Wireless Personal Communications (2024) 137:1559–1580
https://doi.org/10.1007/s11277-024-11396-6

Unified Intrusion Detection Framework: Predictive Analysis of Intrusions in Sensor Networks

Arun Kumar Ramamoorthy[1] · **K. Karuppasamy**[2]

Accepted: 20 June 2024 / Published online: 20 July 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract
Intrusion Detection Model (IDM) is an essential device for network defence in current trend. Malicious users analyse the vulnerabilities of IDSs to capture unauthorized access. Furthermore, intrusion detection encompasses numerous numerical attributes and models, resulting in elevated detection errors and triggering false alarms. Hence, optimal computational intelligence shall be incorporated in IDM to achieve high detection rate and less number of false alarms. Considering the same, a new hybrid IDM framework is developed as the combination of Fuzzy Genetic Algorithm with Multi-Objective Particle Swarm Optimization that maximizes the detection accuracy, minimizes the false alarms and takes less computational complexity which will be explained first phase. The existing IDSs are constraint to the information trained incur into false positives based on user continuity for normal activity. The objective of this proposal is to extract optimal classification rules automatically from training data that helps to identify types of attacks correctly including the unknown attack types. For achieving this goal, Multi-Objective Particle Swarm Optimization (MOPSO) is used as classifier to enhance the identification of the rare attack classes within the IDM. The effectiveness of this method lies in its capacity to leverage information within an unfamiliar search space, guiding subsequent searches towards valuable subspaces. It provides better separability of various classes' i.e. normal behaviour and false alarms. In this FGA-MOPSO model, Principal Component Analysis (PCA) serves as the feature selection technique employed to identify pertinent features within the dataset, thereby enhancing the classifier's performance and Fuzzy Genetic Algorithm (FGA) is used to create new population for training the classifier with the help of three operations namely selection, crossover and mutation that helps to practice more patterns in training phase and to obtain better understanding of the proposed classifier. The simulation will illustrate that the system is competent to speed-up the training and testing process of intrusions detection is important for network applications.Please confirm if the author names are presented accurately and in the correct sequence (given name, middle name/initial, family name). Author 1 Given name: [Arun Kumar] Last name [Ramamoorthy]. Also, kindly confirm the details in the metadata are correct.Checked and Verified for Author 1. In Author 2 name, Given Name was [K.] and last name was[Karuppasamy], But its is just the opposite. Given Name is [Karuppasamy] and Last Name is [K.]. I have edited it.

✉ Arun Kumar Ramamoorthy
arunkramamoorthy@gmail.com

[1] Digital Forensics and Cyber Security, University of South Wales, Treforest CF37 1DL, UK

[2] Department of CSE, RVS College of Engineering & Technology, Coimbatore 641402, India

Keywords Detection Accuracy · Precision · F-measure · Recall · Testing time · Training time · False alarm ratio using NSL-KDD data set

1 Introduction

The emergence of wireless networking significantly relies on the self-organized and multi-hop network environment. It aggregates huge amount of sensor nodes through wireless communication and characterized as simpler and low cost network deployment [1]. It is extensively adopted in real-time environment like military exploration, modern logistics, and environment perception where the connected sensor nodes collaboratively works to carry out detection, monitoring, and tracking of certain malicious nodes or intruders over the network [2]. Specifically, WSN-based intrusion detection system is used to handle security issues encountered during rescuing of post-disaster, region monitoring, and border patrol and turns as generic field of modern research. Thus, it needs constant monitoring and tracking method for the prediction of intrusion and thus there is a need for design to deal with these multi-objective constraints to attain high-quality and persistent handling of the intruder [3].Please confirm the section headings are correctly identified.Checked and Verified.

Some present investigations over intrusion detection is partitioned into diverse two categories: the former one is to perform trace prediction and accurate localization of the target by adoptively sensing the information from diverse nodes based on local voting and decision fusion approaches [4]. The second model relies on the movement and deployment strategies on SNs to attain enhanced dynamic target coverage. It is considered as an addition of conventional coverage optimization issues and it is the specific concern of this work [5]. The coverage quality is drastically influenced by the preliminary deployment of the SN localization. However, owing to the hostile or remote sensing environments, for example, region monitoring or border patrol based sensor deployment is not manually handled in most real-time environment [6]. Therefore, usually, the sensors are deployed with the scattering of aircrafts; moreover the appropriate position for deriving the landing is not controlled owing to the existence of obstacles and wind like mountains and trees. Subsequently, certain sub-areas does not possess appropriate sensor coverage region where diverse sensors are removed and some regions are identified with coverage issues (regions that does not comes under the coverage region) [7].

Generally, it is crucial to get rid of these issues and addition of sensors for predicting intrusions can be attained only with the adoption of miniaturized robots and embedded hardware's. Some sensors possess similar sensing competency and considered as the static sensors and it has the ability to move towards the appropriate locations for offering optimal coverage after the node deployment [8]. Regrettably, the nodes are not competent of tracking and predicting the intruders to enhance the coverage quality. This condition is still worse with the emergence of anti-reconnaissance methods over the prediction of intruders in real-world environment. It is equipped with some sensing devices and attains location information regarding the detection nodes and carry out planning to eradicate the detection process. These intruders are depicted as an 'empowered intruder' and differ from the native intruders and the elegant nature of the SN's tracking makes it stubborn. Thus, the design of effectual intrusion detection approaches for these sorts of intruders are a challenging task [9].

Conventional intrusion detection approaches for region monitoring or border patrol relies on the centralized network architecture. The intruders or the intermediate nodes transfer the information to the cluster nodes or base station and takes necessary action after

information processing or analysis [10]. This method necessitates recurrent interaction among the cluster nodes, base station, and detection nodes. It is occupy huge amount of network nodes and increases the networks' transmission delay. Thus, it outcomes delayed handling issues like interrupted events or intruder prediction [11]. Subsequently, the conventional centralized framework is inappropriate for some real-time scenario specifically over the highly-influenced intruders. The nodes have to maintain the records of the process to perform local computation, tracking of trajectories in the real-time environment [12]. Moreover, the node does not possess certain efficiency to deal with these problems.

In the modern era of computation intelligence, various approaches are non-classical approaches the works like human beings to learn certain tasks from the observations or data [13]. Subsequently, this intelligence system possesses some characteristics to make the model more feasible and to be adopted in the construction of effectual models in diverse fields. Some of its features include fault tolerance, high computational speed, competency to deal with error resilience, adaptability during the model of noisy information [14]. This research work considers Fuzzy Logic (FL) which is one among the intelligence technique that is inspired from the human brain activities with uncertainty measure. It is also considered as the logic system or rule-emergence system with appropriate features and tolerance towards uncertainty and imprecision. Thus, it performs rule-based classification in an effectual manner. Moreover, it is not self-adaptive and it acts as a candidate for optimization purpose. Here, Particle Swarm Optimization is considered which is most popular for handling the multi-objective constraints and functions as global optimization ability with Genetic Algorithm (GA). Thus, this work models a novel Fuzzy Genetic Algorithm with Multi-Objective Particle Swarm Optimization that maximizes the detection accuracy, minimizes the false alarms and takes less computational complexity. The anticipated model is tested, validated and proven with the competency or evolution of optimization model with superior accuracy and lesser FAR, improves classification accuracy for certain attacks. The features are chosen and analyzed using Principle Component Analysis (PCA). The data source is attained from the online accessible NSL-KDD dataset. The simulation takes place within the MATLAB environment, incorporating metrics such as accuracy, precision, FAR, and more.

The structure of the work is as follows: Sect. 2 comprises an in-depth survey of various existing approaches related to IDS, along with their associated pros and cons. Section 3 elaborates on the methodology in a broader sense, focusing on gaining insight into the prediction model. In Sect. 4, the discussion revolves around the results obtained from model evaluation, presented graphically. Finally, Sect. 5 presents the conclusion of the work, along with suggestions for future improvements.

2 Related Works

This section gives the recent updation regarding the data taxonomy along with certain research ideas on IDS up to data and the classification systems used for this prediction taxonomy. It offers a comprehensive and structural overview on prevailing IDS. Therefore, the research becomes proficient with certain key factors in anomaly detection.

Osanaiye et al. [15] discusses signature-based IDS for pattern matching approaches to predict the unknown attacks. Also, it is termed as misuse detection or knowledge-based detection. With this model, matching approaches are utilized to predict various intruders. Subsequently, when the intrusion signature fits with the existing intrusion signature that prevails over the signature database, then an alarm signal is found to be triggered. In case

of SIDS, the host logs are identified to predict the commands sequence or actions that are previously determined as malware. It is also labelled over the reviews as misuse detection or knowledge-based detection process. Li et al. [16] discusses conventional approaches that are used for intrusion detection using network packets and pretends to match against the signature databases. However, these approaches are incapable to predict the attacks that span various packets. It is extremely essential to haul out signature information as the modern malwares are completely sophisticated over the multiple packets. It needs IDS for content recall for various packets. Generally, there are diverse methods that are used for the creation of state machines, semantic conditions, and formal language string patterns indeed of creating various IDS signatures.

Zhou et al. [17] discusses the significant benefits of various IDS to predict zero-day attacks owing to the fact that the prediction of abnormal user functionality does not based on the signature database. It induces some dangerous signals while analyzing the nature that varies from usual characteristics. Moreover, it possesses various advantages. Initially, it has the competency to predict the internal malicious functionalities. When the intruder initiates the tractions of the stolen account that are not identified by the user activities in a typical manner, it triggers the alarm condition. Next, it is extremely complex for the cyber-criminal to predict what sort of user's characteristics is constructed devoid of any alert system form the customized profiles. Almomani et al. [18] discusses various categories of IDS methods and it is known as machine-learning based, knowledge-based, and statistics-based approaches. The last model includes examination and collection of various data records over the set of items and the construction of statistical model with normal user characteristics. Subsequently, knowledge-based model pretends to predict the essential activities from prevailing data systems like network traffic instances and protocol specifications. For instance, machine-learning approaches need complex pattern matching approach for training data.

Ioonnou et al. [19] discusses various machine learning approaches. It is a process of hauling out knowledge from huge amount of data. It is a model which is composed of set of rules, complex transfer functionality, and methods which is used to predict the essential data patterns, predict or examines the nature of the model. The learning approaches are used widely in the field of IDS. Various techniques and algorithms like NN, DT, clustering, association rules, GA and K-NN approaches are adopted for predicting or learning knowledge from intrusion datasets. Ghosal et al. [20] discusses a approach to perform feature selection using the integration of feature selection approaches like correlation attribute evaluation and Information Gain. The author validates the performance by selecting the features by applying diverse classification approaches like NB, C4.5, NB-tree and MLP respectively. Almomani et al. [21] applies genetic-fuzzy rules based mining approaches which is used for evaluating the significance of the IDS characteristics. Ke et al. [22] discusses IDS with the adoption of Random forest to enhance the prediction accuracy and Arun et al. discusses how to diminish the FAR [23]. Khraisat et al. [24] anticipates a classification approach using NSL-KDD dataset with DT algorithm to design of a model with certain metrics and examines the significance of DT approaches.

Ali et al. [25] discusses a classifier model known as Support Vector Machine (SVM) determined by partitioning the hyperplanes. It adopts kernel function to map the training data into high-dimensional space. Therefore, the intrusion is classified in a linear manner. It is well-known for its generalization ability and notably value when the number of attributes is larger and number of data points is completely smaller. Various kinds of hyperplane separation are attained with the adoption of kernel functions like hyperbolic tangent, Gaussian radial basis function, linear and polynomial functions. With IDS dataset, some

features are less influencing and redundant in data point separation into appropriate classes. Thus, feature selections are determined by SVM training. Also, SVM is adopted for classification purpose into multiple classes. Buczak et al. [26] describes SVM with RBF kernel function which is used for categorizing KDD'99 dataset in pre-defined classes. From the provided 41 attributes, the feature subset is selected in a careful manner by selecting feature selection approaches.

Peng et al. [27] depicts k-NN classifier which is a non-parametric classifier in a typical manner and applied over ML approaches. The concept behind this approach is to name the provided unlabelled data sample towards the k-NN classes. Here, 'k' is an integer that predicts the number of neighbours. Generally, $k = 5$ for most cases. Here, 'x' specifies he unlabelled data instances that need to be categorized. From the provided five NN, three NN possess similar patterns from the given intrusion class and two from normal class. With the major voting model, it facilitates 'X' for the intrusion class. Ibrahim et al. [28] anticipates a novel fuzzy-based supervised learning model by adopting unlabelled samples along with supervised learning model to improve IDS classifier performance [29, 30]. Then, the SH-FFNN model is trained for providing the output with fuzzy-based membership vector function and sample classification (high, mid and low fuzzy classifiers) over the unlabelled sample which is done with fuzzy quantifiers. The classifier is then re-trained after the integration of every category into original training set separately. The experimental outcomes use semi-supervised intrusion detection over NSL-KDD dataset and projects unlabelled samples with high and low fuzziness which leads to predominant contributions to improve the IDS prediction accuracy in contrast to conventional approaches.

This section presents a detailed review on various IDS methods, corresponding types and methodologies with significant advantages and constraints. Various machine learning approaches are used for predicting the malicious activities and intruders over sensor networks. Moreover, some of these approaches posses certain constraints during the generation and updation of data regarding the newer attacks and it provides high FAR or least accuracy. The results and methods are summarized and the contemporary models are explored based on the performance enhancements on IDS as an outcome to get rid of IDS issues.

3 Methodology

Here, a detailed discussion is done for validate the performance of proposed fuzzy genetic algorithm and MOPSO model. Some preliminary sets like data acquisition, feature selection, and classification is performed to identify the intrusion over the network. The detection framework is shown in Fig. 1.

3.1 Dataset Description

In this context, the NSL-KDD dataset is employed, where 20% of its instances serve as training data out of a total of 25,192 instances, while the remaining samples, totalling 22,544 instances, constitute the testing dataset. This dataset comprises 42 attributes, with 41 of them classified into four distinct classes.

1. Basic (B) characteristics: TCP/IP connection attributes utilized in identifying delays.

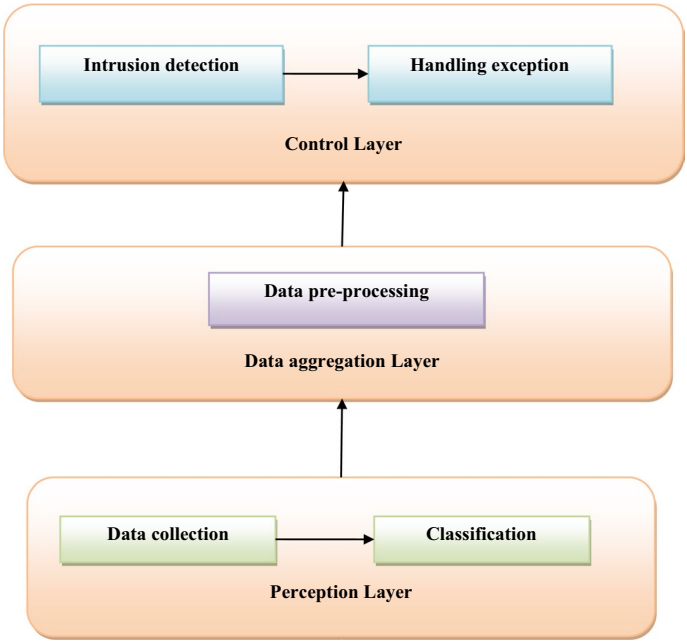

Fig. 1 Intrusion detection framework

2. Traffic (T) characteristics: These attributes pertain to window intervals and encompass two prominent features, namely, same service and same host. The service feature evaluates the overall number of connections sharing the same services within a specific time frame.
3. Host (H) characteristics: These attributes are assigned to assess attacks lasting for 2 s, scrutinizing the overall connections directed towards the destination during this duration.
4. Content (C) characteristics: These attributes, informed by domain expertise, are suggested based on moment intervals.

This dataset encompasses four distinct traffic categories, each associated with 23 types of attacks, along with various features:

1. Denial of Service (DoS): Attackers monopolize network resources, rendering them unavailable to legitimate users.
2. User-to-Root (U2R): Attackers intercept passwords and exploit vulnerabilities on hosts to gain unauthorized access as legitimate users.
3. Remote-to-Local (R2L): Attackers transmit messages from remote locations to hosts, exploiting vulnerabilities in the process.
4. Probe: Attackers scan the network to gather information, leading to network breaches. Tables 1 and 2 detail the dataset's records, labels, and attributes from the NSL-KDD dataset, while Table 3 delineates the four distinct attack categories.

Table 1 Dataset records

Dataset	Record count					
	Total	DoS	R2L	Normal	U2R	Probe
KDD testing	22, 544	7458	2654	9711	200	2421
KDD training	125, 973	45, 927	995	67, 343	52	11, 656
KDD (train + 20%)	25, 192	9234	209	13, 449	11	2289

3.2 Feature Selection Using Principle Component Analysis

PCA is a statistical approach which is applied in various applications like image compression, face recognition, image processing and so on. It is a common approach for predicting the patterns of high dimensional data. The complete statistical data is based on huge dataset and analyzes the relationship among the individual points (See Table 4). The objective of PCA is to diminish the data dimensionality by measuring the variations identified in the original NSL-KDD dataset. It identifies the data patterns by expressing the differences and similarities among the dataset. Please check the edit made in caption of Algorithm 1. Please check if action taken is appropriate. Otherwise, kindly advise us on how to proceed.Yes Its perfect.

Algorithm 1 The flow of PCA functionality

Assume $x_1, x_2, \ldots, x_m$ are $n * 1$ vectors

1. Compute $\bar{x} = \frac{1}{M}\sum_{i=1}^{M} x_i$;
2. Subtract mean $\emptyset_i = x_i - \bar{x}$;
3. Compute matrix $A = \ \phi_1\phi_2 \ldots \phi_M]$ //$(N * M)$ matrix;

$C = \frac{1}{M}\sum_{N=1}^{M} \phi_n\phi_n = AA^T$ //covariance matrix that characterizes the data

4. Perform eigen value computation with 'C' //$C: \lambda_1 > \ \lambda_2 > \cdots > \lambda_N$
5. Evaluate the eigen vectors //$C: u_1, u_2, \ldots, u_N$
6. When $'C'$ is symmetric, then $u_1, u_2, \ldots, u_N$ is also symmetric;
7. Any form of vectors $x_i - \bar{x}$ is a linear combination of eigen vectors;
8. $x_i - \bar{x} = b_1u_1 + \ b_2u_2 + \cdots + b_Nu_N \ = \ \sum_{i=1}^{N} b_iu_i$;
9. Perform dimensionality reduction based on largest eigen values 'K';
10. $x_i - \bar{x} = \ \sum_{i=1}^{K} b_iu_i$;
11. Select 'K' based on following strategy

$\frac{\sum_{i=1}^{K}\lambda_i}{\sum_{i=1}^{N}\lambda_i} > threshold\ value\ (0.9)$;

12. end process

Table 2 Dataset labels and attributes

No	Label	Name	No	Label	Name	No	Label	Name	No	Label	Name
1	T	Srv_count	10	C	Num_root	23	B	Duration	32	H	Dst_host_serror_rate
2	T	Srv_serror_rae	11	C	Num_file_creations	24	B	Protocol_type	33	H	Dst_host_srv_serror_rate
3	T	Srv_rerror_rate	12	C	Num_shell	25	B	Service	34	H	Dst_host_rerror_rate
4	T	Srv_diff_host_rate	13	C	Num_access_files	26	B	Src_bytes	35	H	Dst_host_srv_rerror_rate
5	T	Count	14	C	Num_outbound_cmds	27	B	Dst_bytes	36	H	Dst_host_count
6	T	Serror_rate	15	C	Is_hot_logins	28	B	Flag	37	H	Dst_host_srv_count
7	T	Rerror_rate	16	C	Is_guest_logins	29	B	Land	38	H	Dst_host_same_srv_rate
8	T	Same_srv_rate	17	C	Hot	30	B	Wrong_fragment	39	H	Dst_host_diff_srv_rate
9	T	Diff_srv_rate	18	C	Num_failed_logins	31	B	Urgent	40	H	Dst_host_same_src_port_rate
			19	C	Logged_in				41	H	Dst_host_srv_diff_host_rate
			20	C	Num_compromise				42	——	Class
			21	C	Root_shell						
			22	C	Su_attempted						

Table 3 Classifications of breaches

Attacks	Occurrences of breaches in each category
DoS	Fragmentation attack, DDOS, Ping of death, SYN flood, land, back
Probe	ICMP sweep, Nmap, IPsweep, satan
R2L	Warezclient, chain attack, Spy, Phf Script, guess_passwd, IMAP, FTP_write, Warezmaster
U2R	Rootkit, perl, loadmodule, buffer_overflow exploitation

Table 4 Feature dimensionality reduction

Attack	Features
Warezmaster	f_1, f_{34}
Warezclient	f_5, f_{15}, f_{28}
Teardrop	$f_5, f_{26}, f_{30}, f_{34}$
Spy	$f_{39}, f_{34}, f_{19}, f_{18}, f_{17}, f_{15}, f_{12}$
Smurf	$f_{37}, f_{12}, f_6, f_5, f_1$
Satan	$f_{40}, f_{39}, f_{36}, f_{35}, f_{34}, f_{30}, f_{29}, f_{27}, f_{12}, f_6, f_1$
Root kit	$f_{41}, f_{17}, f_{16}, f_{13}, f_{12}, f_{11}, f_{10}, f_9$
Port sweep	$f_{41}, f_{40}, f_{39}, f_{38}, f_{36}, f_{35}, f_{34}, , f_{30}, f_{29}, f_{28}, f_{27}, f_{26}, f_{25}, f_1$
Pod	f_{36}, f_{35}, f_{34}
Phf	$f_{24}, f_{15}, f_{10}, f_5$
Perl	f_{34}, f_5, f_1
normal	$f_{37}, f_{36}, f_{35}, f_{34}, f_{31}$
Nmap	$f_{39}, f_{38}, f_{36}, f_{35}, f_{34}, f_{26}, f_{25}, f_5$
Neptune	$f_{35}, f_{34}, f_{30}, f_{29}$
Multi-hop	$f_{40}, f_{36}, f_{35}, f_{34}, f_{31}, f_{22}, f_{17}, f_{12}, f_{10}, f_6, f_5, f_1$
Load module	$f_{41}, f_{40}, f_{38}, f_{36}, f_{25}, f_{23}, f_{12}$
Land	$f_{39}, f_{38}, f_{35}, f_{34}, f_{17}, f_{15}, f_6, f_1$
Ipsweep	$f_{41}, f_{40}, f_{37}, f_{36}, f_{35}, f_{34}, f_{31}, f_{28}, f_{27}, f_5$
Imap	$f_{38}, f_{28}, f_{25}, f_{24}, f_{12}, f_8$
Guess_pwd	$f_{41}, f_{40}, f_{39}, f_{38}, f_{31}, f_{28}, f_{27}$
ftp_write	$f_{37}, f_{36}, f_{35}, f_{34}, f_{31}, f_{22}, f_{19}, f_{17}, f_{16}, f_{13}, f_{12}, f_{10}, f_9, f_6, f_5, f_1$
Buffer overflow	$f_{37}, f_{36}, f_{16}, f_{14}$
back	$f_{41}, f_{40}, f_{39}, f_{38}, f_{36}, f_{31}, f_{28}, f_{27}$

3.3 Design of Fuzzy Genetic Algorithm

A classifier model is nothing but the algorithm used for the construction of classification model from the provided dataset to categorize the data. The significance of the model is managed with various parameters like fuzzy set, fuzzy rules, and membership function and prioritization values. Generally, fuzzy logic lacks in learning ability where the optimization process is considered to be more complex. Here, the fuzzy rules, membership function, and fuzzy sets are optimized. The fuzzy rule set is specified by IF–THEN rules. The generation of rule size is based on feature size and it is managed

by the dataset adopted. Moreover, to handle the classification ignorance, the numbers of rules are provided in a constraint manner. Generally, membership functions and fuzzy sets are feature-dependent. The membership function can be either trapezoidal or triangular shapes. Three fuzzy sets are considered to reduce the computational complexity. The fuzzified input mapping towards rule-base model is done with inference process to generate fuzzified output for all appropriate rules. The rule is generated based on the following Eq. (1):

$$\propto R_i = \min\left\{\mu D_1(d_1), \mu D_2(d_2), \ldots, \mu D_n(d_n)\right\} \tag{1}$$

Here, $\propto R_i$ is R_i^{th} fuzzy rule set, $'n'$ is number of features, $d_1, \ldots, d_n$ is input variables, $\mu D_i(d_i)$ is fuzzified membership degree, μ_{Di} is fuzzy set membership function. The fuzzy value (single) is allocated for all output. The final value is related with the output using maximal operator and it is expressed as in Eq. (2):

$$\beta_i = \max_{\text{for all M}}\left\{\alpha_{Ri}\right\} \tag{2}$$

Here, β_i is maximal value for all fuzzy rules, α_{Ri} is fuzzy rule strength, $'M'$ are total fuzzy rules. The defuzzification process evaluates the centroid and transforms the fuzzy output to crisp values using fuzzy rules. It is expressed as in Eq. (3):

$$\text{Output} = \frac{\sum_{i=1}^{n}\left(\alpha_{Ri} * \mu D_i(d_i)\right)}{\sum_{i=1}^{n}\alpha_{Ri}} \tag{3}$$

Here, $\alpha_{Ri} * \mu D_i(d_i)$ is the maximal defuzzification process, $'n'$ is total amount of fuzzy rules. Here, the parameters are evaluated with Genetic algorithm and it is used for categorizing the attacks where the models are used for predicting and classification of attacks. Algorithm 2 iIllustrates the genetic fuzzy algorithm

Algorithm 2 Genetic fuzzy algorithm

Input: NSL-KDD dataset, MaxGen, population, GA population;
Output: classified results
1. begin the process;
2. $data\ pre-processing \rightarrow^{'} d^{'}$;
3. parameter initialization;
4. Extract fuzzy rules using best genetic chromosomes;
5. $final\ output \rightarrow$ fuzzy rule extraction;
6. Test the model with fuzzy rules;
7. $evaluate\ results \rightarrow final\ model$;
8. end

//Fuzzy rule generation

9. begin
10. weighted array → empty;
11. $crispoutput \rightarrow 0$;
12. for all training dataset records do
13. weighted array → fuzzification (chromosomes, records);
14. $crispoutput \rightarrow defuzzification\ (records, weighted\ array, chromosome)$;
15. bestfit (average) → perform classification;
16. error (chromosome, records, crispoutput);
17. end for
18. return (average bestfit);
19. end

//Update chromosomes

20. begin
21. sort fitness ();
22. crossover selection ();
23. mutation ();
24. replacement ();
25. return GA generation (new);
26. end

The genetic algorithm encodes (provides) fuzzy rules and the chromosomes are modelled to encode the rule-base. The fuzzy rules are specified with integer array where the size of the array is equal to the chosen feature size from the NSL-KDD dataset. The encoding process specifies the dataset features through the membership function for the chosen rule-base. The encoded chromosome fitness is evaluated with the fuzzy set, and the chromosomes. The classification accuracy is expressed as in Eq. (4) and Eq. (5):

$$\text{fitness} = \frac{1}{\text{classification error}} \tag{4}$$

$$\text{Error} = 2E^2 + E + 1 \tag{5}$$

Here, $'E'$ is specified as the percentage of inappropriately categorized records. The error (classification) is specified in a quadratic manner. The roulette wheel selection process is used for selecting the appropriate parents for reproduction process. The crossover is adopted for all chromosome pairs in a random manner during reproduction. The chromosome layers are provided with fixed length under a constraint environment. Here, random mutation process is done with mutation selection probability. The best solution is attained with the adoption of elitism and helps to construct the successive generation. It involves in the substitution of the older population by transforming the of fitness candidates into the successive generation. The relationships among the chromosomes are attained with the collaboration of $'K'$ rules to predict the categories of the attack. Figure 2 illustrates the flow diagram of the proposed MOPSO.

3.4 Multi-Objective Particle Swarm Optimization (MOPSO)

PSO is a bionic concept that originates from the bird's characteristics and the preliminary concept behind it is to predict the optimal solution via the information sharing and cooperation between the individual over the group. The speed and position of the bird are considered as an independent variables and food density arrives with the functional values. The search can adjust the speed and direction based on the difference among the optimal location and population history. The entire bird swarm attains optimal location based on the population. Therefore, the findings may get optimal solution, i.e. problem convergence. The predominant benefits of PSO are:

1. Stronger competency towards global search and faster computational speed.
2. It is not so sensitive towards the population size with smaller effect over the training speed.
3. There is no necessity towards the computation of gradient information while performing objective function optimization. It is no constraint towards connectivity, derivability, convexity, and continuity over the feasible areas of the objective function.

Multi-objective PSO intends to give solution to various domain related problems in an efficient manner. It is conceptualized as a random search problem across a D-dimensional space, aiming to optimize the objective function. Here, $'n'$ particles population $p_i = (p_{i1}, p_{i2}, \cdots, p_{iD})^T$ and i^{th} particle composed of $d-$ dimensional position vector $x_i = (x_{i1}, x_{i2}, .., x_{id})^T$ and velocity vector $v_i = \left(v_{i1}, v_{i2}, \ldots, v_{id}\right)^T$. For all population (particle), fitness value is attained based on the evaluation of particle fitness. The fitness function is expressed in Eq. (6):

$$F(X) = \alpha(1-p) + (1-\alpha)\left(1 - \frac{N_f}{N_t}\right) \tag{6}$$

Here, $'\alpha'$ is hyper-parameter, $'p'$ shows the coordinate relationship between the classifier performance, N_f is the feature subset. When the search is over the $D-$ dimensional space, then initialize the random particles and optimal solution is determined via iteration. With constant particle search, the optimal position $p_i = \left(p_{i1}, p_{i2}, \ldots, p_{id}\right)^T$ is the local optimal solution and velocity is specified as $v_i = (v_{i1}, v_{i2}, \ldots, v_{id})^T$. The optimal position

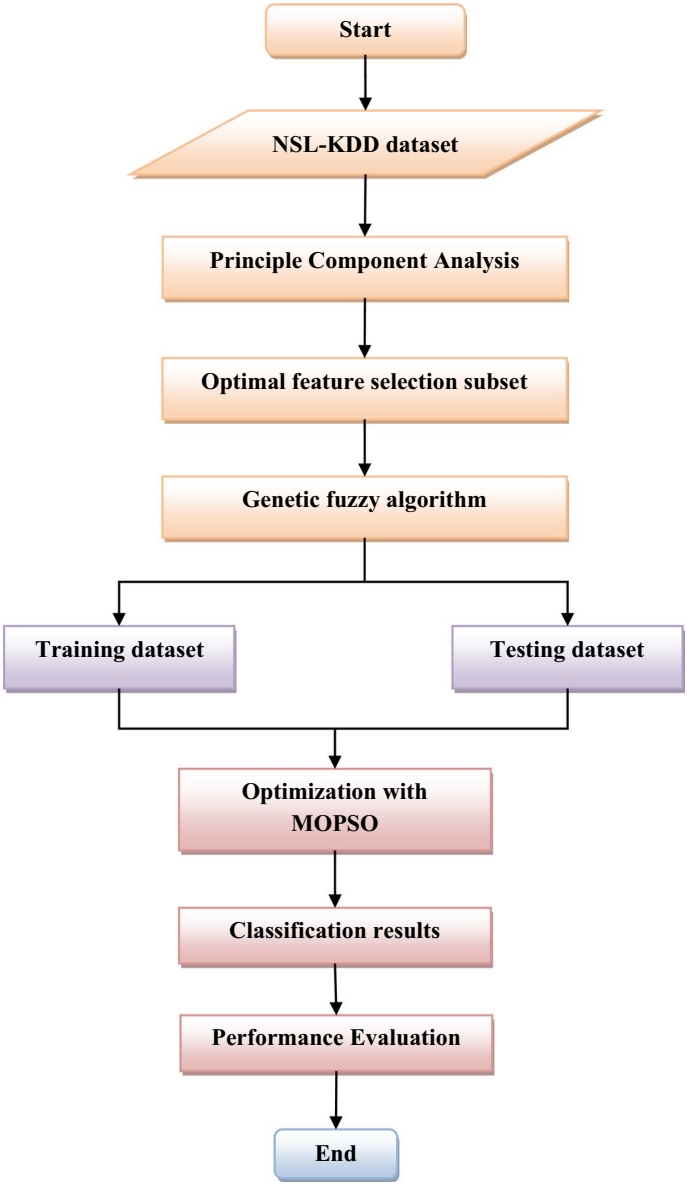

Fig. 2 The proposed algorithm framework

$p_g = (p_{g1}, p_{g2}, \ldots, p_{gd})$ is determined as global optimal solution. For all iteration, the particle needs to update the velocity and the position by measuring the 'optimal solutions', i.e. (p_i, p_g). The updation process is expressed as in Eq. (7):

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 \left(p_{id}(t) - x_{id}(t)\right) + c_2 r_2 \left(p_g d(t) - x_{id}(t)\right) \tag{7}$$

$$x_{id}(t+1) = x(t) + v_{id}(t+1), \text{where } i = 1, 2, \ldots, N; d = 1, 2, \ldots, D \tag{8}$$

Here, $'N'$ is total particles in the population with $d-$ dimensional space, $'t'$ is total present iterations, $'\omega'$ is non-negative inertia factor that manages local and global optimization capabilities. When the value is larger, the global optimization competency is stronger and local optimization competency is weaker. $v_{id}(t)$ and $v_{id}(t+1)$ specifies the current and updates particle velocity; c_1 and c_2 are acceleration factors where $c_1 = c_2 = 2.$ $'r_1'$ and $'r_2'$ are random numbers to improve the particle randomness and eliminates the blinding search. The particles position and velocity are constrained with $\left[-x_{\max}, x_{\max}\right]$ and $\left[-v_{\max}, v_{\max}\right]$. The algorithm for multi-objective PSO is given in Algorithm 3:

Algorithm 3 Multi-objective PSO

Input: Population size, local and optimal position, fitness function;
Output: p_g;
1. Initialize the velocity and position of the particles randomly;
2. while the criteria is not fulfilled do
3. for $i = 1$ to N do
4. compute the fitness value of all particles based on the fitness function;
5. if $fitness\ (x_i)$ is higher than $fitness\ (p_i)$ then
6. $p_i \rightarrow x_i$;
7. if $fitness\ (p_i)$ is higher than$fitness(p_g)$ then
8. $p_g \rightarrow p_i$;
9. Update the velocity and position of the particles;
10. return p_g

4 Results and Analysis of Data

This section presents the numerical results and discussion of the proposed MOPSO model. The simulation is conducted within the MATLAB environment, evaluating various performance metrics. The NSL-KDD dataset is utilized for training, testing, and validation in intrusion detection. The data prediction encompasses four distinct cases: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), with their corresponding analyses provided below.

Table 5 Confusion matrix

Actual	Predicted		
	Positive	Negative	Total
Positive	True positive	False negative	P (actual)
Negative	False positive	True negative	N (actual)
Total	P (predicted)	N (predicted)	P + N

1. TP: Indicates cases where both the predicted and actual labels are positive.
2. FN: Denotes instances where the predicted label is negative despite the actual labels being positive.
3. TN: Represents scenarios where both the predicted and actual values are negative.
4. FP: Refers to situations where the predicted label is positive despite the actual label being negative.

Table 5 depicts the confusion matrix of the anticipated model. Based on the above definitions, there are some metrics like False Alarm Rate (FAR), accuracy, and Detection Rate (DR) are measured for providing a novel IDS scheme. It is discussed below:

1. Detection Rate (DR): It is represented as the appropriate proportion of all positive instances, serving as a coverage measure that assesses the classifier's predictive capability for all positive instances. This is illustrated in Eq. (9):

$$DR = \frac{TP}{TP + FN} \tag{9}$$

2. Accuracy: It is represented as the appropriate prediction outcome relative to the total number of samples, serving as a measure to assess the overall accuracy rate of the classification samples. This is expressed in Eq. (10):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

3. False Alarm Rate: It is depicted as the predicted positive which is actually negative based on the proportional of appropriate negative. It is expressed as in Eq. (11):

$$FAR = \frac{FP}{TN + FP} \tag{11}$$

Table 6 depicts the comparison of prediction accuracy and FAR of the proposed MOPSO and existing ML approaches. The accuracy of the proposed MOPSO is 98.86% which is 12.06% higher than PSO with lightweight GBM, 13.36% higher than decision tree, 15.76% higher than logistic regression, 16.8% higher than NB, 17.46% higher than multi-layer perceptron, 17.41% higher than ANN and 20.46% higher than EM clustering (See Fig. 3). Similarly, the FAR of MOPSO is 9.5 which are 1.1, 6.2, 8.9, 9, 11.6, 11.8 and 14.2 lesser than other approaches. Table 7 depicts the total training and testing time

Table 6 Accuracy and FAR computation

Methods	Accuracy (%)	FAR (%)
MOPSO	98.86	9.5
PSO with lightweight GBM	86.8	10.6
Decision tree	85.5	15.7
Logistic regression	83.1	18.4
Naive bayes	82.06	18.5
Multi-layer perceptron	81.40	21.1
ANN	81.45	21.3
EM clustering	78.4	23.7

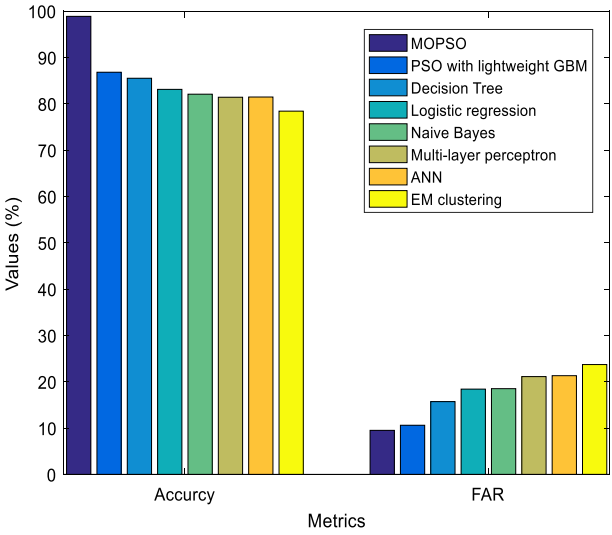

Fig. 3 Accuracy and FAR comparison

Table 7 Total training and testing time (s)

Dataset	Total training time (s)		Total testing time (s)	
NSL-KDD	Time lapsed	Processing time	Time lapsed	Processing time
	11.52	0.30	2.689	0.035

of NSL-KDD dataset w.r.t. elapse time and CPU time. The elapse time based on training is 11.52 s and CPU time is 0.30 s. The elapse time based on testing is 2.689 and CPU time is 0.035 s respectively (See Fig. 4).

Table 8 shows other metrics like precision, recall, F1-score and FAR of the proposed MOPSO respectively. The precision with normal category is 0.947%, recall is 0.995%, F1-score is 0.968 and FAR is 0.015. The values based on attack category shows 0.999% precision, 0.987% recall, 0.993% F1-scoreand FAR is 0.007. The weighted averages of all these metrics are given as 0.986%, 0.987%, 0.989% and 0.008% respectively (See Fig. 5). Table 9 depicts the precision, recall, F1-score and FAR of attack categories like DoS, probe, R2L and U2R respectively. For the DoS attack, the precision stands at 0.9940%, recall at 0.9790%, F1-score at 0.9860%, and FAR at 0.00450. In the case of the probe attack, precision is 0.8600%, recall is 0.8855%, F1-score is 0.9195%, and FAR is 0.5715. Moving to the R2L attack, precision records at 0.6920%, recall at 0.9195%, F1-score at 0.7895%, and FAR at 0.00550. Lastly, for the U2R attack, precision is 0.8880%, recall is 0.5715%, F1-score is 0.6965%, and FAR is 0.00002. The weighted averages of these metrics are 0.99%, 0.9886%, 0.9988% and 0.0996 respectively (See Fig. 6). The execution time (both training (ms) and testing (ms)) of proposed MOPSO is compared with PSO-lightweight GBM, DT, and logistic regression as in Table 10. The training time of MOPSO is 95.4565 ms which is 93.5735 ms, 5.0002 ms, 124.1083 ms lesser than other approaches. The testing duration for MOPSO is 2.5465 ms, representing a reduction of 0.505 ms, 2.3489 ms, and 9.7895 ms compared to alternative approaches (See Fig. 7)Based on these metrics, it is shown that the anticipated model

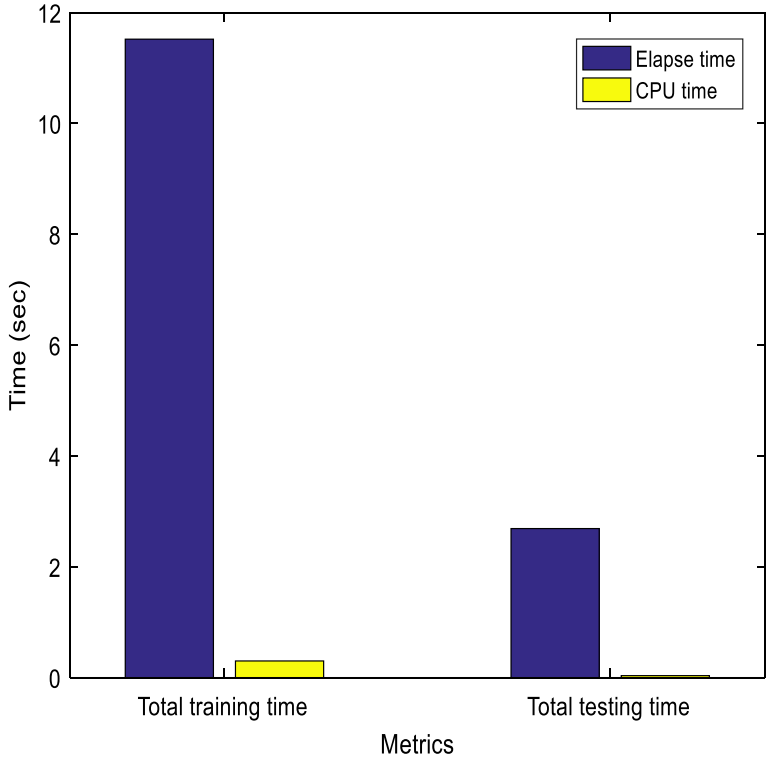

Fig. 4 Training and Testing time evaluation

Table 8 Performance metrics comparison based on attack categories

Categories	Precision	Recall	F1-score	FAR	Accuracy
Normal	0.947	0.995	0.968	0.015	98.86%
Attacks	0.999	0.987	0.993	0.007	
Weighted average	**0.986**	**0.987**	**0.989**	**0.008**	

works efficiently for predicting intrusion over the network with least FAR and higher prediction accuracy.

5 Conclusion

In this work a novel Fuzzy Genetic Algorithm with Multi-Objective Particle Swarm Optimization model is designed for predicting the normal traffic and evaluation time. It includes both the minor or major attack categories specifically for the rare information

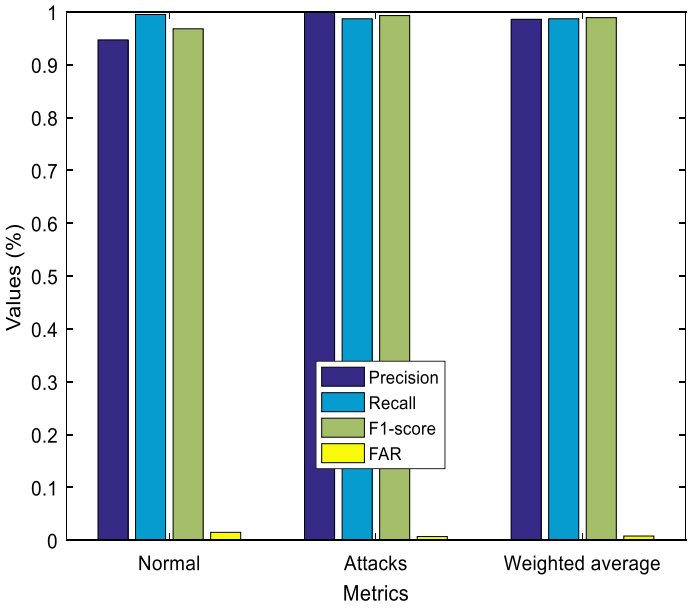

Fig. 5 Performance metrics comparison based on attack categories

Table 9 Weighted average measure of attack categories

Categories	Precision	Recall	F1-score	FAR	Accuracy
Normal	0.9956	0.9960	0.9975	0.16190	98.25%
DoS	0.9940	0.9790	0.9860	0.00450	
Probe	0.8600	0.8855	0.8780	0.00350	
R2L	0.6920	0.9195	0.7895	0.00550	
U2R	0.8880	0.5715	0.6965	0.00002	
Weighted average	**0.99**	**0.9886**	**0.9988**	**0.0996**	

from the provided NSL-KDD dataset. This model includes three essential steps like feature selection, classification and optimization approaches for properly interpreting the accuracy of the given dataset to facilitate human understanding and data analysis. The proposed model is contrasted with several existing approaches. Experimental results illustrate that the proposed model effectively extracts the appropriate rule-based model from network traffic, largely benefiting from the assistance provided by MOPSO. Moreover, certain performance metrics are assessed, revealing how well the proposed model performs in meeting the objectives of the exploitation and exploration criteria, rule evolution, and detection of attack categories with superior detection rate and least FAR compared to other approaches. However, the model attains 98.86% accuracy, 9.5% FAR, 99% precision, 98.86% recall and 99.88% F1-score respectively.

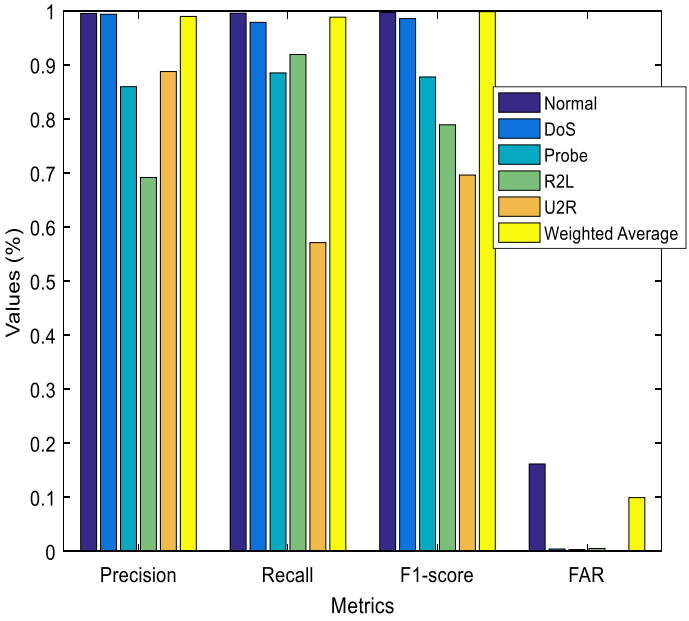

Fig. 6 Weighted average measure of attack categories

Table 10 Average execution time (ms)

Approaches	Training (ms)	Testing (ms)
MOPSO	95.4565	2.5465
PSO with lightweight GBM	189.030	3.0515
Decision tree	100.4567	4.8954
Logistic regression	219.5648	9.7895

The resourceful classification and detection of the primitive normal network traffic and intrusion attacks offer predominant scope in the future. Based on these models, the improved approach is applied to diverse complex problem-based domains like DNA computation. Additionally, with respect to this domain, some optimization approaches are candidate to be used to attain superior accuracy.

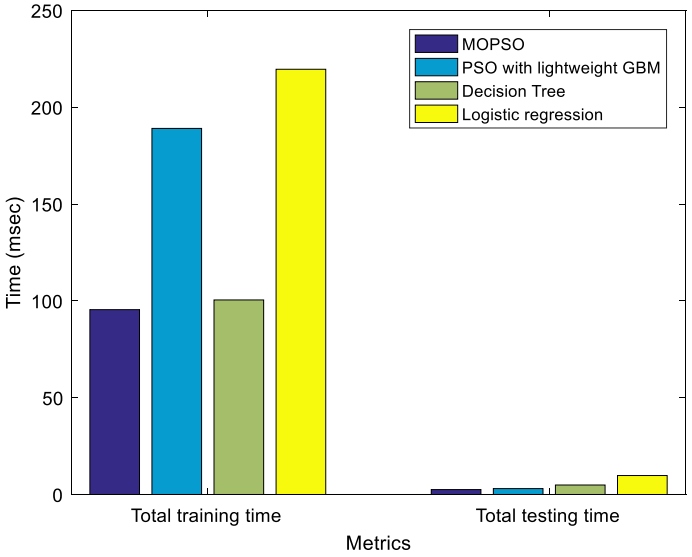

Fig. 7 Average execution time (ms)

Authors Contributions Both the authors Arun Kumar Ramamoorthy and K.Karuppasamy contributed to the study conception and design. Material preparation, data collection and analysis.

Funding The authors have not disclosed any funding.

Data Availability The data sets used in this article are openly available in the name NSLKDD dataset at www.unb.ca/cic/datasets/nsl.html

Declarations

Conflict of interest Dr.Arun Kumar Ramamoorthy declares that he has no conflict of interest. Dr. K.Karuppasamy declares that he has no conflict of interest.

Ethical Approval This article does not contain any studies with human participants performed by any of the authors.

References

1. Aljawarneha, S., Aldwairiab, M., & Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computer Science, 25*, 152–160.
2. Al-Yaseen, W. L., Othman, Z. A., & Nazri, M. Z. (2017). Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Systems with Applications, 67*, 296–303.
3. Huang, J. Y., Liao, I. E., Chung, Y. F., & Chen, K. T. (2011). Shielding wireless sensor network using Markovian intrusion detection system with attack pattern mining. *Information Sciences, 231*, 32–44.
4. Lee, H., & Kim, E. (2015). Genetic outlier detection for a robust support vector machine. *International Journal of Fuzzy Logic Intelligent Systems, 15*(2), 96–101.

5. Osamaa, A., El-said, S. A., & Hassanien, A. E. (2016). Optimized hierarchical routing technique for wireless sensors networks. *Soft Computing, 20*, 4549–4564.
6. Li, J., Zhang, W., & Lun, L. K. (2010). A novel semi-supervised SVM based on tri-training for intrusion detection. *Journal of Computers, 5*(4), 638–645.
7. Palvinder, S. M., & Satvir, S. (2019). Improved artificial bee colony metaheuristic for energy-efficient clustering in wireless sensor networks. *Artificial Intelligence Review, 51*, 329–354.
8. Urtnasan, E., Park, J. U., Lee, S. Y., & Lee, K. J. (2017). Optimal classifier for detection of obstructive sleep apnea using a heartbeat signal. *International Journal of Fuzzy Logic Intelligent Systems, 17*(2), 76–81.
9. Borkar, G. M., Patil, L. H., Dalgade, D., et al. (2019). A novel clustering approach and adaptive SVM classifier for intrusion detection in WSN: A data mining concept. *Sustainable Computing Informatics and Systems, 23*, 120–135.
10. Huang, S. H., Chen, W. Z., & Li, J. (2017). Network intrusion detection based on extreme learning machine and principal component analysis. *Journal of Jilin University, 35*(5), 576–583.
11. Liang, W., Tang, M., Long, J., Peng, X., Xu, J., & Li, K.-C. (2019). A secure fabric blockchain-based data transmission technique for industrial internet-of-things. *IEEE Transactions Industrial Informatics, 15*(6), 3582–3592.
12. Shone, N., Ngoc, T. N., Phai, V. D., et al. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence, 2*(1), 41–50.
13. Yin, C., Zhu, Y., Fei, J., et al. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access, 5*(2), 21954–21961.
14. Wang, C. R., Xu, R. F., Lee, S. J., et al. (2018). Network intrusion detection using equality constrained-optimization-based extreme learning machines. *Knowledge Based Systems, 147*, 68–80.
15. Osanaiye, Alfa, A. S., & Hancke, G. P. (2018). Denial of service defence for resource availability in wireless sensor networks. *IEEE Access, 6*, 6975–7004.
16. Li, P., Zhao, W., Liu, Liu, X. and Yu, L. (2018). Poisoning machine learning based wireless IDSs via stealing learning model. In *Proceedings of International Conference on Wireless Algorithms, Systems and Applications*, pp. 261–273.
17. Zhou, Y., Liu, Y. Wang., & Tian, Z. (2019). Anonymous crowdsourcing-based WLAN indoor localization. *Digital Communications and Networks, 5*(4), 226–236.
18. Almomani, Al-Kasasbeh, B., & Al-Akhras, M. (2016). WSN-DS: A dataset for intrusion detection systems in wireless sensor networks. *Journal of Sensors, 2016*, 1–16.
19. Ioannou, Vassiliou, V. and Sergiou, C. (2017). An intrusion detection system for wireless sensor networks. In *Proceedings of 24th International Conference on Telecommunications (ICT)*, pp. 1–5.
20. Ghosal, & Halder, S. (2017). A survey on energy efficient intrusion detection in wireless sensor networks. *Journal of Ambient Intelligence and Smart Environments, 9*(2), 239–261.
21. Almomani, & Alenezi, M. (2018). Efficient denial of service attacks detection in wireless sensor networks. *Journal of Information Science and Engineering, 34*(4), 977–1000.
22. Ke, Meng, Q., Finley, T.,Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 3146–3154.
23. Arun Kumar, R., & Karuppasamy, K. (2022). Integration of fuzzy with incremental import vector machine for intrusion detection. *International Journal of Computers Communications & Control, 17*(3), 4481.
24. Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. *Cyber-Security, 2*(1), 20.
25. Ali, Al Mohammed, B. A. D., Ismail, A., & Zolkipli, M. F. (2018). A new intrusion detection system based on fast learning network and particle swarm optimization. *IEEE Access, 6*, 20255–20261.
26. Buczak, & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys and Tutorials, 18*(2), 1153–1176.
27. Peng, Leung, V. C. M., & Huang, Q. (2018). Clustering approach based on mini batch kmeans for intrusion detection system over big data. *IEEE Access, 6*, 11897–11906.
28. Ibrahim, Basheer, D. T., & Mahmod, M. S. (2013). A comparison study for intrusion database (Kdd99, Nsl-Kdd) based on self-organization map (SOM) artificial neural network. *Journal of Engineering Science and Technology, 8*(1), 107–119.
29. Divekar, Parekh, M., Savla, V., Mishra, R. and Shirole, M. (2018). Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives. In *Proceedings of IEEE 3rd International Conference on Computing, Communications and Cyber-Security. (ICCCS)*, pp. 1–8.
30. Liu, Y., Fu, J.-S., & Zhang, Z. (2016). K-nearest neighbors tracking in wireless sensor networks with coverage holes. *Personal and Ubiquitous Computing, 20*(3), 431–446.

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Dr. Arun Kumar Ramamoorthy was born in India in 1988. He received his B.Eng (CSE) from Karunya University, Coimbatore, India, his M.Tech (IT) from Anna University, Chennai, and his Ph.D. from Anna University, Chennai. He is currently working as a Lecturer in Digital Forensics & Cyber Security at the University of South Wales, Treforest, United Kingdom. He has 14 years of academic experience in the field of Information Technology. He has published more than 15 articles in Information Security/Network Security. His research interests include Information Security, Digital Forensics, Mobile Application Security, Artificial Intelligence, Deep Learning, and Network Security. He has also conducted Guest Lecture programs and workshops on various topics, including Digital Forensics, Security Fundamentals, Network Protection Strategies, and Security Attacks on LLMs, at various universities and engineering colleges. He is a member of the Computer Society of India (CSI) and the Indian Society of Technical Education (ISTE).

Dr. K. Karuppasamy was born in India in 1978. He received his B.E (CSE), M.E (CSE), and Ph.D. from Anna University, Chennai. With 18 years of academic experience in the field of Computer Science and Engineering, he currently serves as Professor & Head at RVS College of Engineering and Technology, Coimbatore, India. His extensive research background led him to become a research supervisor at Anna University, Chennai, and inspired him to establish a research lab for innovation and development at RVSCET. His primary areas of interest include wireless sensor networks, Mobile AdHoc Networks, Network Security, and Mobile Computing. He has authored over 50 articles in various journals and has mentored research scholars and postgraduate students. Recognized for his outstanding contributions to academic excellence, he has received numerous awards and accolades from Anna University. Additionally, he actively participates as a member of CSI, ISTE, and other academic bodies.