COMPUTERS & SECURITY 110 (2021) 102448

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

An effective genetic algorithm-based feature selection method for intrusion detection systems

Zahid Halim[a,*], *Muhammad Nadeem Yousaf*[a], *Muhammad Waqas*[b,d], *Muhammad Sulaiman*[a,e], *Ghulam Abbas*[b], *Masroor Hussain*[a], *Iftekhar Ahmad*[c], *Muhammad Hanif*[a]

[a] *Machine Intelligence Research Group (MInG), Faculty of Computer Science and Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi 23460, Pakistan*

[b] *Telecommunications and Networking (TeleCoN) Research Lab, Faculty of Computer Science and Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi 23460, Pakistan*

[c] *School of Engineering, Edith Cowan University, Joondalup, WA 6027, Australia*

[d] *Engineering Research Center of Intelligent Perception and Autonomous Control, Faculty of Information Technology, Beijing University of Technology, Beijing, 100124, People's Republic of China*

[e] *Department of Computer Science, Capital University of Science and Technology, Islamabad, Pakistan*

ARTICLE INFO

Article history:
Received 25 October 2020
Revised 8 July 2021
Accepted 13 August 2021
Available online 26 August 2021

Keywords:
Feature selection
Genetic algorithm
Intrusion detection
Machine learning
Data analysis

ABSTRACT

Availability of suitable and validated data is a key issue in multiple domains for implementing machine learning methods. Higher data dimensionality has adverse effects on the learning algorithm's performance. This work aims to design a method that preserves most of the unique information related to the data with minimum number of features. Addressing the feature selection problem in the domain of network security and intrusion detection, this work contributes an enhanced Genetic Algorithm (GA)-based feature selection method, named as GA-based Feature Selection (GbFS), to increase the classifiers' accuracy. Securing a network from the cyber-attacks is a critical task and needs to be strengthened. Machine learning, due to its proven results, is widely used in developing firewalls and Intrusion Detection Systems (IDSs) to identify new kinds of attacks. Utilizing machine learning algorithms, IDSs are able to detect the intruder by analyzing the network traffic passing through it. This work presents parameter tuning for the GA-based feature selection along with a novel fitness function. The present work develops an enhanced GA-based feature selection method which is tested over three benchmark network traffic datasets, namely, CIRA-CIC-DOHBrw-2020, UNSW-NB15, and Bot-IoT. A comparison is also performed with the standard feature selection methods. Results show that the accuracies improve using GbFS by achieving a maximum accuracy of 99.80%.

1. Introduction

In the current digital era having numerous connected devices, one of the primary issues at hand is to ensure the security and privacy of individuals' and organizations' data. As the technology is evolving, it is creating a number of threats and challenges in regard with the security concerns. In recent years a variety of attacks and malware samples have been observed by the research community. The term cybersecurity and the

* Corresponding author.
E-mail addresses: zahid.halim@giki.edu.pk (Z. Halim), muhammad.waqas@giki.edu.pk (M. Waqas), abbasg@giki.edu.pk (G. Abbas), hussain@giki.edu.pk (M. Hussain), i.ahmad@ecu.edu.au (I. Ahmad), muhammad.hanif@giki.edu.pk (M. Hanif).
https://doi.org/10.1016/j.cose.2021.102448
0167-4048/

information security are usually used interchangeably. However, the cybersecurity goes beyond the defined boundaries of information security (Von Solms and Van Niekerk, 2013). It refers to the protection of information as well as other devices from intrusion. The term "threat intelligence" has been introduced to assist the security practitioners and analysts to identify the cyber-attack in computer networks. Cyber Threat Intelligence (CTI) is defined as the process of collecting a set of data, applying certain statistical analysis regarding security threats and identify the risk actors, exploits, malware types, and defining defense mechanism (Riesco et al., 2020). Through the CTI, the security analysts are able to collect data, identify the threat and response back with a stronger defensive approach in a timely manner to neutralize the cyber-attack before it exploits system vulnerabilities. The challenge is how to provide such intelligence to the systems for analyzing and reacting against the cyber-attacks. In this regards, the research community has developed methods demonstrating the real-time working and securing the individual and organizational data.

Identifying and neutralizing attackers is a challenging task because of the fact that the attackers adopt different (and unique) ways to launch an offense on the victim. This is done either to collect personal information, like the financial statements, passwords, and contact details, or to take access of a victim's device to operate the malicious activities. The malicious activities include delivering malwares using botnets or locking the useful information from a legitimate user like ransomware. The life cycle of various cyber-attacks is more or less similar to each other. It starts from victim reconnaissance and ends at exploiting vulnerabilities to perform the desired task (Conti et al., 2018). The three main aspects of a cybersecurity defensive approaches mentioned in Giraldo et al. (2017) are: prevention, detection, and response.

1.1. *Intrusion detection system*

Intrusion Detection System (IDS) is the main component of a security system which aims to strengthen system security. Its task is similar to what the antivirus or firewalls are supposed to do. The main purpose of IDS is to identify the malicious activity that occurs in a computer network and to provide a mechanism to cope up with network attacks (Tribak et al., 2012). The IDS is able to maintain the security by blocking an attack when it occurs (Haq et al., 2015). Currently, the major issue with the IDS is that they generate high False Positive (FP) and False Negatives (FN) alarms (Gamal et al., 2020). Therefore, there is a need of implementing such techniques on the collected data that helps the IDS to increase its accuracy (Tausif et al., 2017). There are two main categories of an IDS, namely, network-based (NIDS) and host-based (HIDS) (Stampar and Fertalj, 2015).

NIDS is placed at points where it can conveniently monitor the incoming traffic. Whenever, there exist any malicious packets that comes from outside the network, it detects and apply preventive measures or alerts the administrator. Whereas, HIDS runs as an agent on the individual devices. It monitors the local device's operating system and the network traffic. HIDS can only detect the intrusion attacks on the device where it is installed (Carlin et al., 2015). The present work focuses on the NIDS. Moreover, on the basis of detection mechanisms NIDS can be of two types; signature-based and anomaly-based. The signature-based NIDS maintain the database having a proof data pattern that matches with certain intrusion attacks (Tu et al., 2021; Wan et al., 2021). The signature-based NIDS is also known as knowledge-based IDS. It detects the malicious packets from the traffic and matches those with the pre-stored signature. If any match is found, it triggers the alarm or takes a predefined preventive measure. However, in the anomaly-based IDS, there is a baseline that defines the normal behavior of the network where the IDS is deployed. Finding this baseline is important that is why implementing anomaly-based IDS is more challenging than the signature-based IDS deployment. However, the anomaly-based IDS can easily detect the deviated behavior of the network caused by any intrusion. Anomaly-based IDS is better in detecting the novel and unknown attacks (Carlin et al., 2015). It detects the violation and matches it with the baseline before triggering an alarm. Within the NIDS category, the present work focuses on the signature-based IDS (Protić, 2018).

For a responsive behavior, IDS has two reaction modes, i.e., passive mode and reactive mode. If the IDS is not defined with any reactive approach then it only detects the threat and informs the network security administrator. The reaction is to eliminate the threat and make the system safe. Such type of IDS is categorized as passive one. In contrast, reactive type of IDS not only identify the threat, but they take the reactive measures. These measures are either in the form of blocking such malicious packets to enter the network or to react in any other way defined by the administrator and referred to as Intrusion Detection and Prevention System (IDPS) (Stampar and Fertalj, 2015).

1.2. *Machine learning-based intrusion detection systems*

A challenge in detecting the cyber-attack is to identify the kind of attack that occurred and the vulnerabilities that can be exploited through that. As the security is moving towards gaining more strength, cyber criminals are using smarter and innovative methods to attack the victims. These methods range from using files of different formats (like, word documents, PDFs, images) (Elingiusti et al., 2018) to encapsulate the attack that the attacker needs to injecting victim's device to more adverse attacks like spreading ransomware which has a worm-like behavior that can infect hundreds of individual devices. Such advancements in attack methodologies make the identification of cyber-attack a more challenging issue. To address the issues of cybersecurity, the emerging field of CTI considers the use of machine learning-based security tools to act against the cyber-attacks. During recent years, the research community has shown significant research regarding implementation of Machine Learning (ML) in the domain of cybersecurity to recognize CTI (Afzaliseresht et al., 2020; Alloghani et al., 2020). Machine learning and data mining approaches (Halim et al., 2021) are widely used now to enhance the efficiency of IDS because of their proven efficiency in malware detection and classification (Shalaginov et al., 2018). For implementing the machine learning approaches, one need to train the model that can help in classification of malicious and normal traffic. For that, the data plays an impor-

tant role. Without the quality data, the machine learning models cannot be efficiently trained. Therefore, this work utilizes three popular intrusion detection datasets. This work tests the proposed technique against three classifiers, namely, *k*-Nearest Neighbor (*k*-NN), Support Vector Machine (SVM), and XgBoost. Additionally, a comparison with four feature selection techniques, namely, recursive feature elimination, sequential feature selector, correlation-based feature selection, and select*K*best has also been performed. Other than this, the proposed solution is also compared with three closely related state-of-the-art methods.

1.3. *Feature selection*

Today, handling the Big Data for machine learning purposes is an essential undertaking for most of the domains including cybersecurity. Such rapid growth of the data in the field of security needs effective and efficient way to manage it. For high-dimensional datasets, Data Mining (DM) and ML techniques focus on extracting useful insights by reducing the dataset features. For DM and ML approaches to be implemented the critical issue is the curse of dimensionality. Dimensionality problem refers to the state where the data is scattered in high dimensional space and has an adverse effect on the learning techniques which are designed for low-dimensional space (Liu and Tang, 2013). Another issue is overfitting that adversely influences the machine learning model's accuracy when there exists large number of features in the data. Moreover, having large number of features mean more memory requirement and computational cost. The best way to handle the issue of high dimensionality is to reduce the dimensions of a given dataset, referred to as dimensionality reduction. Dimensionality reduction can be done through feature selection (Xue et al., 2021, 2019, 2016). This mechanism transforms the large number of features of the big data into a new feature space with low dimensionality. Feature selection refers to the selection of most suitable feature subset from the given input feature vector that helps the machine learning model to train efficiently. In the real-world scenario, the datasets comes with noise which adds redundant and irrelevant features. Removing such noise from the data can positively influence the learning rate and the detection accuracy of the classifier while decreasing the FPs and FNs (Ho, 2006).

The feature selection methods can be classified as supervised and unsupervised. The supervised feature selection techniques are generally designed for the purpose of classification or for the regression problems. The purpose of such techniques is to extract the feature subset from the given original features based on the capability of discriminating between available classes of data or to estimate the targets in the regression analysis. Unsupervised feature selection techniques are useful for the clustering problem. Unlike the supervised way of feature learning where the correlation is calculated between the features and the class label, there is an alternative criteria to define the feature relevance (Halim and Rehan, 2020).

1.4. *Problem statement*

On the basis of different selection strategies, there are three categories of feature selection methods, namely, wrapper, filter, and embedded techniques. The first two categories work in a different fashion while the later one uses both the strategies together. In wrapper method, the approach rely on the predictive performance of a defined learning algorithm in the solution to a specific problem. Using that performance, the selected features are evaluated. The wrapper method has two steps. First, it searches for a feature subset and in the second step it evaluates the selected features using the learning algorithm which act as a black box. These steps are iteratively performed until a predefined stopping criteria is met. The issue with the wrapper method is that the search space for any n features is $2n$, which is a challenge for the datasets with very large dimensions. To cope up with this, different strategies, like best-first search, hill-climbing, branch-and-bound search and genetic algorithms can help to yield a locally optimum learning performance. The filter methods of feature selection are independent of the learning algorithms. These are efficient than the wrapper methods. However, the selected features may not be the optimal because there exist no specific learning algorithm guidance. The filter methods have two steps. First, the features are ranked according to some ranking criteria. The feature ranking method can be univariate where each feature is ranked individually or it can be multivariate where multiple features are ranked in a batch way. In the second step, it extracts the features with the mentioned ranking criteria (Liu and Motoda, 2007).

This work aims to address the issue of selecting a subset of original features that can have higher positive influence on the detection accuracy of an IDS. In the presence of class label, the technique of feature selection gets a bit easier by calculating the effect of each feature on the prediction of the class label. However, even in the presence of class labels (supervised learning) feature selection can be performed through an unsupervised way. This method ensures that the feature subset of original attributes is the optimal number of features and can attain higher detection accuracy.

Selecting the optimal representative features using machine learning is a crucial step. Dimensionality reduction is implemented to reduce the number of features by removing the irrelevant and duplicated features. During this process of finding the optimal number of features from data having large number of features, the computation cost is higher. This work implements a Genetic Algorithm (GA) to search for the optimum features on the basis of which the performance of classification can be improved. While implementing the GA for the feature optimization problem, this work also preforms parameter tuning. The present proposal also devices a novel fitness function for the task at hand. On the basis of the tuned parameters, the GA converges quickly and provides the optimum features having higher detection accuracies.

1.5. *Key contributions and novelty*

The present work contributes an evolutionary computing-based solution for the IDS, where the machine learning classifiers are trained using the features identified by the proposed solution, i.e., GbFS to improve the prediction accuracy. Selecting the optimum features for machine learning is a crucial step. Dimensionality reduction is implemented to reduce the number of features by removing the irrelevant and dupli-

cate attributes. During this, identifying the optimum number of features from data having large number of attributes has a high computational cost. This work develops a GA-based solution to search for the optimum features on the basis of which classifications are tuned for better prediction of an intruder. The present work also contributes a novel fitness function that acts at the core of GA to identify better features from the network traffic data. The proposed fitness function is based on the computation of correlation between the selected features in absence of the actual class labels. On the basis of selected parameters through the proposed fitness function, the GA converges quickly and provides the optimum features enabling higher detection accuracy. The key contributions of this work are summarized in the following.

- Development of a novel fitness function of the GA to rank features.
- Development of a novel evolutionary computing-based feature selection technique, GbFS, for intrusion detection systems.
- Training of machine learning classifiers utilizing the optimum features selected though the developed GA module.
- Performance evaluation on benchmark datasets.
- Random variables' selection process replaced with the GA-based guided values to improve quality of final solution.
- Effectiveness of GbFS demonstrated through a comparison with state-of-the-art intrusion detection methods and standard feature selection techniques.
- Proposed strategy effectiveness demonstrated by comparing all features' results with the selected features results.

The rest of the paper is organized as follows. Section 2 presents the background knowledge and the related work on feature optimization using GA. Section 3 explains the proposed methodology and the parameter including the selection process, fitness function, and crossover and mutation rates. Section 4 presents the experimental results. Finally, Section 5 concludes this work and presents a few future directions.

2. Related work and problem formulation

Due to the exponential increase in the volume of the network traffic, the need for an efficient feature selection method has grown. This has the utility to develop an IDS having an optimal level of detection accuracy and low false alarm rate. Evolutionary Algorithms (EA) are bio-inspired methods which are based on the Darwinian's principle (Wang et al., 2020). This section presents the problem formulation and literate review on feature selection methods using EAs specifically for the IDS.

2.1. *Literature review*

The work in Stein et al. (2005) introduces GA-based feature selection method. The authors use Decision Trees (DTs) with the GA. They apply GA as the feature selection method on the KDD dataset and classify the test data using the DT classifier. The initial population is generated randomly, which is passed on for the feature selection process. They implement the ranked-based selection and a two point crossover to produce new population. Their approach implements bit-level mutation of the offspring and then keeps the two elite parents and replaces the rest of the population. The fitness of the chromosomes is evaluated using the sum of the validation error rates as the fitness function. Their work presents 32 optimum features out of 41 total attributes of the KDD dataset. The work in (Ho, 2006) attempts to develop an anomaly-based IDS to detect novel attacks through unsupervised learning and a bio-inspired and stochastic clustering model referred to as Ant Colony Clustering Model (ACCM). Regarding the supervised learning, the authors propose a multi-objective genetic fuzzy intrusion detection mechanism. Their technique acts like a genetic feature selection wrapper method to search for the optimal number of features that can represent most of the information related to the data. For evaluating the detection accuracy, they present 27 features as the optimum ones out of the 41 features of the KDD dataset. Their work archives an accuracy of 99.24% on the KDD network traffic dataset.

In Ahmad et al. (2011) the authors present a feature selection approach using the GA, Principal Component Analysis (PCA) and Multilayer Perceptron (MLP) for the intrusion detection. The aim of their research is to enhance the detection rate of the classifiers for intrusion detection. They compare their approach with the use of simple PCA. They apply GA to search for the principal feature space that has optimal sensitivity with the classifier. The proposal is evaluated using three different experiments with 12, 20 and 27 features having the highest accuracy of 99% with 12 features subset. Another work done in Sindhu et al. (2012) use the GA as the feature selection technique. They initialize the optimization process by generating random population and through the computation of chromosome's fitness, they generate the population for the next generation. The fitness function takes the feature count, sensitivity, and specificity under consideration to evaluate the feature subset. They perform a comparison with other feature selection techniques and achieve an accuracy of 98.38% with 16 features extracted from 41 original features of the KDD dataset.

The work in Kuang et al. (2014) presents a novel SVM model with Kernel Principal Component Analysis (KPCA) and the GA. Their methodology is implemented for detecting the normal and malicious network traffic. They test their methodology on KDD cup dataset. The GA is used in their work to optimize the parameters for the SVM and KPCA to reduce the dimensionality of feature space. By selecting 12 optimal features they get the detection accuracy of 94.22% and also a fast convergence occurs with better generalization. The work in Aslahi-Shahri et al. (2016) evaluates the performance of SVM classifier on intrusion detection dataset. According to the authors, SVM does not get higher accuracies while working in isolation. This is because the SVM needs a dataset with a proper pattern and selects optimal features with minimum redundancy. Due to this, the authors implement a GA to search the optimal features for the SVM and then perform the classification on intrusion detection dataset. An ensemble feature selection using bi-objective generic algorithm is proposed in Das et al. (2017). The authors address the problem of selecting optimal features for data mining. They merge the two concepts to-

gether to develop a bi-objective genetic algorithm that are the boundary region analysis of rough set theory and the multivariate mutual information. Their method is tested across the well-known datasets and evaluated for performance. Among other datasets, they have used the spambase dataset to classify the spam and legitimate emails and obtained the best accuracy of 92.6%.

The work in Gharaee and Hosseinvand (2016) propose an IDS that utilizes the GA for feature selection with an innovative fitness function. They achieve high prediction accuracy while maintaining a low false positive rate. They proposal is tested on KDD cup and UNSW-NB 15 datasets. They report the accuracies for each class in their paper. They also generate a separate dataset for each class and then apply their technique. The work in Xu et al. (2018) present a feature selection method based on an improved binary Whale Optimization Algorithm (WOA) for network intrusion detection. They test the technique on KDD dataset and report the results. In their work, WOA converges slowly and may fall into the local optima during updating mechanism that has an adverse effect on classification. They compare their mechanism with the GA and report higher accuracy of their technique. On KDD cup dataset, their technique selects 5 out of 41 features and attains an accuracy of 97.89%. Whereas, the GA selects 11 out of 41 features with an accuracy of 95.58%. The work in Yousefi-Azar et al. (2017) propose the use of autoencoders as the generative model for the purpose of feature learning. They explain how the autoencoder is capable of learning the latent representation and the semantic similarity between the features of the dataset. Their technique is tested for the intrusion detection as well as for the malware classification. For this purpose, they take KDD cup dataset and Microsoft Malware Classification Challenge (BIG 2015) dataset. For intrusion detection, they report the best results having 83.3% of accuracy with Gaussian naïve Bayes classifier. Table 1 lists the key features of the past works and the current proposal.

The work in Tahir et al. (2021) also presents a feature selection method based on GA by incorporating chaotic maps. Their solution is tested on the data from the affective computing (Halim et al., 2021) and healthcare systems. Chaotic maps, in their work, are applied to the initial population of the GA which is followed by the reproduction operations to produce the optimum set of features. Their method is evaluated on the seven class emotion identification problem. The work in Viharos et al. (2021) address the task of integrating the most appropriate attribute identification technique for a given problem to attain optimum feature ordering. Their proposal basically is a fusion of multiple feature selection methods to obtain a generalized solution. Experiments in their work are performed using UCI repository data and a couple of real-life datasets. The work of Nouri-Moghaddam et al. (2021) present a wrapper feature selection method which is based on a multi-objective forest optimization scheme. The Pareto front in their solution is maintained through the archive, grid, and region-based selection schemes. Their algorithm is named as multi-objective wrapper method based on Forest Optimization (MOFOA). In addition to experiments on the UCI repository data, they perform evaluation on two microarray datasets (Uzma et al., 2021). A feature selection approach for the network intrusion identification is presented in Li et al. (2021). Their solution is based on Krill Herd (KH) algorithm which is a swarm intelligence technique. Linear nearest neighbor lasso step optimization is performed in their solution to update the krill herd position in the search space. This enables to derive the global optimal solution. Similarly, the proposal by Dwivedi et al. (2021) is a swarm intelligence-based contribution for the IDS. The grasshopper algorithm from the domain of swarm intelligence is utilized in their work which is integrated with ensemble feature selection approach.

A two-layer feature selection approach is presented in Amini and Hu (2021). The first layer of their solution is the GA-based wrapper, whereas, the second one is the Elastic Net embedded scheme. Key aim of their solution is to enhance the prediction accuracy. The second layer address the optimality issues caused by the GA in the first layer. Performance of their solution is evaluated using Maize genetic dataset from NAM population. The solution is named as two-layer wrapper-embedded (GA-EN). The obtained results suggest that GA-EB results in smaller root mean square error with different feature space dimensions when compared with embedded wrapper method. A GA-based enhanced feature selection scheme is presented in Maleki et al. (2021). Their work utilized k-NN for lung cancer prognosis integrated with an evolutionary computing-based feature selection. A genetic algorithm in their work is adopted as a hybridized approach for an efficient attribute selection. Their proposal is evaluated on a lungs cancer dataset yielding better performance. The work in Guo et al. 2(021) present a new feature selection approach utilizing long short-term memory network. The proposal is tested for the prediction of surface roughness as a case study. Their work extracts multiple features in the time and frequency domains from original and decomposed signals. Obtained results in their work suggest better performance of the deep learning methods on the selected task. The work in Sumaiya Thaseen et al. (2021) present an integrated intrusion detection system. Their solution utilizes correlation-based attribute selection integrated with the artificial neural network. Their solution is a machine learning-based framework (Halim et al., 2020) for the IDS. The correlation-based feature selection enables to rank attributes according to the highest correlation value between the attributes and the ground truth. A feature selection technique to detect malware from Android is presented in Mahindru and Sangal (2021). Their solution is based on machine learning methods. The machine leaning module in their work is based on Least Square Support Vector Machine (LSSVM) which is evaluated through three kernels, namely, linear, radial basis function, and polynomial. For performance evaluation, two million distinct Android apps are utilized.

2.2. *Problem formulation*

The IDS play a critical role in an organizational network. IDS is the guard that protects the network from the outside attacks. Machine learning due to its promising results is widely used in the development of IDS. As the network traffic is generated with high dimensions, there can be an adverse effect on the detection accuracy of the machine learning models. The best way to cope up with the curse of dimensionality is to reduce the features while preserving the information and

Table 1 – Key features of the past works and present contribution.

Reference	Methodology	Dataset	Feature length	Classifier	Performance
(Stein et al., 2005)	GA	Intrusion detection dataset KDD CUP'99	32	Decision tree	Detection error rate: 0.095 for U2R and 19.9 for R2L
(Ho, 2006)	GA	Intrusion detection dataset KDD CUP'99	27	Multiobjective genetic fuzzy rule classifier	Accuracy: 99.24%
(Ahmad et al., 2011)	PCA + GA	KDD CUP'99	12	Multilayer perception	Accuracy: 99%
(Sindhu et al., 2012)	Neural Network + GA	KDD CUP'99	16	Decision Tree	Accuracy:: 98.38%
(Kuang et al., 2014)	Genetic algorithm with Hybrid kernel principal component analysis	KDD CUP'99	12	Multilayer SVM	Accuracy: 94.22%
(Aslahi-Shahri et al., 2016)	Hybrid method of GA and SVM	KDD CUP'99	10	SVM	True Positive rate: 0.97
(Das et al., 2017)	Ensemble bi-objective GA for feature selection	Spambase	-	Decision Tree	Accuracy: 92.6%
(Gharaee and Hosseinvand, 2016)	GA	KDD CUP'99, UNSW-NB 15	-	SVM	-
(Xu et al., 2018)	Improved binary whale optimization algorithm	KDD CUP'99	5	-	Accuracy: 97.89%
(Yousefi-Azar et al., 2017)	Auto-encoder based feature learning	KDD Cup'99	-	Naïve Bayes	Accuracy: 83.3%
(Proposed)	GA-based feature selection	KDD Cup'99, UNSW-NB15, and Bot-IoT	10, variable	SVM, k-NN, and XgBoost	Accuracy: 99.80%

the reduced features should provide aid in enhancing the detection accuracy of the classifier. Numerous feature selection techniques are proposed and they are working quite well too but the process of enhancing the techniques still continues. Extracting the useful features from the dataset can be done with better efficiency in regard with increasing detection accuracy through the unsupervised fashion. Using unsupervised feature selection techniques, the computations become complex because of the larger number of combinations that need to be evaluated before selecting the best feature subset. The problem addressed here is to device an improved unsupervised feature selection technique using EAs that yields better detection accuracies with high TP and low FP.

3. Evolutionary feature selection for intrusion detection

This section presents the proposed solution for the feature selection problem using an evolutionary algorithm focusing to improve the classifiers' performance for an IDS. The proposed methodology implements GA for unsupervised feature selection and observes an improvement in the detection accuracy. The architecture of the proposed solution is represented at a higher level of abstraction in Fig. 1. The block diagram shows input as the network traffic data and the final output is the enhanced detection rate of attack and normal traffic. The whole process consists of three phases, these include, (a) dataset preparation, (b) GA-based learning module, and (c) classification module to predict a threat. The present proposal utilizes GA for the optimization task based on the earlier mentioned fitness function. There are multiple other optimization and search space exploration techniques, like Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). The GA has an advantage of searching from a population of points instead of a single point approach. The GA also has an advantage of utilizing a payoff (i.e., the objective function) information and not relying on the derivatives. This enables to achieve an optimum solution. Another advantage of the GA is its ability to support multiple objectives. All these pros of the GA are kept in view while devising the proposed solution. Whereas, the techniques like PSO and ACO can fall into local optimum in a high-dimensional space and suffers from a low convergence rate in the iterative process. Due to this reasoning the present work has opted for the GA to address the problem at hand.

3.1. *Datasets preparation*

Data preparation is the first step of any learning system. The proposed system records the incoming data to the system and stocks it in its data store. The data pre-processing step is important to clean data from noise and irregularities before it is presented to the learning module. The data pre-processing includes: input output coding, normalization, and scaling. In fact, some data may not be in the form suitable for the learning module to operate on, since the present solution only accepts data in numerical form and produces output in a specific range depending on the fitness function. Input values of various data may not be on the same scale. To handle this, scaling is required. This work scales all the attributes using Eq. (1).

$$x_i' = \frac{(x_i - X_{min})}{(X_{max} - X_{min})} \tag{1}$$

Where, $x_i, x_i', X_{max}, X_{min}$ are the original values, scaled value, maximum, and minimum, respectively. The attributes are first

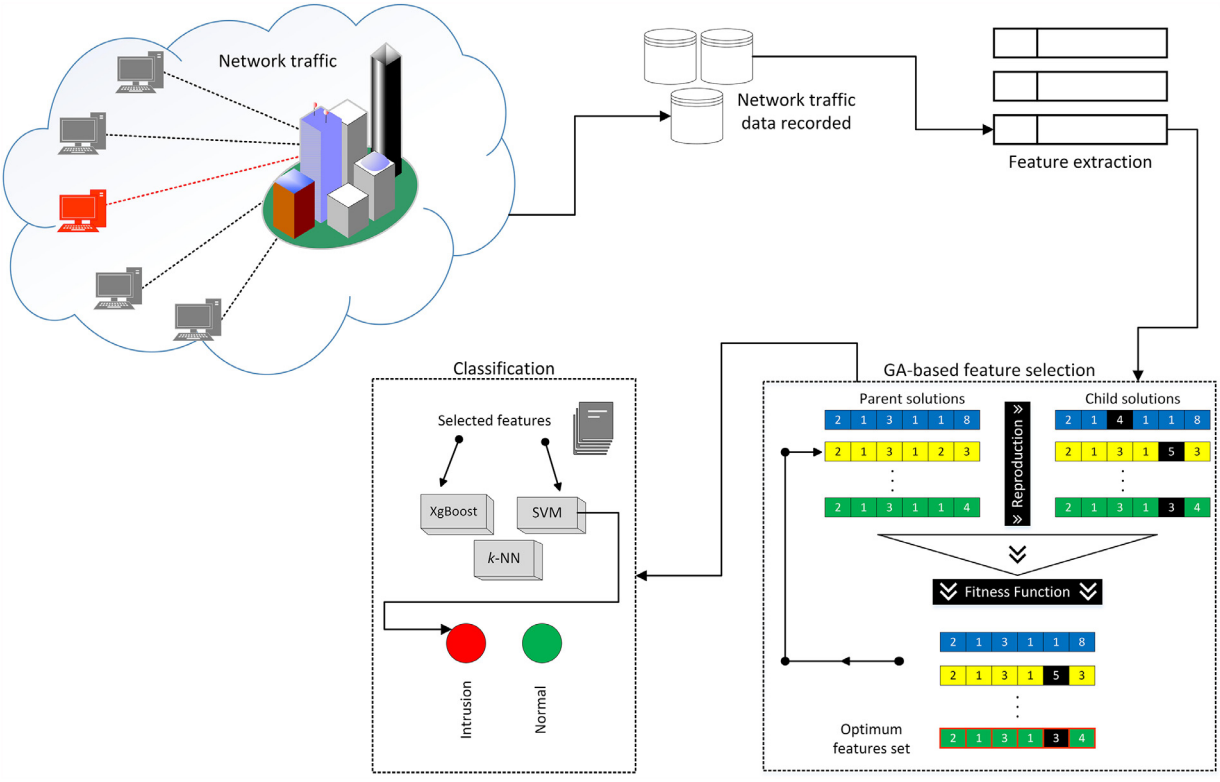

Fig. 1 – Overall working of the proposed solution.

scaled using Eq. (1) and later the data is normalized. Normalization is done to confine all the input values in a desirable range for the processing by the learning module. The formula used for normalization is listed in Eq. (2).

$$x_i' = X_{max} - X_{min} \times \left[\left[\frac{x_i - X_{min}}{X_{max} - X_{min}}\right]\right] + X_{min} \quad (2)$$

To transform alphanumeric values to numeric ones, 1-of-n encoding scheme is used. For evaluation of the proposed technique, simulations are performed on benchmark intrusion detection datasets namely, CIRA-CIC-DOHBrw-2020, UNSW-NB15, and Bot-IoT dataset. These datasets are explained in the results and evaluation section. These three datasets contain textual and numeric values. The categorical features in the datasets are encoded into a format that can be implementable for machine learning purpose using label encoder.

3.2. *GA-based learning module*

The second module of the proposed framework is the GA-based feature selection component, i.e., the main learning module. As a solution to the problem at hand, this work utilizes a GA for optimal feature selection in an unsupervised manner. Here, the GA is tested with different parameters which are listed in the experimental setup. The GA comprises of following steps: (a) initial population generation, (b) fitness function creation, (c) selection strategy for the selection of parents which produces offspring for the next generation, (d) cross-over, (e) mutation, and (f) the final next generation production.

(A) Chromosome structure and initial population

Initial population generation is the first step in the implementation of the GA. Here, the initial population is generated randomly. Chromosomes are created by joining the randomly selected unique genes. According to the selected number of genes in each chromosome, the genes are chosen as a features from the original data attributes randomly without the duplication of genes in a chromosome. Every chromosome in the population represents one solution for selection problem. Fig. 2 shows the chromosome structure. The chromosome is a 1-dimensional array with N cells. The value of N is set to 10 in this work. However, later a number of experiments have been performed by varying the value of N. Each cell of the chromosome holds a numeric value V, where feature set length$\geq V \geq 0$. The numeric value in each gene indicates the feature number. For example, if the gene at index 0 has a value 8, it represents 8th feature from the original data. By that means, each chromosome holds 10 features from the feature set creating a subset that is optimized later to be the optimal solution.

(B) Fitness function

Based on the computation of correlation between the selected features without the presence of class labels, this work presents a novel fitness function. Fitness function provides an instrument to find out those features which have low similarity among them. This allows to collect such features which have higher diversity and are able to represent most of the information related to the dataset. After the correlation of the given features is calculated, the proposed fitness function computes the correlation average.

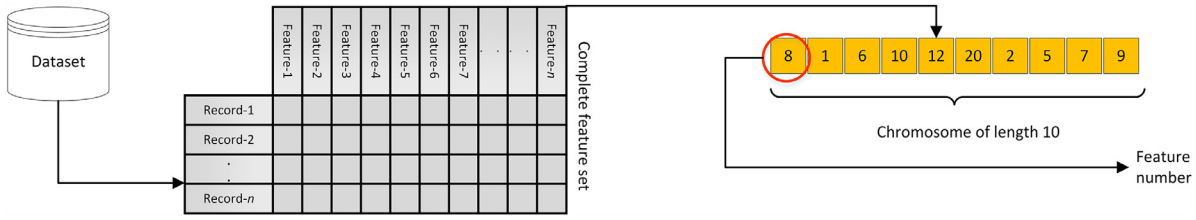

Fig. 2 – Chromosome structure.

After getting the resultant value of the average correlation, optimization is required for the values of both obtained accuracy and average correlation by increasing them throughout the GA generations. As a solution is needed where the selected features must be diverse and have low correlation. For this, transformation is required for the value of average correlation to average non-correlation. This is achieved by taking the difference of average correlation value from 1 (the maximum possible value) in each generation. In this way, the average of the correlation values and accuracy can be represented in increasing/decreasing order. Afterwards, the fitness values can be calculated using the transformed average correlation value and the accuracy of that particular chromosome. Average of the transformed values and the accuracy is taken here. The resultant value is the fitness of the particular chromosome. Main goal of the proposed technique is to maximize the average values of correlation and the accuracy simultaneously. Eq. (5) shows the proposed fitness function.

$$Corr_{avg} = \frac{Sum\ (s)\ of\ values\ above\ the\ diagonal}{Number\ of\ Values} \tag{3}$$

$$Corr^{t}_{avg} = (1\ -\ Corr_{avg}) \tag{4}$$

$$F_i\ =\ \frac{A_i + (1 -\ M_i)}{2} \tag{5}$$

Where, $Coor_{avg}$ is the average correlation value, $Coor^{t}_{avg}$ is the transformed average un-correlation and A_i is the accuracy, and M_i is the computed correlation matrix. The Eq. (4) returns fitness value of i^{th} chromosome.

The fitness function here uses two procedures; objective function and scaling function. Accuracy that is achieved is the objective function which this work optimizes. This is done through the scaling function that is the un-correlation function which scales the performance of the accuracy.

(C) Selection method

For the selection procedure of parents to produce the offspring, the roulette wheel selection strategy is used in this work. The roulette wheel selection strategy has an advantage that when implemented in parallel its execution time decreases and it does not require scaling/sorting like other selection methods. Selection of parents is directly proportional to the fitness values generated through the fitness function. It means, higher the fitness value, higher will be the chances of selection. Linear search is used as the principle in roulette wheel selection where the slots of the wheel are weighted with the fitness values of the individual chromosomes. The chromosome with the higher fitness values cover more area on the roulette wheel thus having higher probability of it being selected.

(D) Crossover and mutation

Reproduction operators has a significant role of diversification in EAs. Mutation and crossover both are used in this work. Once the fitness of all the chromosomes in the population is evaluated, the crossover and mutation operations are performed. Three different crossover rates, i.e., 0.25, 0.5, and 0.75 are tested to select the optimum one. Where, the simulations suggested better performance with 0.5 as crossover probability. For crossover (Fig. 3), the process is designed as follows:

- Two parents are selected using the roulette wheel method for crossover.
- Second half of first parent is directly transferred to child chromosome.
- The remaining genes are copied from the second parent to the child chromosome in the same sequence as in case of the second parent.
- The above two steps are repeated until the number of required individuals in the population is completed.
- Only one child is reproduced using the selected parent pair.

Mutation operation in the GA is used to provide the exploration means where the mutation rate effects the search space breadth. Mutation rate has an effect on the GA convergence. If the mutation rate is very high it will have worse convergence in most cases because this will cause GA to lose valuable portions of solution before convergence. The proposed mutation process is represented in Fig. 3.

For mutation, the process is designed as follows.

- As was the case for crossover, three different mutation rates are tested here, i.e., 0.25, 0.5, and 0.75 to find the optimal one. Based on simulations, 0.5 is selected as the mutation rate.
- Unlike the traditional mutation, the mutation is not applied on each newly created child chromosome. This is the reason for a higher mutation rate. For mutation, a parent is selected with the help of roulette wheel method.
- In mutation, even genes of the parent are directly transferred to the child.
- To increase the diversification, the remaining genes are randomly selected in the range [1-*n*] by avoiding redundancy in the features.

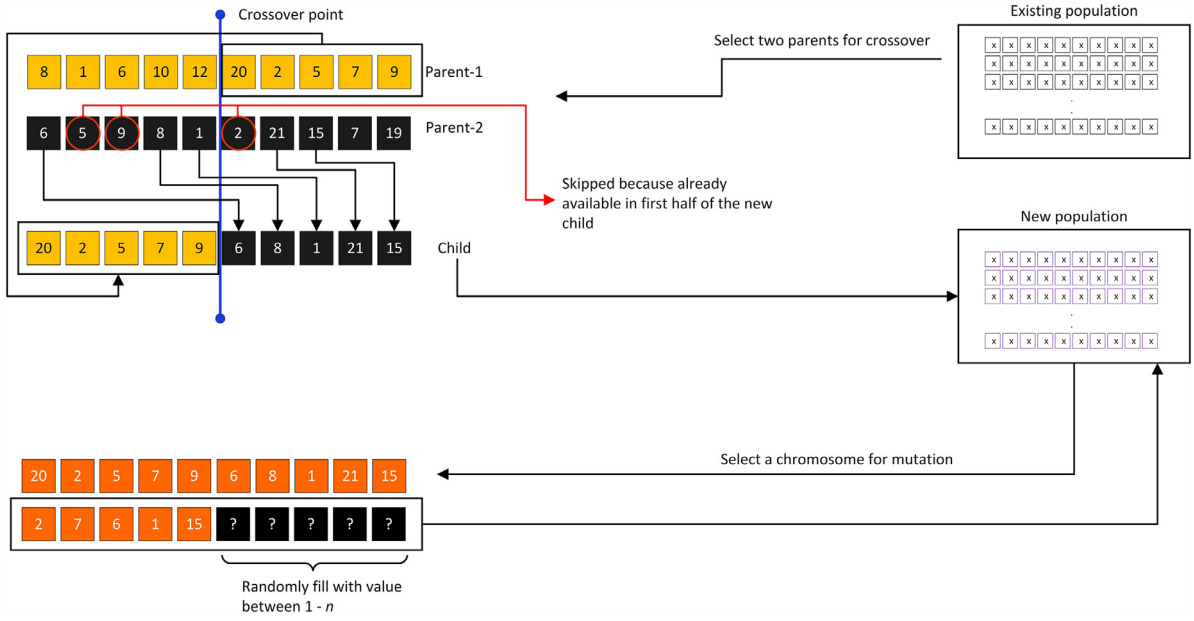

Fig. 3 – Crossover and mutation operations.

(E) Stopping criteria

In the proposed solution, there is a general terminating criteria which when arrives, the total number of generations are terminated. During the GA iterations, the maximum fitness values of the generation are stored. In the first generation, the maximum value V_i is saved as the maximum fitness value V_f. For second generation, during chromosome evaluation, if there exist any value that is greater than V_f, it should replace the current value and the new fitness value is considered as the V_f. If during any n generations the maximum fitness value of that generation is less than the V_f, the GA stopping criteria is met as the fitness value stops increasing. However, during simulations the proposed approach has also been evaluated by executing it for a fixed number of iterations. Once the GA execution is terminated, the best chromosome represents the optimum feature set that is expected to produce better results when used with a standard classifier.

(F) Classification

The final phase of the proposed solution is the classification. When each chromosome is created using the random selection of the genes (features), before classification, the new dataset is generated using only those selected genes. However, once the GA get converged, the features represented by the best chromosome for a particular dataset are considered only. The data is then split into training and testing sets using k-folds cross validation. The classification is performed using the test and train dataset for each of the chromosome. The present work uses three classifier, namely, SVM, k-NN, and XgBoost.

Support Vector Machines (SVMs): The SVM is a supervised learning model used for classification. SVMs are commonly applied to linearly separable data, however, they can also be used for non-linear classification in a high dimensional feature space. It creates a set of hyperplanes (decision planes) in a high dimensional space to classify the data. The advantage of SVM is its effectiveness in high dimensional space, its versatility (different kernel functions can be used and also the custom kernels), and also memory efficiency. However, if the number of features in larger compared to the number of samples, over fitting may occur.

k-Nearest Neighbor (k-NN): The k-NN is a method used for both classification and regression. It classifies the data on the basis of majority votes by the k-neighbors. For a simple case, if $k = 1$, there is one class of data. The optimal value of k can be decided either by inspecting the data or via a series of experiments. A larger value of k is better, as it reduces noise. Votes are decided on the basis of the distance between two points. The distance function can be Euclidean, Manhattan, Minkowski or any other. The distance measure is usually decided on the basis of the type of the data.

XgBoost (eXtreme Gradient Boosting): It is a sophisticated algorithm capable of dealing with different sort of irregularities present in a dataset. XgBoost classification algorithm provide regularization, parallel processing, higher flexibility is term of user defined evaluation criteria, optimization algorithms, and missing value management. The XGBoost provides a parallel tree boosting facility. This enables to solve multiple machine learning problems quickly with better accuracy.

3.3. *Time and space complexity*

There are five key steps in the proposed algorithm. All these are executed for each generation in GA. The time complexity of initial population generation is $O(pop_{size})$. In the fitness function, correlation matrix is computed among the 10 selected features which is equal to the length of chromosome.

Time complexity of fitness function is $(pop_{size} \times chrom^2_{len})$. The time complexity of executing roulette wheel, which is used for parent selection, is $O(pop_{size})$. Time complexity of crossover and mutation is $O(pop_{size})$. Summing this up, the time complexity of the proposed approach becomes $O(g((pop_{size}) + pop_{size} \times chrom^2_{len} + (pop_{size}) + (pop_{size})))$. Overall time complexity of the proposed approach therefore is $O(g(pop_{size} \times chrom^2_{len}))$. Where, g is the number of generations in GA.

Space complexity of the proposed algorithm is dependent on the size of chromosome and population. Size of both chromosome and population are fixed, i.e., chromosome size is 10 and population size is 100. As correlation matrix for each chromosome is also computed to calculated average correlation for that chromosome, and the space complexity of correlation matrix is $O(chrom^2_{len})$. Thus, the overall space complexity of the proposed algorithm is $(pop_{size} \times chrom^2_{len})$.

4. Experiments and results

This section presents the conducted experiments and obtained results. The proposed approach is compared with four state-of-the-art feature selection methods, namely, recursive feature elimination, sequential feature selector, correlation-based feature selection, and select*K*best using three benchmark datasets. Other than this, the present work is also compared with three closely related state-of-the-art methods. The prediction accuracy of classifiers is reported using the features obtained from both the proposed approach and the standard feature selection methods. The proposed solution is coded from scratch using the mathematical scripting language of MATrix LABoratory (MATLAB) where a few built-in libraries are utilized for file reading and evaluation metrics computation. Parts of the code are implemented in Python 3.6 utilizing Spyder as an editor. The machine used for simulations had Intel® Core™ i3 processor, Microsoft's Windows 10 operating system and RAM of 4.00 GB.

4.1. *Datasets*

Experiments are performed using three benchmark intrusion detection datasets implemented and reconfigured on the proposed GA-based framework for feature selection to perform classification. These include: CIRA-CIC-DOHBrw-2020, UNSW-NB15, and Bot-IoT datasets. Table 2 lists a summary of these datasets. They are explained in the following.

CIRA-CIC-DoHBrw-2020: The CIRA-CIC-DoHBrw-2020 dataset captures benign and malicious DoH traffic along with non-DoH traffic. The HTTPS (benign DoH and non-DoH) and DoH traffic is generated to obtain the representative dataset. This is achieved by accessing top 10,000 Alexa websites, and using browsers. Other than this, the DNS tunneling apps are also utilized that support DoH protocol for the browsers. The dataset has a total of 34 features and four classes. The number of samples in this dataset are around 1.4 million. During the data collection phase, too small packets to carry data are ignored for the sake of dimensionality reduction.

Bot-IoT dataset: The Bot-IoT dataset is created at the center of UNSW Canberra Cyber, through a realistic network environment which incorporates the normal as well as the botnet traffic. The dataset has about 72,000,000 records including DDoS, DoS, OS, and service scan, keylogging, and data exfiltration attack classes (Koroniotis et al., 2019).

UNSW NB-15: The UNSW NB-15 dataset contains the raw network packets that are generated through IXIA PerfectStorm tool at Australian Center for Cyber Security (ACCS). The dataset is based on the real normal activities and synthetic attack behaviors. Nine families of attack are included in UNSW NB-15 dataset which are fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode and worms (Moustafa and Slay, 2015). The dataset consists a total of 49 features representing data of different classes.

Keeping the list of presently considered datasets for the experiments (Table 3) in view, the maximum number of features/dimensions evaluated is not greater than 50. This is primarily a constraint due to the utility of existing benchmarks. However, the proposed solution can still classify the data with larger number of dimensions. Generally, increasing the number of features or input variables to a classifier influence the prediction result in an opposing direction. However, too less features also cause stunted learning. In the present work, classifiers are trained with both full features and reduced number of features to compare the performance. The obtained results show that reduced number of features improve the results here. Very large feature set results in overfitting of the classifier, whereas, too few features causes under fitting. Due to this reasoning, different number of features are evaluated here and the relevant optimum outcomes are concluded with 10 features. Although datasets up to 50 features are utilized in

Table 2 – Datasets summary.

	CIRA-CIC-DoHBrw-2020	Bot-IoT	UNSW NB-15
No. of features	34	29	49
No. of classes	4	5	10
No. of Samples	~1.4 million	~3 million	~0.25 million

X

Table 3 – Parameter settings.

Variable	Value(s)
Total number of features	41, 32, 49
Population size	100
Chromosome length	10
Crossover rate	0.5
Mutation rate	0.5
max depth	5
learning rate	0.1
n estimators	20
objective	multi:softmax
booster	gbtree
SVM kernel	Linear
Value of *k*	9

X

this work, however the best results are achieved with 10 features only over all datasets.

4.2. Evaluation metrics

The obtained results are evaluated using two standard evaluation metrics, namely: accuracy and recall. These measures are utilized in assessment of the final classification module.

Accuracy: Accuracy is the measure to evaluate a classifier's performance. It is defined as a ratio of correctly identified observations to the total observations. The higher the accuracy, better are the results. Its value ranges between 0 (worst) and 1 (best). Eq. (6) shows the computation of accuracy. Where, TP (true positives) shows positive instances predicted as positive, FP (false positives) indicates negative instances predicted as positive, FN (false negatives) are the positive instances predicted as negative, and TN (true negatives) represent negative instances predicted as negative.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{6}$$

Recall/ Sensitivity: Sensitivity, Eq. (7), is the ratio of correct positive predictions and the total number of positive samples. The highest value of recall is 1 and the lowest, i.e., worst is 0. At times, it is also called True Positive Rate (TPR) or sensitivity.

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

4.3. Parameter settings

For feature selection, this work utilizes the GA in its second phase with several parameter settings. These parameters include the number of generations, population size, number of genes in each chromosome, fitness function, crossover, and mutation probabilities. Crossover is an important parameter for better convergence of the GA in finding the best solution. In experiments, two different crossover probabilities are used. Uniform crossover for first experiment which produces best results than the second experiment where two point crossover is performed. The crossover rate is set to 0.5, as half of the bits are used to perform crossover. Mutation is the divergence operator of the GA. The ultimate objective of the crossover and mutation is to pull the generations towards the better convergence to find the optimal solution. In experiments, 0.5 is set as mutation rate. However, not every newly created individual is mutated. Table 3 lists the parameter settings used in the simulations. The proposed approach, this point onwards referred to as the GA-based Feature Selection (GbFS), receives the input data, gets executed and returns a subset of optimal features enabling better classification. Later, the standard classifiers are executed with the features selected by the GbFS for performance evaluation. To evaluate the performance of GbFS a number of experiments are performed. These are listed in the following sections.

4.4. Experiments with full features

An experiment has been performed by executing the proposed approach on each of the benchmark datasets using their complete features. As mentioned earlier, the proposed framework, i.e., GbFS has three phases, namely, (a) dataset preparation, (b) GA-based learning module, and (c) classification. For this experiment, the phase-b of the framework has not been utilized so that the classifiers' accuracies while using all dataset features can be obtained. All classification task in this work are performed using a training set and a previously unseen test set.

Table 4 – Attack classes in CIRA-CIC-DoHBrw-2020.

Classification of attack	No. of records	Attack name
DoH	269643	DNS over HTTPs
None-DoH	897493	None DNS over HTTPs
Benign-DoH	19807	Benign DNS over HTTPs
Malicious	249836	Malicious

CIRA-CIC-DOHBrw-2020: With total 34 features of CIRA-CIC-DOHBrw-2020 dataset, intrusion detection has been performed using three classifiers, SVM, *k*-NN and XgBoost. Table 4 presents the total number of records for each particular attack class in the given dataset. Table 5 shows the confusion matrix obtained using the three classifiers, i.e., SVM, *k*-NN, and XgBoost. Fig. 4 shows the results of this experiment. It has been noted from the results that the non-DoH class has 55%–63% detection accuracy using the three classifiers. The detection accuracies for different classes using SVM is higher than others. Fig. 4 also shows the results obtained for the XGBoost of chosen classes from the CIRA-CIC-DOHBrw-2020 data. As compared to the *k*-NN and SVM, XGBoost has an average classification accuracy of 61%.

Bot-IoT dataset: Bot-IoT dataset has total of 29 features, this work has calculated the detection accuracies of each class mentioned in Table 6. Table 7 shows the confusion matrix utilizing the three classifiers, i.e., SVM, *k*-NN, and XgBoost for the Bot-IoT data. Fig. 5 shows the results of this experiment. It is observed that among the five classes, SVM classifier achieves best accuracy for Theft class while the SVM classifier struggles in detecting DDOS class. The Bot-IoT dataset has been used for *k*-NN classifier and the reported results are presented in Fig. 5. For *k*-NN classifier, two classes, i.e., reconnaissance and theft show a significant detection accuracy. While looking at remaining classes, DDOS and normal class achieves 87% and 83% of detection accuracy, respectively. The XGBoost classifier is used for the intrusion detection using Bot-IoT dataset. Results of the experiment are demonstrated in the figure where the TPs and the accuracies of each class of Bot-IoT dataset is mentioned. XGBoost provides quite better results in detecting available classes as compared to the other classifier. It has been observed that the theft class has the highest detecting class with an accuracy of 98.8% while normal class has the least accuracy of 72.4%.

UNSW NB-15 dataset: Table 8 lists the classes of attacks contained in the UNSW NB-15 dataset. Fig. 6 shows the results of this experiment. It can been observed that among ten classes of UNSW NB-15 dataset, only generic attack class has an accuracy of 98.1% using SVM. On the other hand, it has been

Table 5 – Confusion matrix obtained for the three classifiers over CIRA-CIC-DOHBrw-2020 dataset.

		DoH			non-DoH			Benign-DoH			Malicious		
		SVM	*k*-NN	XgBoost	SVM	*k*-NN	XgBoost	SVM	*k*-NN	XgBoost	SVM	*k*-NN	XgBoost
DoH	SVM	157571			21247			26386			64439		
	k-NN		149632			100239			4568			15204	
	XgBoost			136719			110894			3698			18332
non-DoH	SVM	147852			553496			100478			95667		
	k-NN		168243			493654			120010			115586	
	XgBoost			11007			569173			147852			169461
Benign-DoH	SVM	1247			3971			11321			3268		
	k-NN		1478			6874			10253			1202	
	XgBoost			3550			123			11147			4987
Malicious	SVM	12047			15478			12598			209713		
	k-NN		37999			24245			17014			170578	
	XgBoost			13922			14471			35789			185654

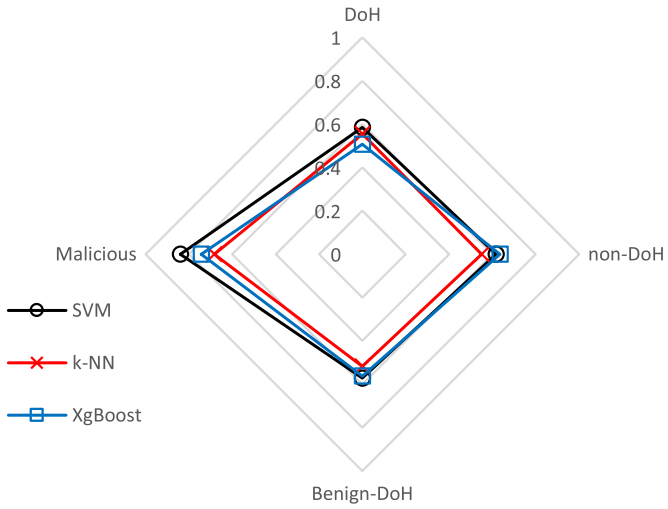

Fig. 4 – Results obtained using CIRA-CIC-DoHBrw-2020 dataset.

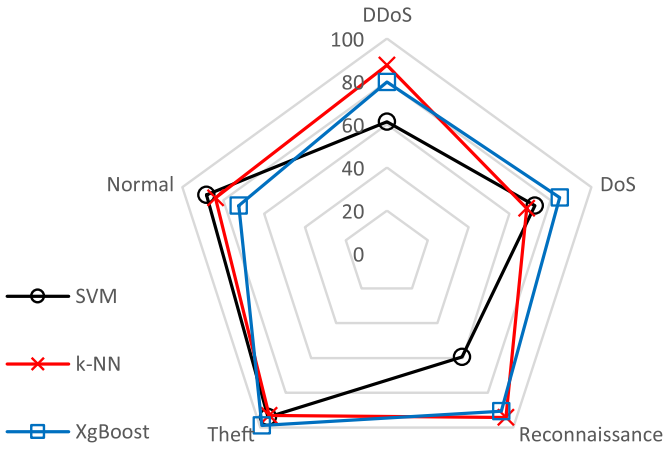

Fig. 5 – Results obtained using Bot-IoT dataset.

noted that two classes have near to 0% detection accuracies. Experiment on UNSW NB-15 dataset have been performed with the k-NN classifier having 9 neighbors. As is the case of SVM, the generic class has the highest accuracy of 98.3% using k-NN. Using XGBoost classifier, it can be seen that the generic class has highest detection accuracy of 96.8%.

4.5. *Experiments with feature selection using GbFS*

Another set of experiments has been performed by executing the proposed approach on each of the benchmark datasets. However, unlike the previous experiment, here all phases of the proposed approach are utilized. These include, dataset preparation, GA-based learning module, and classification. This enables to select a set of optimum features form the data to perform classification instead of utilizing all attributes of the data. For this experiment, among full features of the three benchmark datasets, three different experiments are performed for each dataset on the basis of different number of features selected by the GbFS's learning module. The purpose of having three different number of features is to identify the optimal number of features that can be extracted to rep-

Table 9 – Performance of the four competing methods

Datasets	(Zhao et al., 2021)	(Kannari et al., 2021)	GbFS (proposed)	(Tama et al., 2019)
CIRA-CIC-DOHBrw-2020	93.33	97.80	98.94	94.55
Bot-IoT	94.25	98.10	98.90	92.66
UNSW NB-15	95.00	96.40	96.48	91.27
Average	94.00	97.43	98.11	92.83

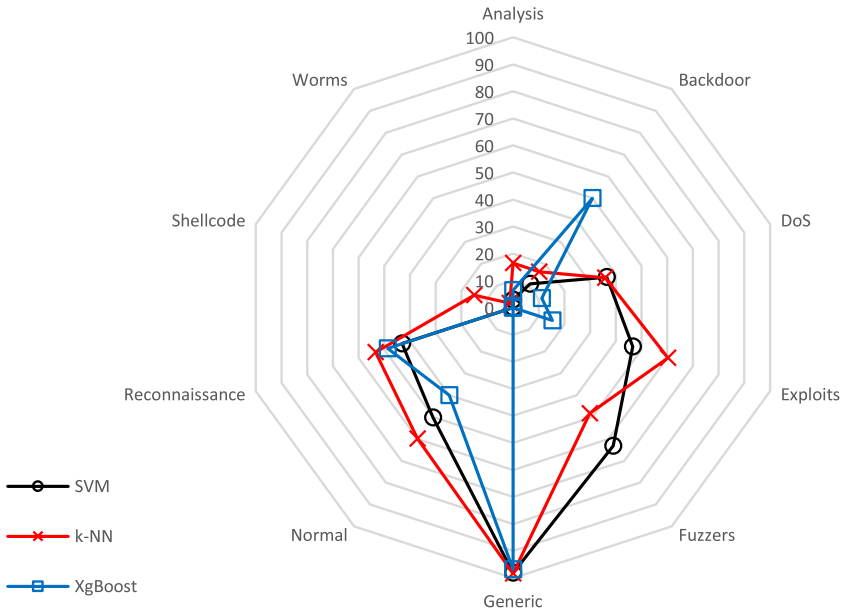

Fig. 6 – Results obtained using UNSW NB-15 dataset.

Table 6 – Attack classes in Bot-IoT dataset.

Classification of attack	No. of records	Attack name
DDoS	240,000	DDos
DoS	242,788	DoS
Reconnaissance	182,166	OS and Service Scan
Theft	160	Keylogging and Data Exfiltration

resent the most of the information in the dataset to enhance classifiers' learning. For this, separate experiments are performed by selecting 8, 10, and 12 features form the datasets. The decision of selecting 8, 10 and 12 features among the chosen datasets is derived from the work done in the field of feature selection from the literature (Bolon-Canedo et al., 2011). Either these number of features are not tested before or the range of the number of features lies between the minimum and maximum number of features selected by the past works (Bolon-Canedo et al., 2011; Alazzam et al., 2020). Therefore, from the CIRA-CIC-DoHBrw-2020 dataset (having 34 features), Bot-IoT (with 29 features) and UNSW NB-15 dataset (containing 47 features) this experiment selects 8, 10, and 12 features, respectively. Fig. 7 shows the convergence graph of the proposed approach. The figure illustrates two values that are the best fitness values and the average fitness values achieved in every generation. Through a graphical representation, it can be observed as with an increment in the number of generations, solutions to the problem evolves and produces better overall average fitness values. The proposed model can be categorized as greedy GA that is why it can be noticed that there is a continuous increase in the fitness values. For the CIRA-CIC-DoHBrw-2020 data, after 130th generation the best fitness values stops converging and hence met the stopping criteria that terminates the GA process and gives the feature lists with selected features as the optimal solutions. For the UNSW data, after 210th generation the best fitness values stops evolving which results in termination of GA as the stopping criteria met.

4.6. *Classifiers results*

Once the features are selected by the proposed approaches' learning module, an experiment has been performed to compute the classifier's' accuracy. For this, three classifiers, namely, SVM, *k*-NN, and XgBoost are used and are executed on the three benchmark datasets with full feature set and the features extracted using the GbFS method.

CIRA-CIC-DOHBrw-2020: Fig. 8 (a) shows the results of this experiment when executed on the CIRA-CIC-DOHBrw-2020 dataset. It can be observed that the classification accuracy using all features is less than the selected features using the proposed method in all three cases (i.e., the three classifiers). Among the three classifiers, SVM performs better when executed on all features and also the features selected using the proposed method. However, the performance difference between SVM and XgBoost is marginal for the features selected using GbFS, i.e., the difference is just 0.78 in terms of accuracy. Whereas, in terms of recall as a performance evaluation metric, XgBoost performs better than SVM.

Bot-IoT: Fig. 8 (b) shows the results of this experiment when executed on the Bot-IoT dataset. In this case, the proposed approach performs better by providing the classifiers with appropriate features of the data to achieve higher accuracy and recall values. For the Bot-IoT data, the XgBoost classifier performs better than others by achieving the accuracy of 99.7%. However, in terms of recall, the SVM classifier beats others with a recall value of 99.73%.

UNSB NB-15: Fig. 8 (c) shows the results of this experiment when executed on the UNSB NB-15 dataset. As is the case of

Table 7 – Confusion matrix obtained for the three classifiers over Bot-IoT dataset.

		DDoS			DoS			Reconnaissance			Theft			Normal		
		SVM	k-NN	XgBoost	SVM	k-NN	XgBoost	SVM	k-NN	XgBoost	SVM	k-NN	XgBoost	SVM	k-NN	XgBoost
DDoS	SVM	147120			20012			37181			489			35198		
	k-NN		210240			5079			19784			978			3919	
	XgBoost			147120			20012			37181			489			35198
DoS	SVM	13371			175778			4194			35020			14425		
	k-NN		4568			166066			38798			7814			25542	
	XgBoost			133			175778			419			9			0
Reconnaissance	SVM	23089			21589			108206			7693			21589		
	k-NN		4789			1558			171600			679			3540	
	XgBoost			1108			4186			102206			2894			0
Theft	SVM	2			4			3			150			1		
	k-NN		3			6			0			148			3	
	XgBoost			489			117			69			150			0
Normal	SVM	28			36			35			12			842		
	k-NN		38			91			2			1			703	
	XgBoost			28			36			35			12			842

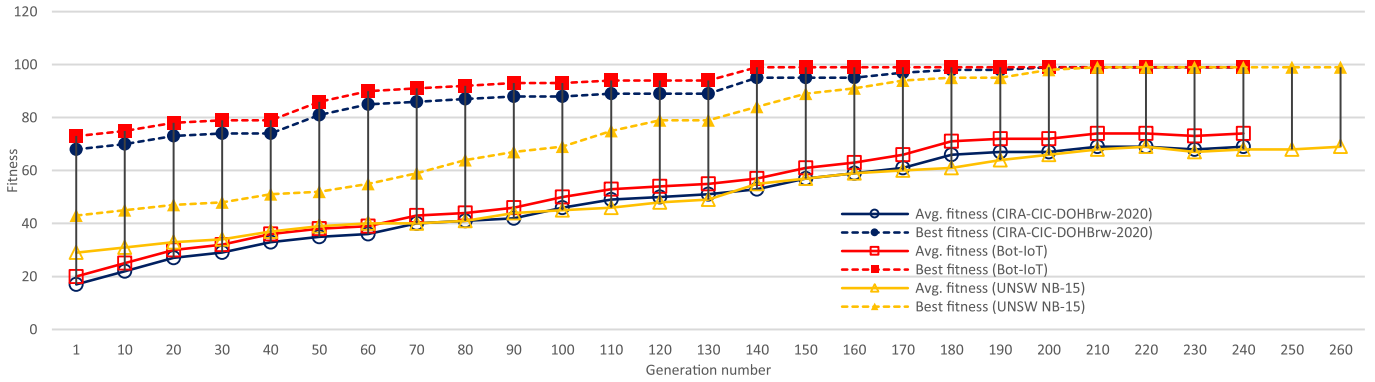

Fig. 7 – Convergence graph of the proposed approach.

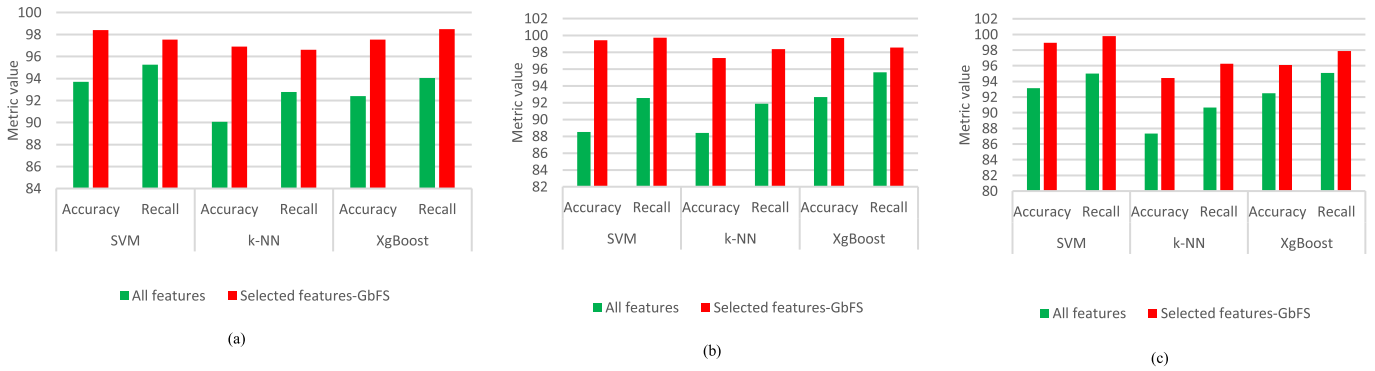

Fig. 8 – Classification results

Table 8 – Per class records in UNSW NB-15 dataset.

Classification of attack	No of records
Analysis	677
Backdoor	577
DoS	4089
Exploits	7061
Fuzzers	12,062
Generic	5016
Normal	31,395
Reconnaissance	1695
Shellcode	378
Worms	44

other two datasets, for the UNSW NB-15 data, performance of the classifiers is enhanced when they use features extracted by the proposed method. The average accuracy of all three classifiers when they use the complete feature set is 90.98%. Whereas, there is an increase of 5.5% in the accuracy when the same classifiers utilize the features extracted by the proposed approach.

4.7. *Comparison with the standard feature selectin methods*

The proposed approach is a novel GA-based feature selection method targeted towards the intrusion detection systems. It is important to see the performance of the proposed approach against the standard feature selection methods. Therefore, an experiment has been conducted here to see the performance difference between the proposed method and four existing feature selection techniques, namely, Recursive Feature Elimination (RFE), Sequential Feature Selector (SFS), Correlation-based Feature Selection (CFS), and select*K*best. Different feature selection methods produces varying results when executed on the same data. Therefore, the four existing feature selection methods are first executed on each of the datasets in isolation and the final set of features is opted for using the majority vote. Afterwards, the three classifiers are executed using the feature set based on majority vote of the standard feature selection methods. The results of this experiment are mentioned in Fig. 9. These results are an average of the three classifiers. It can be seen from the results that the proposed approach, i.e., GbFS performs better than the past methods in majority of the cases having and average accuracy of 98.10%. Whereas, the average accuracy of the past methods is 97.11%.

4.8. *Comparison with existing solutions*

Other than the comparison of the proposed solution with existing classifiers, an experiment is performed to compare the proposed solution, i.e., GbFS with three closely related state-of-the-art methods for the same task. These include, representativeness-based instance selection for intrusion detection (Zhao et al., 2021), network intrusion detection using sparse autoencoder with Kannari et al. (2021), and two-stage classifier Ensemble (Tama et al., 2019). The work in Zhao et al. (2021) develop two representative-based algorithms named as Representative-based Instance Selection for Balanced Data

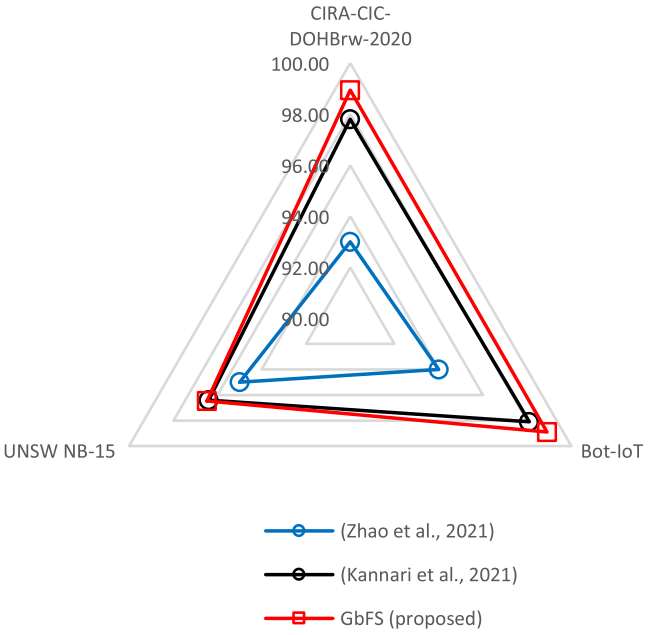

Fig. 9 – Comparison with the standard feature selection methods.

(RBIS) and Representative-Based Instance Selection for Imbalanced data (RBIS-IM). RBIS selects the same portion of representative instances of each class and then the classification is performed using 1-NN, SVM, and Adaboost. RBIS-IM selects important representative majority instances according to the number of instances in the minority class. The proposal of Kannari et al. (2021) use labels and one hot encoder to convert the text data into numeric format for the classification purpose. Two feature selection techniques are utilized to extract active features from the feature set. Second percentile method and recursive feature selection are used to select relevant features from the dataset to reduce the computational time and complexity of the model. Sparse autoencoder with swish-PReLU activation function is adopted for classification. Higher accuracy is obtained through sparse autoencoder. However, feature selection and classification is applied to a single class at a time. The work in Tama et al. (2019) present a hybrid feature selection approach. Their method is a fusion of three optimization techniques, namely, PAS, ACO, and GA. These are utilized to reduce the feature size of the training data to achieve better classification accuracy. Table 9 lists the results of this experiment. Where, the results suggest better performance of GbFS in comparison to the three state-of-the-art methods. However, the method of Kannari et al. (2021) performs close to GbFS primarily due to the utility of deep learning approach.

5. Conclusions

In any enterprise network environment, an Intrusion Detection System (IDS) plays a critical role as a security measures. IDS monitors the incoming traffic of the network and analyzes the data packets to classify the connection being normal or

malicious. This seems straightforward; however, the IDS must be continuously fine-tuned to attain high detection rate. As in the modern world, the traffic and the hackers are evolving which creates new challenges for intrusion detection systems due to the presence of big data. To gain maximum performance from an intrusion detection system, the curse of dimensionality in the massive datasets needs to be addressed. This work presented a novel Genetic Algorithm (GA)-based approach for the feature selection to enhance the detection accuracy of the IDS. This proposed approach selected the most appropriate features from the IDS datasets. Experiments were performed by selected 8–10 features from the data. This work also presented a novel objective function for the GA that assigned the fitness values to the individuals in the GA population enabling to select the chromosomes that represented the optimum feature set. The proposal was evaluated using three benchmark datasets and a comparison was performed with four standard feature selection methods, namely, recursive feature elimination, sequential feature selector, correlation-based feature selection, and select*K*best. The solution was also compared with three closely related state-of-the-art methods. The obtained results suggested better performance of the proposed approach in majority of the cases.

The present work has multiple future prospects. The GA is used at the core of present solution which is fundamentally slow due to multiple repetitive iterations and the reproduction operations. To address this concern, (1 + 1)-Evolutionary Strategy (ES) can be adopted in the future to attain quicker results. Binary chaotic genetic algorithms have recently shown promising results. In the future, such optimization techniques can be utilized for the same task. The proposed framework used three classifies in the prediction phase, which is a supervised learning approach. Another future direction can be to utilize an unsupervised learning method, like clustering to make the machine self-learn new kinds of attacks.

Compliance with ethical standards

Funding

This work was sponsored by the GIK Institute graduate research fund under GA-PSS scheme. Grant number GCS1NY5

Ethical approval

All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Informed consent

Informed consent was obtained from all individual participants included in the study.

Declaration of Competing Interest

There are no competing interests to declare.

REFERENCES

Afzaliseresht N, Miao Y, Michalska S, Liu Q, Wang H. From logs to stories: human-centred data mining for cyber threat intelligence. IEEE Access 2020;8:19089–99.

Ahmad I, Abdullah A, Alghamdi A, Alnfajan K, Hussain M. Intrusion detection using feature subset selection based on MLP. Sci. Res. Essays 2011;6(34):6804–10.

Alazzam H, Sharieh A, Sabri KE. A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. Expert Syst. Appl. 2020;148.

Alloghani M, Al-Jumeily D, Hussain A, Mustafina J, Baker T, Aljaaf AJ. Implementation of machine learning and data mining to improve cybersecurity and limit vulnerabilities to cyber attacks. In: Nature-Inspired Computation in Data Mining and Machine Learning; 2020. p. 47–76.

Aslahi-Shahri BM, Rahmani R, Chizari M, Maralani A, Eslami M, Golkar MJ, Ebrahimi A. A hybrid method consisting of GA and SVM for intrusion detection system. Neural Comput. Appl. 2016;27(6):1669–76.

Amini F, Hu G. A two-layer feature selection method using genetic algorithm and elastic net. Expert Syst. Appl. 2021;166.

Bolon-Canedo V, Sanchez-Marono N, Alonso-Betanzos A. Feature selection and classification in multiple class datasets: an application to KDD Cup 99 dataset. Expert Syst. Appl. 2011;38(5):5947–57.

Carlin A, Hammoudeh M, Aldabbas O. Intrusion detection and countermeasure of virtual cloud systems-state of the art and current challenges. Int. J. Adv. Comput. Sci. Appl. 2015;6(6).

Conti M, Dargahi T, Dehghantanha A. Cyber threat intelligence: challenges and opportunities. In: Cyber Threat Intelligence. Cham: Springer; 2018. p. 1–6.

Das AK, Das S, Ghosh A. Ensemble feature selection using bi-objective genetic algorithm. Knowl. Based Syst. 2017;123:116–27 2018.

Dwivedi S, Vardhan M, Tripathi S. Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection. Clust. Comput. 2021:1–20. doi:10.1007/s10586-020-03229-5.

Elingiusti M, Aniello L, Querzoni L, Baldoni R. PDF-malware detection: a survey and taxonomy of current techniques. Cyber Threat Intelligence 2018. doi:10.1007/978-3-319-73951-9_9.

Gamal M, Abbas H, Sadek R. Hybrid approach for improving intrusion detection based on deep learning and machine learning techniques. In: Proceedings of the Joint European-US Workshop on Applications of Invariance in Computer Vision; 2020. p. 225–36.

Gharaee H, Hosseinvand H. A new feature selection IDS based on genetic algorithm and SVM. In: Proceedings of the 8th International Symposium on Telecommunications (IST); 2016. p. 139–44.

Giraldo J, Sarkar E, Cardenas AA, Maniatakos M, Kantarcioglu M. Security and privacy in cyber-physical systems: a survey of surveys. IEEE Des. Test 2017;34(4):7–17.

Guo W, Wu C, Ding Z, Zhou Q. Prediction of surface roughness based on a hybrid feature selection method and long short-term memory network in grinding. Int. J. Adv. Manuf. Technol. 2021;112(9):2853–71.

Halim Z, Ali O, Khan G. On the efficient representation of datasets as graphs to mine maximal frequent itemsets. IEEE Trans. Knowl. Data Eng. 2021;4(33):1674–91.

Halim Z, Waqar M, Tahir M. A machine learning-based investigation utilizing the in-text features for the identification of dominant emotion in an email. Knowl. Based Syst. 2020;208.

Halim Z, Rehan M. On identification of driving-induced stress using electroencephalogram signals: a framework based on wearable safety-critical scheme and machine learning. Inf. Fusion 2020;53:66–79.
Haq NF, Onik AR, Hridoy MAK, Rafni M, Shah FM, Farid DM. Application of machine learning approaches in intrusion detection system: a survey. IJARAI Int. J. Adv. Res. Artif. Intell. 2015;4(3):9–18.
Ho TC. Network-Based Anomaly Intrusion Detection using Ant Colony Clustering Model and Genetic-Fuzzy Rule Mining Approach PhD Thesis. City University of Hong Kong; 2006.
Kannari PR, Shariff NC, Biradar RL. Network intrusion detection using sparse autoencoder with swish-PReLU activation Model. J. Ambient Intell. Hum. Comput. 2021:1–3. doi:10.1007/s12652-021-03077-0.
Koroniotis N, Moustafa N, Sitnikova E, Turnbull B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. Future Gener. Comput. Syst. 2019;100:779–96.
Kuang F, Xu W, Zhang S. A novel hybrid KPCA and SVM with GA model for intrusion detection. Appl. Soft Comput. 2014;18:178–84.
Li X, Yi P, Wei W, Jiang Y, Tian L. LNNLS-KH: A feature selection method for network intrusion detection. Secur. Commun. Netw. 2021;2021. doi:10.1155/2021/8830431.
Liu H, Motoda H, editors. Computational Methods of Feature Selection. CRC Press; 2007.
Liu X, Tang J. Mass classification in mammograms using selected geometry and texture features, and a new SVM-based feature selection method. IEEE Syst. J. 2013;8(3):910–20.
Mahindru A, Sangal AL. FSDroid:-a feature selection technique to detect malware from android using machine learning techniques. Multimed. Tools Appl. 2021;14:1–53.
Maleki N, Zeinali Y, Niaki ST. A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection. Expert Syst. Appl. 2021;164.
Moustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: Proceedings of the Military Communications and Information Systems Conference (MilCIS); 2015. p. 1–6.
Nouri-Moghaddam B, Ghazanfari M, Fathian M. A novel multi-objective forest optimization algorithm for wrapper feature selection. Expert Syst. Appl. 2021;175.
Protić DD. Review of KDD Cup'99, NSL-KDD and Kyoto 2006+ datasets. Vojnoteh. Glas. 2018;66(3):580–96.
Riesco R, Larriva-Novo X, Villagrá VA. Cybersecurity threat intelligence knowledge exchange based on blockchain. Telecommun. Syst. 2020;73(2):259–88.
Shalaginov A, Banin S, Dehghantanha A, Franke K. Machine learning aided static malware analysis: A survey and tutorial. In: Cyber Threat Intelligence; 2018. p. 7–45. doi:10.1007/978-3-319-73951-9_2.
Sindhu SSS, Geetha S, Kannan A. Decision tree based light weight intrusion detection using a wrapper approach. Expert Syst. Appl. 2012;39(1):129–41.
Stampar M, Fertalj K. Artificial intelligence in network intrusion detection. In: Proceedings of the 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO); 2015. p. 1318–23.
Stein G, Chen B, Wu AS, Hua KA. Decision tree classifier for network intrusion detection with GA-based feature selection, 2; 2005. p. 136–41.
Sumaiya Thaseen I, Saira Banu J, Lavanya K, Rukunuddin Ghalib M, Abhishek K. An integrated intrusion detection system using correlation-based attribute selection and artificial neural network. Trans. Emerg. Telecommun. Technol. 2021;32(2):e4014.
Tahir M, Tubaishat A, Al-Obeidat F, Shah B, Halim Z, Waqas M. A novel binary chaotic genetic algorithm for feature selection and its utility in affective computing and healthcare. Neural Comput. Appl. 2021;18:1–22.
Tama BA, Comuzzi M, Rhee KH. TSE-IDS: a two-stage classifier ensemble for intelligent anomaly-based intrusion detection system. IEEE Access 2019;7:94497–507.
Tausif M, Ferzund J, Jabbar S, Shahzadi R. Towards designing efficient lightweight ciphers for internet of things. KSII Trans. Internet Inf. Syst. 2017;11(8).
Tribak H, Delgado-Marquez BL, Rojas P, Valenzuela O, Pomares H, Rojas I. Statistical analysis of different artificial intelligent techniques applied to intrusion detection system. In: Proceedings of the 2012 International Conference on Multimedia Computing and Systems; 2012. p. 434–40.
Tu S, Waqas M, Rehman SU, Mir T, Abbas G, Abbas ZH, Halim Z, Ahmad I. Reinforcement learning assisted impersonation attack detection in device-to-device communications. IEEE Trans. Veh. Technol. 2021;70(2):1474–9.
Uzma Al-Obeidat F, Tubaishat A, Shah B, Halim Z. Gene encoder: A feature selection technique through unsupervised deep learning-based clustering for large gene expression data. Neural Comput. Appl. 2021:1–23. doi:10.1007/s00521-020-05101-4.
Viharos ZJ, Kis KB, Fodor Á, Büki ÁM. Adaptive, hybrid feature selection (AHFS). Pattern Recognit. 2021;11.
Von Solms R, Van Niekerk J. From information security to cyber security. Comput. Secur. 2013;38:97–102.
Wan J, Waqas M, Tu S, Hussain SM, Shah A, Rehman SU, Hanif M. An efficient impersonation attack detection method in fog computing. CMC Comput. Mater. Contin. 2021;68(1):267–81.
Wang J, Liu C, Zhou M. Improved bacterial foraging algorithm for cell formation and product scheduling considering learning and forgetting factors in cellular manufacturing systems. IEEE Syst. J. 2020;14(2):3047–56.
Xue Y, Tang Y, Xu X, Liang J, Neri F. Multi-objective feature selection with missing data in classification. IEEE Trans. Emerg. Top. Comput. Intell. 2021. doi:10.1109/TETCI.2021.3074147.
Xue Y, Xue B, Zhang M. Self-adaptive particle swarm optimization for large-scale feature selection in classification. ACM Trans. Knowl. Discov. Data 2019;13(5):1–27.
Xue B, Zhang M, Browne WN, Yao X. A survey on evolutionary computation approaches to feature selection. IEEE Trans. Evol. Comput. 2016;20(4):606–26.
Xu H, Fu Y, Fang C, Cao Q, Su J, Wei S. An improved binary whale optimization algorithm for feature selection of network intrusion detection. In: Proceedings of the IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems; 2018. p. 10–15.
Yousefi-Azar M, Varadharajan V, Hamey L, Tupakula U. Autoencoder-based feature learning for cyber security applications. In: 2017 International joint conference on neural networks; 2017. p. 3854–61. doi:10.1109/IJCNN.2017.7966342.
Zhao F, Xin Y, Zhang K, Niu X. Representativeness-based instance selection for intrusion detection. Secur. Commun. Netw. 2021:1–13. doi:10.1155/2021/6638134.

Zahid Halim received the B.S. degree in computer science from the University of Peshawar, Pakistan, in 2004, M.S. degree in computer scie'nce from the National University of Computer and Emerging Sciences, Pakistan, in 2007, and also the Ph.D. degree in computer science from the National University of Computer and Emerging Sciences, Pakistan, in 2010. He was with the National University of Computer and Emerging Sciences, Islamabad, Pakistan, as a Faculty Member from 2007 to 2010. Currently he is an Associate Professor with GIKI, Pakistan. His current research interests include machine learning and data mining with an emphasis on proba-

bilistic/uncertain data mining. Dr. Halim is a member of the IEEE Computational Intelligence Society.

Muhammad Nadeem Yousaf received his BS degree in Computer System Engineering Science from University of Engineering and Technology, Pakistan, in 2016, an M.S. degree in Computer Systems Engineering from Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences and Technology, Pakistan, in 2019. He worked presently working as a Data Scientist at VisionX. He is also a member of the Machine Intelligence Research Group (MInG).

Muhammad Waqas received the B.Sc. and M.Sc. degrees from the Department of Electrical Engineering, University of Engineering and Technology, Peshawar, Pakistan, in 2009 and 2014, respectively. He received his Ph.D. degree from the Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing, China. He is currently working as an Assistant Professor at GIKI, Pakistan. He has several research publications in IEEE journals and conferences. His current research interests are in the areas of networking and communications, including 5G networks, D2D communication resource allocation and physical layer security and information security, mobility investigation in D2D communication, Fog computing, and MEC.

Muhammad Sulaiman received his BS degree in Computer Science from COMSATS University, Pakistan, in 2016, an M.S. degree in Computer Systems Engineering from Ghulam Ishaq Khan (GIK) Institute of Engineering Sciences and Technology, Pakistan, in 2019. He worked as a Lecturer at Capital University of Science and Technology, Islamabad, Pakistan from 2019 till now. He is also a member of the Machine Intelligence Research Group (Ming). Sulaiman's research interest includes resource scheduling for parallel and distributed systems, big data analysis, machine learning, and evolutionary algorithms.

Ghulam Abbas received the B.S. degree in computer science from University of Peshawar, Peshawar, Pakistan, in 2003, and the M.S. degree in distributed systems and the Ph.D. degree in computer networks from the University of Liverpool, Liverpool, U.K., in 2005 and 2010, respectively. From 2006 to 2010, he was a Research Associate with Liverpool Hope University, Liverpool, where he was associated with the Intelligent and Distributed Systems Laboratory. Since 2011, he has been an Assistant Professor with the Faculty of Computer Sciences & Engineering, GIK Institute of Engineering Sciences and Technology, Pakistan. His research interests include Internet architecture, congestion control and routing. He is a Fellow of the Institute of Science & Technology, U.K., and a member of the IEEE Computer and Communications Societies.

Masroor Hussain received his B.S. degree in computer science in 2001 and M.S. degree in computer science in 2003 from National University of Computer and Emerging Science, Lahore, Pakistan. He received his Ph.D. degree in high performance computing from GIK Institute of Engineering Sciences and Technology, Topi, Pakistan in 2011. Since 2011, Dr. Hussain has been with the Faculty of Computer Sciences and Engineering, GIK Institute of Engineering Sciences and Technology, Topi, Pakistan. His research interests includes parallel and distributed systems, mesh reordering and partitioning techniques and spatial temporal neural networks.

Iftekhar Ahmad received the Ph.D. degree in communication networks from Monash University, Clayton, VIC, Australia, in 2007.

He is currently an Associate Professor with the School of Engineering, Edith Cowan University, Joondalup, WA, Australia. His current research interests include 5G technologies, green communications, quality of service in communication networks, software-defined radio, wireless sensor networks, and computational intelligence

Muhammad Hanif obtained his B.Sc degree (March 2002 - Jan 2006) in Computer Engineering from Electrical and Computer Engineering Department, COMSATS Institute of Information Technology, Abbottabad, Pakistan. He completed his M.Sc degree (Aug 2007 - Aug 2009) in Information Technology with specialization in Signal Processing from Department of Signal Processing, Faculty of Engineering Sciences at Tampere University of Technology, Tampere, Finland. He pursued his PhD degree (Jun. 2011 - Jun 2015) in Image Processing from College of Engineering and Computer Science, Australian National University, Canberra, Australia. Hanif received the European Research Consortium for Informatics and Mathematics (ERCIM) fellowship (Jan 2017 – Nov 2019) for his postdoc at Italian National Research Council (CNR), Pisa, Italy. He mainly work on sparse signal representation based approaches to address ancient manuscripts restoration, retrieval and classification.