature-based IDS is more accurate, precise and efficient in times, the lack of identification of new attacks significantly hinders their usage. Therefore, anomaly-based IDS is gaining interest from researchers and industry alike.

Current research indicates that anomaly-based IDS is developed using machine learning approaches (Krishnaveni et al. 2020). A variety of algorithms such as support vector machine (SVM), nearest neighbour, decision trees, and random forest has been used with different rates of accuracy and speed. With the evolution of machine learning algorithms and their optimization techniques (Khan et al. 2019), intrusion detection using artificial intelligence (AI) techniques is an open area of research. Moreover, research shows that merged or hybrid scheme to solve computational problems gives better results when compared to singular approaches (Imran et al. 2018; Khan et al. 2019; Atefi et al. 2013). In this work, we propose a novel cuckoo search (CS)-based neural network scheme for intrusion detection using the NSL-KDD dataset. The selection of cuckoo search optimization is made because of i) its fast convergence rate, ii) the use of Levy flight instead of random walk, and iii) the limited research utilizing CS for the development of IDS. The objective of this research is to propose a CS-based neural network scheme for the development of IDS. The results of this research are compared with state-of-the-art existing work, namely fuzzy clustering artificial neural network (FC-ANN), intrusion detection with artificial bee colony (IDS-ABC), neural network intrusion detection (NNID) system, and selection of relevant feature (SRF), and the significance of the current work is highlighted.

Key contributions of the proposed work are listed below:

- Constructing a novel intrusion detection system that utilizes CS-based neural network meta-heuristic approach.
- Reducing the mean absolute error, mean square error, and root-mean-square error of the proposed method as compared to the existing models in the literature.
- Comparison of the proposed method in terms of accuracy (correct classification of intrusions) with standard algorithms in the literature.

The rest of the paper is organized as follows. Section 2 presents the importance of machine learning algorithms in the development of IDS. This section highlights the key research work in different areas such as Internet of things (IoT) and big data. Section 3 presents the details of the dataset. Section 4 outlines the methodology of the proposed research as well as the CS algorithm and the proposed method. Section 5 presents result of the research work and comparison with state-of-the-art literature. Section 6 provides concluding remarks along with future directions.

2 Literature review

Traditionally, IDS is signature-based, i.e. they look for certain types of data payloads in the network traffic and make decisions for malicious data and inform the administrator of the network (Hubballi and Suryanarayanan 2014). Due to the lack of identifying new intruders in such systems, anomaly-based systems gained more interest from the research community. With the advent of technologies such as artificial intelligence and machine learning, anomaly-based IDS gained more significance as evident in the literature (Alamiedy et al. 2019; Garcia-Teodoro et al. 2009; Krishnaveni et al. 2020). Such systems make decisions for normal and malicious traffic based on deep learning methods as opposed to signature-based identification. Such systems add intelligence to IDS because they can learn from past behaviour and evolve with new traffic types. In this section, we briefly study the addition of AI techniques into traffic analysis.

Anomaly-based IDS is categorized into three types based on the detection method (statistical, neural networks, and data mining) as shown in Fig. 1. Statistical-based IDS takes a simple approach for the identification of intruders. These models define normal network activity, and all the traffic that falls outside the boundaries of normal network traffic is flagged as anomalous traffic. These systems learn by observing activity, and any event that deviates from a normal activity is considered an intrusion. Generally, a threshold value is set

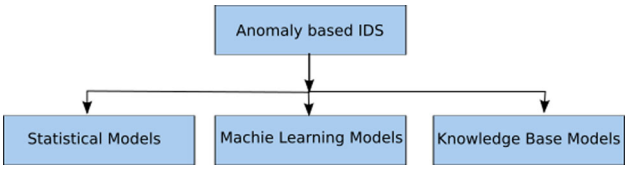


Fig. 1 Anomaly-based IDS detection methods

for anomaly detection in such methods. Statistical models may use univariate, multivariate, and time series methods for defining the threshold value and anomaly detection.

Knowledge-based IDS is also referred to as data mining-based IDS or expert systems (Aldweesh et al. 2020; Batista et al. 2019). These IDS utilizes varieties of approaches such as association, sequence rules, classification, clustering, and forecasting for the identification of intruders. In this method, a pre-existing database reflects the normal traffic. This database is analysed to produce new information such as hidden patterns using any of the mentioned approaches.

Some of the current IDS can identify attacks and intruders based on heuristics (Zhou et al. 2004; Rao et al. 2017; Aghdam and Kabiri 2016). In such a system, the IDS is configured with a set of characteristics that might define an attack. When the traffic passes through the network, the heuristics examines the traffic and decides if an attack is taking place. These systems extract knowledge from a huge amount of data and look for any deviation from normal trends or behaviour. Machine learning techniques for IDS are also called black box modelling approaches because input data are checked for any data patterns using a set of rules or transfer functions.

Anomaly-based IDS has extensively applied machine learning approaches for the detection of intruders. Some of the common machine learning methods to discover knowledge from a dataset include decision trees, neural networks, genetic algorithms, and nearest neighbour (Wang et al. 2017; Maseer et al. 2021). A recent review from Ferdiana et al. (2020) indicates the significance of the machine learning methods in intrusion detection systems. A hybrid approach combining neural networks and classification algorithms is also gaining significance from the research community.

IDS is essential to keep us safe from various intruders. Every transaction using the Internet is prone to different types of attacks such as race, DDoS, and routing. The application area of IDS includes Internet of things, sensor networks, big data, and mobile communication. All these application areas use machine learning techniques for developing IDS. Since this research work is focused on the use of machine learning approaches, we provide a summary of the recent research work in the key areas.

Wang et al. (2011) have proposed integrated IDS for cluster-based wireless sensor networks. They have adopted a mechanism where rule-based and misuse-based methods of IDS are implemented at cluster head, sink node and sensor node. The KDD CUP 99 (Tavallaee et al. 2009) dataset was utilized for training and testing purposes of the IDS. The performance of the proposed model is measured in terms of detection rate, accuracy, and false positive. The result shows that their system has achieved a 90.96% detection rate, 2.06% false positive rate, and 91.26% accuracy. Changing the threshold value for anomaly detection, the accuracy is increased to 97.31%. We believe that accuracy can be further increased using the advancement in machine learning approaches such as deep learning models.

More recently, Gao et al. (2019) have proposed an IDS for vehicular ad hoc network using random forecast classification algorithms. The focus of this research is on distributed denial of service (DDoS) attacks because of their importance in VANETs. They have utilized NSL-KDD and UNSW-NB15 datasets with 99.95% and 98.75% accuracy. Their research work can be enhanced by including more attacks found in the dataset, and therefore, accuracy of results may vary. Farahnakian and Heikkonen (2018) have proposed a deep learning model that uses an auto-encoder scheme for attack detection using the KDD-CUP 99 dataset. Their research reports a 94.71% accuracy for attack identification and has been compared with existing research. Similarly, Ali et al. (2018) have proposed a fast learning algorithm using particle swarm optimization using the KDD 99 dataset. The accuracy of their model is reported at 98.92%. However, the model suffers from high complexity and is not suitable for devices with power capacity problems.

The research work by Alamiedy et al. (2019) has used grey wolf optimization algorithm for the detection of anomalies in the NSL-KDD dataset. The result of their research work claims 93.64% accuracy for DoS attacks, 91.02% accuracy for probe attacks, 57.72% accuracy for R2L attacks, and 53.7% accuracy for U2R attacks. The variation in the result is because of the fitness function in the proposed algorithm. With the increasing amount of data in today's world, IDS for big data is also becoming important. The research work by Othman et al. (2018) proposes Spark-Chi-SVM (apache spark for big data, Chisq for feature selection, and SVM for classification) model for intrusion detection in the domain of big data analytics. Their experiment result shows a 99.55% accuracy for the proposed model along with a high speed of detection. Their model supports only binary classification and can be extended to support multiclass classification. Some of the recent survey reports (da Costa et al. 2019; Saranya et al. 2020; Almseidin et al. 2017; Baraneetharan 2020; Axelsson 2000) show the importance of machine learning algorithms in the development of IDS.

After a careful literature review, it is evident that machine learning approaches are the recent trend in intrusion detection systems. Different models like grey wolf optimization, Spark-Chi SVM, integrated IDS, amongst others, have been

developed in the literature (Ahmad et al. 2021). These models show various rates of accuracy and error values (MAE, MSE). Hereby, this work introduces a novel IDS based on cuckoo search optimization to increase the value of accuracy and reduce the different error rates.

3 Dataset selection

This research uses the NSL-KDD dataset for the classification problem. The dataset is available for offline evaluation of IDS and is derived from the KDD Cup 99 dataset which is one of the most used datasets available to the research community. The problems in KDD Cup 99, i.e. redundancy and skewness of data as discussed in Tavallaee et al. (2009), lead to the development of the NSL-KDD dataset. The NSL-KDD dataset is divided into four subtypes, namely KDDTest+, KDDTrain+, KDDTest-21, and KDDTrain+20Percent. In this research, we utilize KDDTrain+ and KDDTest+ simply referred to as training and testing datasets. It is important to note that records found in KDDTest-21 and KDDTrain+20Percent also exist in our training and testing dataset.

The dataset comprises 43 features where 41 features are network traffic input and the remaining two represent labels for network packet (attack vs normal) and score (severity of the attack). The attacks in the dataset are divided into 4 different classes, namely remote to local (R2L), probe, user to root (U2R), and denial of service (DoS) attacks. DoS is a powerful attack in which a hacker makes a system unavailable to the end user, whereas R2L, probe, and U2R attacks try to infiltrate the system undetected. The distribution of the individual attacks and normal data as shown in Table 1 is according to the real-life traffic where DoS attacks are more common than probe or U2R. However, the testing dataset shows some skewed distribution in the case of R2L attacks. The DoS, probe, U2R, and R2L are divided into 11, 6, 7, and 15 subclasses, respectively.

Features of the NSL-KDD dataset are divided into four categories, namely basic, content, time-based, and host-based features. Basic features (1 to 9) contain the basic information about the network packet. Content features (10 to 22) contain information about the payload. Time-based features (23 to 31) are counts and rates of attempts that are made to connect with the host, whereas host-based features (32 to 41) represent the number of connections made to the same host. This research chose NSL-KDD because i) the dataset has no duplicate records (in test sets), ii) the dataset contains a sufficient number of records, and iii) the number of features and categories of attacks in the dataset makes it an effective benchmark in research to compare different types of IDS.

4 Proposed method

In this section, we present the proposed method which is divided into three modules. Firstly, we present the necessary pre-processing on the NSL-KDD dataset. Secondly, we present the basics of the cuckoo search algorithm and the reason to choose it. Lastly, we discuss the ANN and its classification using the proposed algorithm. Figure 2 presents the key modules of this research which are explained in detail in the following subsections.

4.1 Data pre-processing

Real-world data are sometimes inconsistent and lack certain behaviour or trends. Therefore, pre-processing the original data to address this issue is a proven research method. Pre-processing is just like data mining where the goal is to transform the data into an understandable format. This step is also important to improve the quality of data such

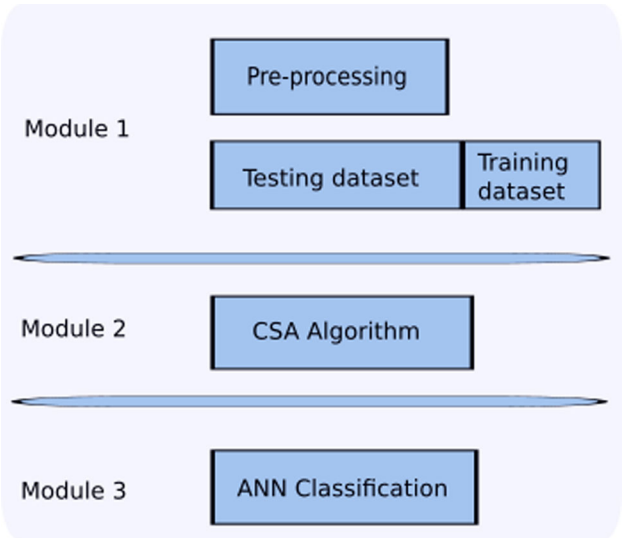


Fig. 2 Key modules of the proposed method

Table 1 Distribution of normal data and attacks in NSL-KDD

	Normal (%)	DoS (%)	Probe (%)	U2R (%)	R2L (%)	Total
Training set	53	37	9.11	0.04	0.85	100
Testing set	43	33	11	0.9	12.1	100

as removing noise and redundant information. Data cleaning, transformation, integration, and reduction are some of the key methods used in data pre-processing. In this work, we have performed data transformation and data normalization pre-processing steps to align the data according to the proposed method.

4.1.1 Data transformation from string to numerical values

The values of the feature, namely protocol type, service, flag, and class, in the NSL-KDD dataset are of string data type as shown in Fig. 3. However, the proposed algorithm only supports numerical values. Therefore, we need to convert the string values in the dataset to numerical ones. For this purpose, we have applied the transformations shown in Fig. 4. Figure 5 represents a snapshot of the modified dataset.

4.1.2 Data normalization

Normalization is a key step that is applied in pre-processing to prepare the data for machine learning algorithms. The objective is to change the numeric values in the dataset to a common scale. Normalization becomes important when values of features in the dataset have a different range. The previous step has already converted the dataset into numeric values; however, the values of different features are not in range as shown in Fig. 4. Therefore, we normalized the dataset as per Eq. 1 and the result is shown in Fig. 6.

$$Xnorm = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

where Xnorm is the normalized values and x represent the original values in the dataset.

4.1.3 Training and testing dataset

In this work, we have used a general rule of thumb to divide the dataset into training and testing where 70% of the data is used for training purposes and 30% of data is used for testing purposes.

4.2 Cuckoo search

Cuckoo search was initially proposed by Xin-She Yang and Suash Deb in 2009 to solve optimization problems (Yang and Deb 2010). The algorithm is inspired by the behaviour of the cuckoos birds which lays eggs in the nests of the other host birds (host birds are of other species). This creates a direct conflict between the cuckoo and the host bird where the host bird can discover the cuckoo's egg with a probability P between 0 and 1. If a host bird discovers cuckoos eggs in its nest, it will either throw away the egg or abandon the nest. These two phenomena are the basics of the cuckoo search algorithm. The main characteristics of CS are:

- One egg is laid by a cuckoo in a randomly chosen nest. This represents a possible solution to an optimization problem.
- Nest with the best eggs is passed to the next iteration, i.e. the best solutions are retained.
- The number of potential nests is fixed, and each egg deposited by a cuckoo has a chance $P_a \in \{0, 1\}$ of being

Fig. 3 Snapshot of original dataset

duration	protocol_t	service	flag	src_bytes	dst_bytes	land	wrong_fra	urgent	hot
13	tcp	telnet	SF	118	2425	0	0	0	0
0	udp	private	SF	44	0	0	0	0	0
0	tcp	telnet	S3	0	44	0	0	0	0
0	udp	private	SF	53	55	0	0	0	0
0	tcp	private	SH	0	0	0	0	0	0
0	tcp	http	SF	54540	8314	0	0	0	2
0	tcp	imap4	REJ	0	0	0	0	0	0
7570	tcp	telnet	SF	0	44	0	0	0	0
0	udp	private	SF	56	52	0	0	0	0
0	tcp	ftp_data	SF	192	0	0	0	0	0
0	tcp	other	REJ	0	0	0	0	0	0
0	tcp	other	REJ	0	0	0	0	0	0
0	tcp	telnet	SF	21	97	0	0	0	0
0	udp	private	SF	45	0	0	0	0	0
0	tcp	telnet	S3	0	44	0	0	0	0
0	tcp	imap4	REJ	0	0	0	0	0	0
0	tcp	http	S0	0	0	0	0	0	0
0	tcp	ctf	S0	0	0	0	0	0	0
0	tcp	telnet	S3	0	44	0	0	0	0
0	udp	private	SF	1	1	0	0	0	0
0	tcp	telnet	S3	0	44	0	0	0	0

B	C	D	E	F	G	H	I	J	K	L	M	N	O
normal	1			Z39_50	13.9			sunrpc	14.17			efs	16.2
anomaly	0			smtp	13.15			discard	14.18			nntp	16.3
tcp	11.1			uucp_path	13.11			auth	14.4			pop_3	16.4
udp	11.2			domain_u	13.12			bgp	14.19			printer	16.5
icmp	11.3			finger	13.13			gopher	15.1			pop_2	16.6
sf	12.1			csnet_ns	13.14			whois	15.2			netbios_s	16.7
so	12.2			hostname	13.16			ecr_i	15.3			sql_net	16.8
rej	12.3			supdup	13.17			login	15.4			red_i	16.9
RSTR	12.4			telnet	13.18			daytime	15.5			pm_dump	16.11
RSTO	12.5			iso_tsap	14.1			netbios_d	15.6			urh_i	16.12
SH	12.6			klogin	14.2			kshell	15.7			tim_i	16.13
OTH	12.7			imap4	14.3			ntp_u	15.8				
S1	12.8			rje	14.5			vmnet	15.9				
S2	12.9			ldap	14.6			systat	15.11				
RSTOS0	12.11			ftp	14.7			uucp	15.12				
S3	12.12			exec	14.8			time	15.13				
ftp_data	13.1			nnsp	14.9			link	15.14				
other	13.2			courier	14.11			IRC	15.15				
remote_jo	13.3			ctf	14.12			echo	15.16				
private	13.4			ssh	14.13			X11	15.17				
netbios_n	13.5			urp_i	14.14			netstat	15.18				
http	13.6			name	14.15			shell	15.19				
eco_i	13.7			mtp	14.16			domain	16.1				
mtp	13.8												

Fig. 4 Guideline to convert textual data into numerical data

duration	protocol_t	service	flag	src_bytes	dst_bytes	land	wrong_fra	urgent	hot
(Ctrl)	11.1	13.1	12.1	491	0	0	0	0	0
0	11.2	13.2	12.1	146	0	0	0	0	0
0	11.1	13.4	12.2	0	0	0	0	0	0
0	11.1	13.6	12.1	232	8153	0	0	0	0
0	11.1	13.6	12.1	199	420	0	0	0	0
0	11.1	13.4	12.3	0	0	0	0	0	0
0	11.1	13.4	12.2	0	0	0	0	0	0
0	11.1	13.4	12.2	0	0	0	0	0	0
0	11.1	13.3	12.2	0	0	0	0	0	0
0	11.1	13.4	12.2	0	0	0	0	0	0
0	11.1	13.4	12.3	0	0	0	0	0	0
0	11.1	13.4	12.2	0	0	0	0	0	0
0	11.1	13.6	12.1	287	2251	0	0	0	0
0	11.1	13.1	12.1	334	0	0	0	0	0
0	11.1	14.15	12.2	0	0	0	0	0	0
0	11.1	13.5	12.2	0	0	0	0	0	0
0	11.1	13.6	12.1	300	13788	0	0	0	0
0	11.3	13.7	12.1	18	0	0	0	0	0
0	11.1	13.6	12.1	233	616	0	0	0	0
0	11.1	13.6	12.1	343	1178	0	0	0	0
0	11.1	14.16	12.2	0	0	0	0	0	0
0	11.1	13.4	12.2	0	0	0	0	0	0

Fig. 5 Snapshot of modified dataset

duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_frag	urgent	hot	
0	1.77E-07	2.10E-07	1.93E-07	3.34E-07	1.54E-06	0	0	0		
0	1.78E-07	2.13E-07	1.93E-07	7.16E-07	0	0	0	0		
0	0	2.10E-07	1.93E-07	0	7.00E-07	0	0	0		
0	1.77E-07	2.28E-07	1.96E-07	0	0	0	0	0		
0	1.77E-07	2.16E-07	1.94E-07	0	0	0	0	0		
0	1.77E-07	2.25E-07	1.94E-07	0	0	0	0	0		
0	1.77E-07	2.10E-07	1.93E-07	0	7.00E-07	0	0	0		
0	1.78E-07	2.13E-07	1.93E-07	1.59E-08	1.59E-08	0	0	0		
0	1.77E-07	2.10E-07	1.93E-07	0	7.00E-07	0	0	0		
0	1.78E-07	2.10E-07	1.93E-07	1.59E-08	1.59E-08	0	0	0		
0	1.77E-07	2.10E-07	1.93E-07	3.82E-06	9.85E-06	0	0	0		
1.59E-07	1.77E-07	2.61E-07	1.93E-07	4.30E-07	1.48E-06	0	0	0		
0	1.77E-07	2.16E-07	1.94E-07	0	0	0	0	0		
0	1.78E-07	2.13E-07	1.93E-07	6.69E-07	0	0	0	0		
0	1.77E-07	2.16E-07	1.93E-07	0.000868	0.000132	0	0	0	3.18E-	
4.49E-06	1.77E-07	2.34E-07	1.93E-07	2.48E-06	9.44E-06	0	0	0	3.18E-	
0	1.78E-07	2.09E-07	1.93E-07	6.84E-07	6.84E-07	0	0	0		
0	1.77E-07	2.56E-07	1.96E-07	0	0	0	0	0		
0	1.80E-07	2.18E-07	1.93E-07	1.27E-07	0	0	0	0		
7.96E-08	1.77E-07	2.61E-07	1.93E-07	5.09E-07	1.48E-06	0	0	0		

Fig. 6 Snapshot of the normalized dataset

discovered and abandoned. This means that for each iteration t, a fraction P_a of the entire population will be changed.

The process of CS is divided into three distinct operations, i.e. (i) Levy flight, (ii) replace nests with new solutions, and (iii) greedy selection. The main goal of Levy flight is to generate new solutions. Once a new solution is generated, its fitness value is checked for any improvement to the existing solutions. In case of improvement, some nests are replaced with new solutions. Finally, we can preserve the best value until the goal is achieved.

4.3 Why cuckoo search

- The cuckoo search algorithm improves performance by using Levy flight instead of random walk. Many animals and insects have been seen to display the typical Levy flying characteristics. A Levy flight is a random walk whose step lengths are controlled by a heavy-tailed probability distribution, as shown in Eq. 2. The distance from the random walk's origin tends to a stable distribution after a large number of steps. It is evident from Gandomi et al. (2013) that random walk is better performed by Levy flight. Therefore, we have chosen cuckoo search algorithm in this research work because it provides us with a faster convergence rate.

$$X_i(g+1) = X_i(g) + \alpha \oplus levy(\lambda) \tag{2}$$

where g indicates current generation of solution, α represents the step size, $\oplus$ indicates entrywise multiplication, and $levy(\lambda)$ is Levy exponent.

- Since this research proposes an intrusion detection system, therefore, the complexity of the underlying algorithm is very important. Algorithms with high computational complexity require more time and resources which is not always feasible. This research uses cuckoo search because the number of parameters that must be specified in the initial search is quite low, and it can naturally and efficiently deal with multimodal issues (Gandomi et al. 2013).

4.4 Cuckoo search artificial neural networks (CSANN)

Artificial neural networks (ANNs) are computing systems inspired by the neural networks of the human brain. These networks are made up of interconnected units called neurons. Each neuron-to-neuron connection can send a signal from one neuron to the next. The signal is processed by the receiving neuron, which then sends messages to downstream neurons. Neurons are typically structured in layers, with each layer performing a particular type of transformation on the given data. The signal flows from the first layer, referred to

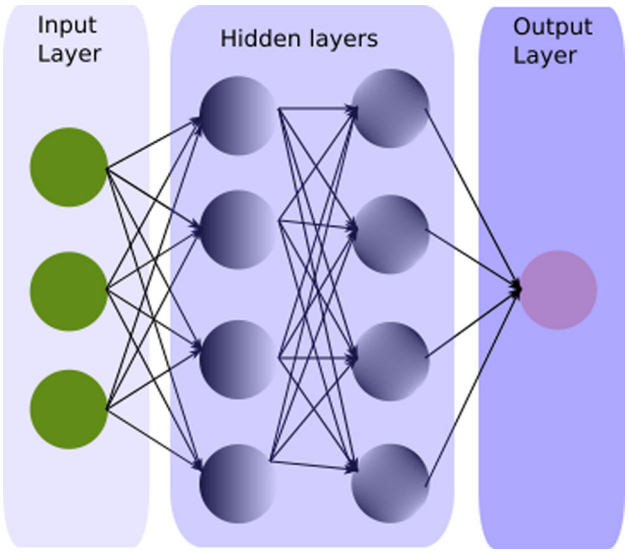


Fig. 7 Basic architecture of ANN

as the input layer, to the final layer, referred to as the output layer. All the layers between the input and output layer are called hidden layers. The illustration of a basic ANN is depicted in Fig. 7 where we have one input layer, more than one hidden layers, and one output layer. ANNs are impressive in solving optimization problems; however, they suffer from slow convergence because of initial random weights. Therefore, algorithms such as cuckoo search show faster convergence because of the possibility to initialize weight.

Each best nest represents a possible solution in the proposed method where the initial weights and corresponding biases are selected for ANN optimization. The quality of the solution is determined by the weight optimization issue and the population size. The solution works in two phases. Firstly, CS is used to initialize the best weight and biases in the first cycle. Secondly, using back propagation, the weights are compared with the best solution. This process is repeated where CS updates the weights with the best possible solution and searches for optimized weights until the last cycle. Figure 8 depicts the flow of the steps in the proposed method where (i) ANN is initialized randomly with population of n nest, (ii) a new solution (nest) is obtained using Levy flight and its fitness is evaluated, and (iii) with each iteration, new solution is compared with old solution until the best solution is obtained.

5 Results and discussion

The effectiveness of the proposed method, i.e. CSA-ANN, is evaluated using the NSL-KDD dataset. For this purpose, we have used an evaluation metric consisting of four key parameters, i.e. mean absolute error (MAE), mean square error (MSE), root-mean-square error (RMSE), and accuracy.

5.1 MAE

MAE is the sum of positive errors for all values. It is calculated by finding the difference between the actual value and predicted value and taking their absolute value. The positive value of all the errors is taken, and the mean is calculated using Eq. 3

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{3}$$

where y_i is the actual value and x_i is the predicted value and n represents the total number of data points in the dataset.

Figure 9 shows MAE result and its convergence during the experiments for the proposed method for both the testing and training datasets. MAE starts from 1.2323 and consistently decreases down to 0.011163 after 500 epochs as shown in the MAE graph for the testing dataset. After 500 to 1000 epochs, the change in MAE value is negligible as shown with a straight line in the graph. The final calculated value for MAE is 0.0097501 in our experiment. Similarly, the calculated value of MAE on the testing dataset starts from 0.07123 and then consistently decreases down to 0.001244 after 120 epoch. The value of MAE stays almost the same until 1000 epochs.

5.2 MSE

MSE is calculated by squaring the difference between the estimated and actual value and then finding the average as shown in Equation 4. MSE closer to zero is considered the best value when comparing multiple models and is one of the most used evaluation parameters in neural networks.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i)^2 \tag{4}$$

where y_i is the actual value and x_i is the predicted value and n represents the total number of data points in the dataset.

Figure 10 shows the simulation result of MSE for both training and testing datasets. The initial calculated value of MSE is 0.751 on the testing dataset. After early fluctuation, MSE starts to decrease and converge to 0.01104 at 1000 epoch. The value of MSE decreases after some epoch because at each epoch, ANN is trained for better results with different weight values. Similarly, the initial value of MAE on the training dataset is recorded at 0.99 and it decreases after few simulations. The value of MAE stays almost constant after 50 epochs at 0.00577941. The similarity in the results for

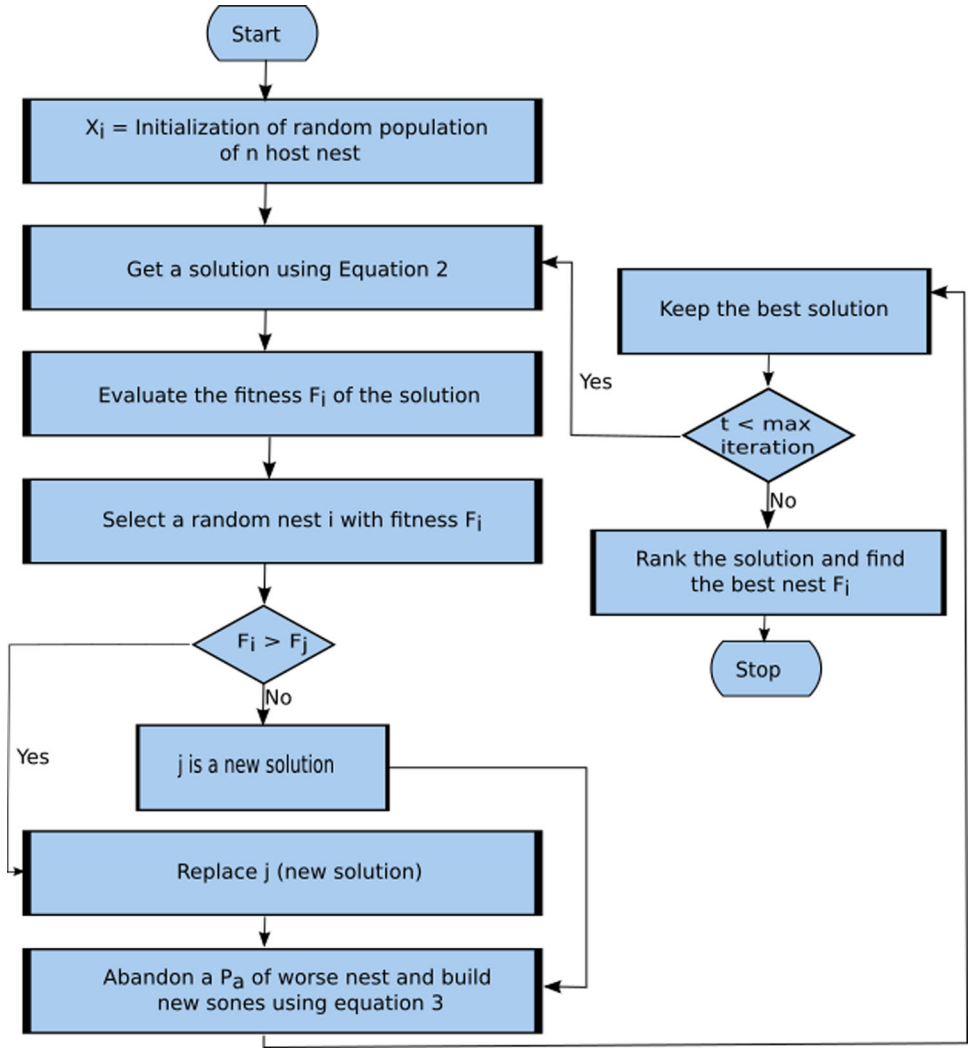
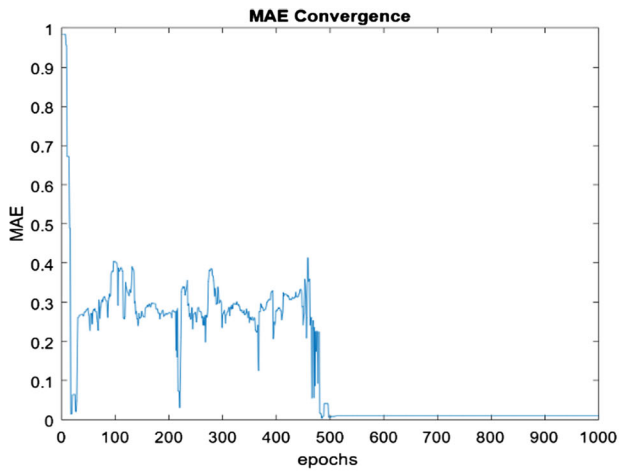
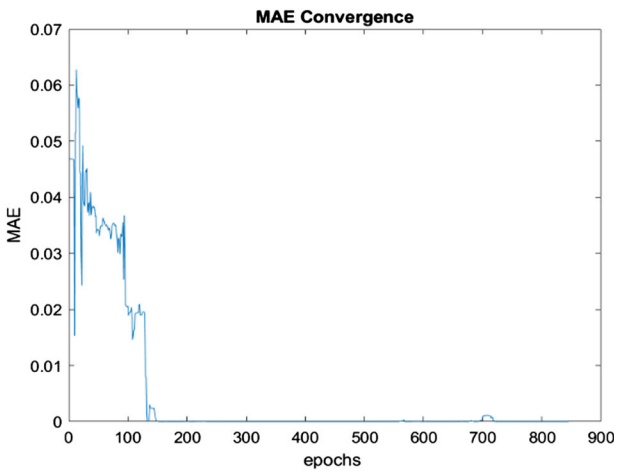


Fig. 8 The flow of the proposed CSANN method

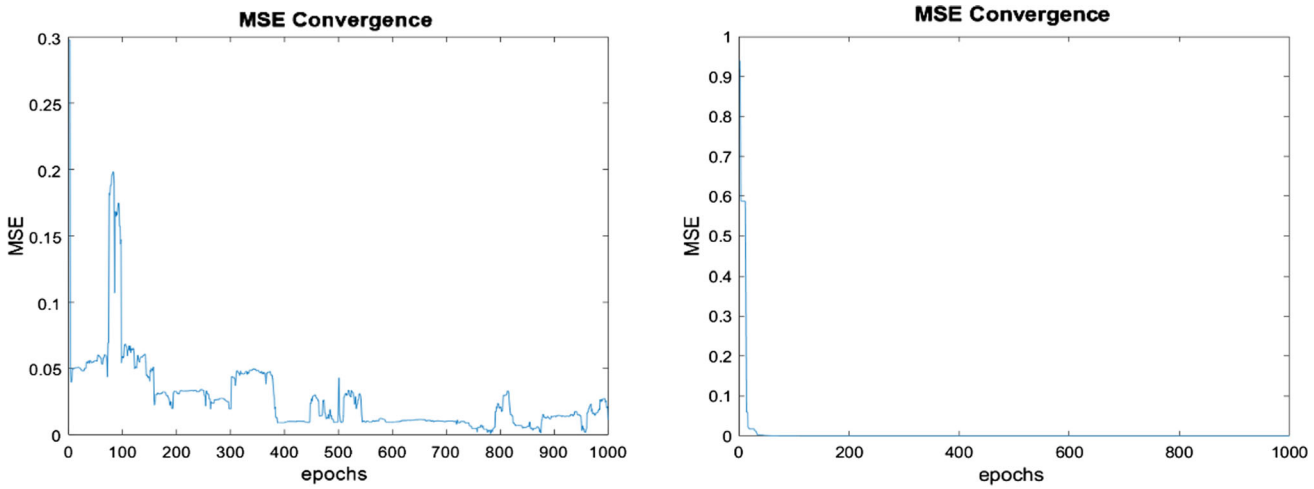


(a) Testing Dataset



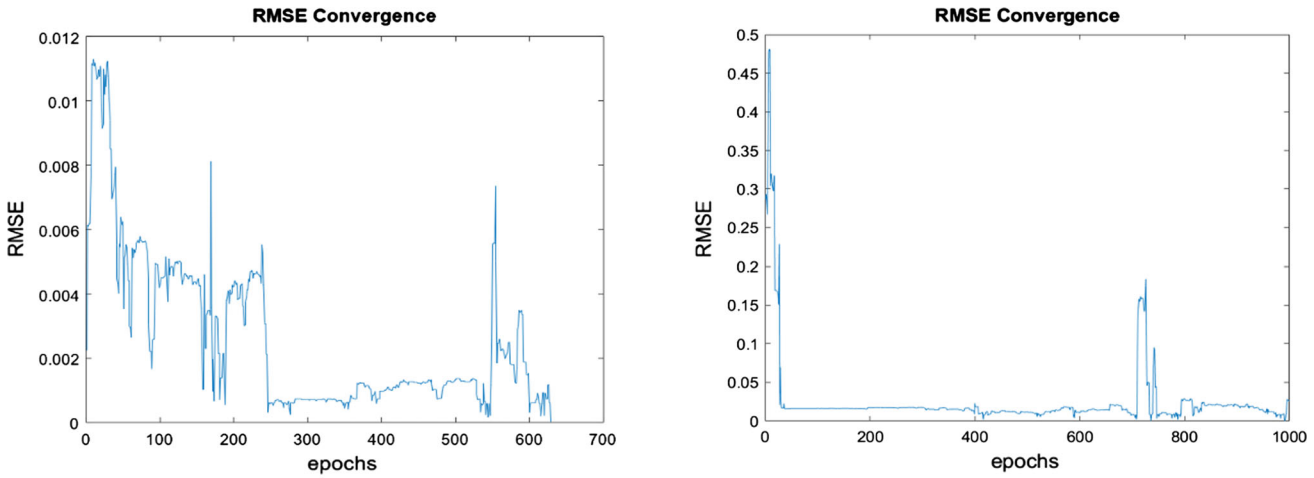
(b) Training Dataset

Fig. 9 MAE convergence for testing and training datasets



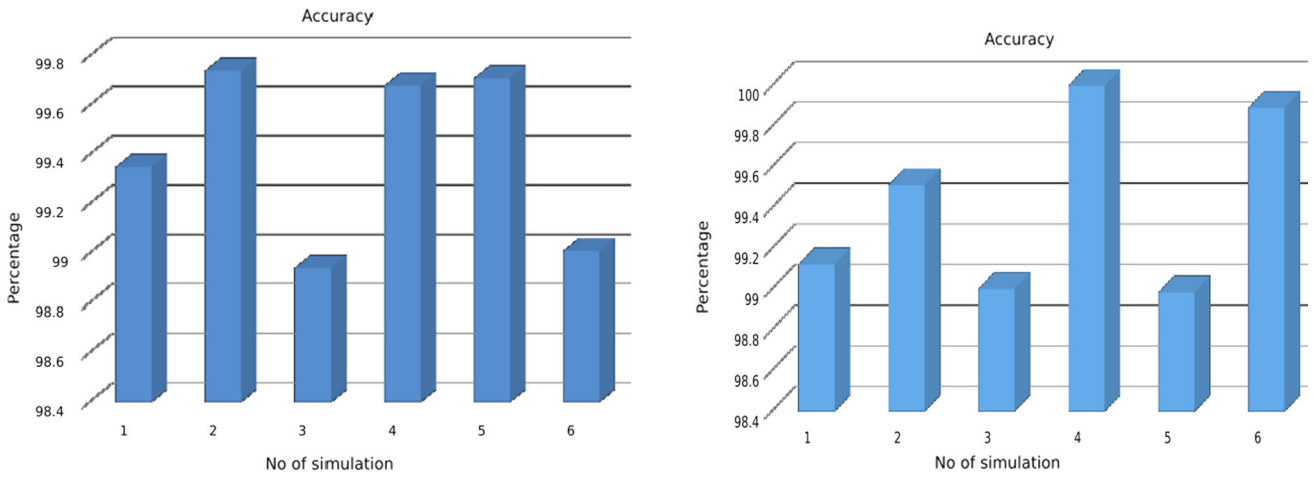
(a) Testing Dataset **(b)** Training Dataset

Fig. 10 MSE convergence for testing and training datasets



(a) Testing Dataset **(b)** Training Dataset

Fig. 11 RMSE convergence for testing and training datasets



(a) Testing Dataset **(b)** Training Dataset

Fig. 12 Accuracy for testing and training datasets

both the training and testing dataset validates the proposed model.

5.3 RMSE

The value of RMSE is calculated by taking the square root of MSE as shown in Eq. 5. RMSE is always positive values where zero denotes a perfect fit to the data. Hence, a lower value of RMSE is better than a large value.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \tag{5}$$

where y_i is the actual value and x_i is the predicted value and n represents the total number of data points in the dataset.

Figure 11 shows the result of the RMSE of our proposed scheme on both the testing and training datasets. The result of RMSE on the testing dataset converges to 0.005934 after 650 epochs with some early fluctuations. Similarly, RMSE starts from 0.45547 on the training dataset and converges to 0.0267 after 1000 epochs.

5.4 Accuracy

Accuracy represents the quality of the proposed method in terms of how often the input data are correctly classified. Accuracy is extremely important for intrusion detection systems and is calculated by computing the ratio of correct predictions with incorrect predictions. During the experiment, we calculated the average accuracy for both the testing model and the training model. As shown in Fig. 12, the accuracy of the proposed method varies for the different number of simulations. The average accuracy on the testing dataset is calculated by executing six simulations, and the result stands at 99.35, whereas the average accuracy on the training dataset is 99.41.

5.5 Comparison with standard models

The results of the proposed model in terms of MAE, RMSE, and accuracy are compared with standard models available in the literature including FC-ANN, IDS-ABC, NNID system, and SRF. The comparison is performed for the testing dataset. As shown in Fig. 13, the proposed model outperforms other models in terms of MAE. Similarly, the proposed method shows a lower RMSE value as shown in Fig. 14. The lower values of MAE and RMSE mean that the proposed method has decreased the incorrectly classified instances, i.e. true negative and false negative. Our method also shows higher accuracy standing at 99.35% when compared with the existing models as shown in Fig. 15.

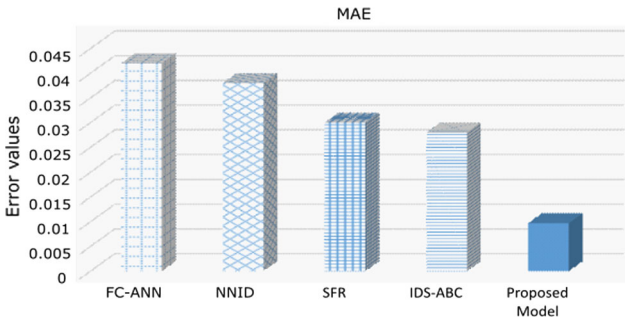


Fig. 13 MAE comparison of the proposed method with standard models

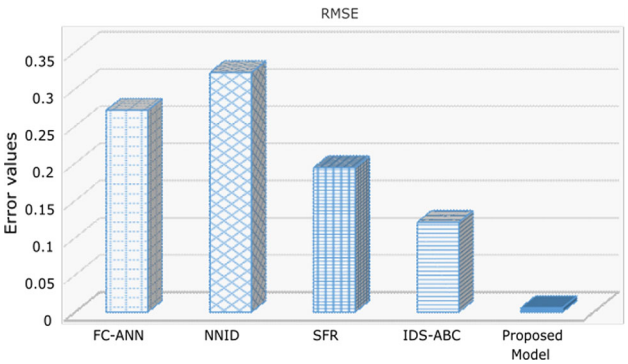


Fig. 14 RMSE comparison of the proposed method with standard models

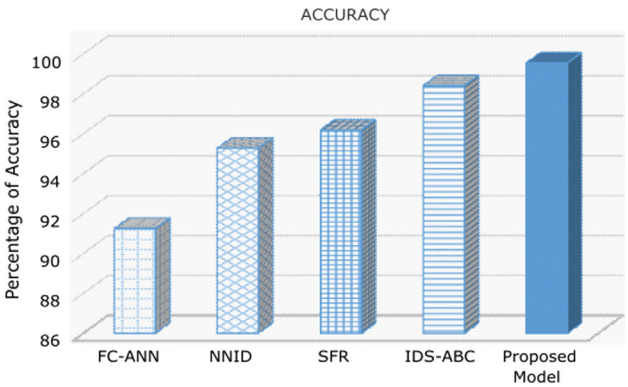


Fig. 15 Accuracy comparison of the proposed method with standard models

Discussion: When compared to established models in the literature, the proposed scheme gives better results in terms of accuracy, MAE, and RMSE. Cuckoo search employs Levy's flight, whereas other algorithm uses random walk. A random walk is not as effective as Levy's flight in exploring an area for a feasible solution. In the neural network, Levy flying chooses the nearest best weight value. In comparison with other models such as ABC-ANN, SFR, NNID, and FC-ANN, the network performance is enhanced due to optimal weight choices. This demonstrates that we were successful in achieving our goal.

6 Conclusion

This work presents a novel IDS based on the combination of ANN and the cuckoo search optimization technique. ANN is used for classification, and cuckoo search is used to train ANN. The training of ANN is accomplished with the updating value of weights for better results. The proposed scheme is evaluated on the benchmark NSL-KDD dataset for the identification of normal and abnormal traffic. The evaluation metric contains parameters MAE, MSE, RMSE, and accuracy. The proposed method is also compared with standard methods available in the literature such as FC-ANN, NNID, SRF, and IDS-ABC. The simulation results, i.e. accuracy $= 99.35\%$, MAE $= 0.0097$, MSE $= 0.011$, RMSE $= 0.0059$, clearly indicate the superior performance of the proposed method against standard methods available in the literature. In the future, we plan to evaluate the proposed model on further datasets and compare the results with other standard models available in the literature.

Acknowledgements The authors would like to thank Austrian Agency for International Cooperation in Education and Research (OeAD) and Ernst Mach Follow Up Grant Program.

Author Contributions All authors contributed to the study conception and design. Data curation, methodology, and software were performed by Sangeen Khan, Sajid Anwar, and Fakhri Alam Khan. Formal analysis, investigation, validation, and writing were performed by Muhammad Imran and Helmut Hlavacs. Muhammad Imran, Sangeen Khan, and Fakhri Alam Khan contributed to the design and implementation of the research.

Funding This research received no funding from any agency in the public, commercial, or not-for-profit sectors.

Declarations

Ethical approval For this type of study, formal consent is not required.

Conflict of interest The authors declare that they have no conflict of interest.

Informed consent For this type of study, formal consent is not required.

References

Aghdam MH, Kabiri P et al (2016) Feature selection for intrusion detection system using ant colony optimization. Int J Netw Secur 18(3):420–432

Ahmad Z, Shahid Khan A, Wai Shiang C, Abdullah J, Ahmad F (2021) Network intrusion detection system: a systematic study of machine learning and deep learning approaches. Trans Emerg Telecommun Technol 32(1):4150

Alamiedy TA, Anbar M, Alqattan ZN, Alzubi QM (2019) Anomaly-based intrusion detection system using multi-objective grey wolf optimisation algorithm. J Ambient Intell Human Comput 1–22

Aldweesh A, Derhab A, Emam AZ (2020) Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues. Knowl Based Syst 189:105124

Ali MH, Al Mohammed BAD, Ismail A, Zolkipli MF (2018) A new intrusion detection system based on fast learning network and particle swarm optimization. IEEE Access 6:20255–20261

Almseidin M, Alzubi M, Kovacs S, Alkasassbeh M (2017) Evaluation of machine learning algorithms for intrusion detection system. In: 2017 IEEE 15th international symposium on intelligent systems and informatics (SISY). IEEE, pp 000277–000282

Atefi K, Yahya S, Dak A.Y, Atefi A (2013) A hybrid intrusion detection system based on different machine learning algorithms. In: Proceedings of the 4th international conference on computing and informatics, ICOCI. pp 312–320

Axelsson S (2000) Intrusion detection systems: a survey and taxonomy. Technical report, Citeseer

Baraneetharan E (2020) Role of machine learning algorithms intrusion detection in WSNs: a survey. J Inf Technol 2(03):161–173

Batista LO, de Silva GA, Araújo VS, Araújo VJS, Rezende TS, Guimarães AJ, Souza PVDC (2019) Fuzzy neural networks to create an expert system for detecting attacks by SQL injection. arXiv:1901.02868

da Costa KA, Papa JP, Lisboa CO, Munoz R, de Albuquerque VHC (2019) Internet of things: a survey on machine learning-based intrusion detection approaches. Comput Netw 151:147–157

Debar H (2000) An introduction to intrusion–detection systems. Proc Connect 2000

Debar H, Dacier M, Wespi A (1999) Towards a taxonomy of intrusion–detection systems. Comput Netw 31(8):805–822

Farahnakian F, Heikkonen J (2018) A deep auto-encoder based approach for intrusion detection system. In: 2018 20th international conference on advanced communication technology (ICACT). IEEE, pp 178–183

Ferdiana R et al (2020) A systematic literature review of intrusion detection system for network security: research trends, datasets and methods. In: 2020 4th international conference on informatics and computational sciences (ICICoS). IEEE, pp 1–6

Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29(1):17–35

Gao Y, Wu H, Song B, Jin Y, Luo X, Zeng X (2019) A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc network. IEEE Access 7:154560–154571

Garcia-Teodoro P, Diaz-Verdejo J, Maciá-Fernández G, Vázquez E (2009) Anomaly-based network intrusion detection: techniques, systems and challenges. Comput Sec 28(1–2):18–28

Hubballi N, Suryanarayanan V (2014) False alarm minimization techniques in signature-based intrusion detection systems: a survey. Comput Commun 49:1–17

Imran M, Hlavacs H, Khan FA, Jabeen S, Khan FG, Shah S, Alharbi M (2018) Aggregated provenance and its implications in clouds. Future Gener Comput Syst 81:348–358

Khan A, Shah R, Imran M, Khan A, Bangash JI, Shah K (2019) An alternative approach to neural network training based on hybrid bio meta-heuristic algorithm. J Ambient Intell Humaniz Comput 10(10):3821–3830

Khan FA, Shaheen S, Asif M, Rahman AU, Imran M, Rehman SU (2019) Towards reliable and trustful personal health record systems: a case of cloud-dew architecture based provenance framework. J Ambient Intell Humaniz Comput 10(10):3795–3808

Krishnaveni S, Vigneshwar P, Kishore S, Jothi B, Sivamohan S (2020) Anomaly-based intrusion detection system using support vector machine. In: Artificial intelligence and evolutionary computations in engineering systems. Springer, pp 723–731

Kumar V, Sangwan OP (2012) Signature based intrusion detection system using snort. Int J Comput Appl Inf Technol 1(3):35–41

Maseer ZK, Yusof R, Bahaman N, Mostafa SA, Foozy CFM (2021) Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. IEEE Access 9:22351–22370

Othman SM, Ba-Alwi FM, Alsohybe NT, Al-Hashida AY (2018) Intrusion detection model using machine learning algorithm on big data environment. J Big Data 5(1):1–12

Rao KR, Battula SK, Krishna TLSR (2017) A smart heuristic scanner for an intrusion detection system using two-stage machine learning techniques. Int J Adv Intell Paradigms 9(5–6):519–529

Saranya T, Sridevi S, Deisy C, Chung TD, Khan MA (2020) Performance analysis of machine learning algorithms in intrusion detection system: a review. Proc Comput Sci 171:1251–1260

Singh AP, Singh MD (2014) Analysis of host-based and network-based intrusion detection system. Int J Comput Netw Inf Sec 6(8):41–47

Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the KDD cup 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications. IEEE, pp 1–6

Wang S-S, Yan K-Q, Wang S-C, Liu C-W (2011) An integrated intrusion detection system for cluster-based wireless sensor networks. Expert Syst Appl 38(12):15234–15243

Wang H, Gu J, Wang S (2017) An effective intrusion detection framework based on SVM with feature augmentation. Knowl Based Syst 136:130–139

Yang X-S, Deb S (2010) Engineering optimisation by cuckoo search. Int J Math Model Numer Optim 1(4):330–343

Zhou AT, Blustein J, Zincir-Heywood N (2004) Improving intrusion detection systems through heuristic evaluation. In: Canadian conference on electrical and computer engineering 2004 (IEEE Cat. No. 04CH37513), vol 3. IEEE, pp 1641–1644