A Survey of Network Intrusion Detection Using Machine Learning Techniques

Thomas Rincy N and Roopam Gupta

Abstract Nowadays, a huge amount of information flows daily on public and private computer networks. Since sensitive information has a high probability of being transmitted, there is an important need to protect networks from intrusions. Hence, adopting an intrusion detection system is imperative. As the frequency of sophisticated attacks has been increasing tremendously over the past years, machine learning approaches were introduced to identify intrusion patterns and prevent sophisticated attacks. This survey provides an up-to-date review of leading-edge techniques used by intrusion detection systems that rely on machine learning techniques. Moreover, it introduces important key machine learning concepts such as ensemble learning and feature selection that are applied to protect networks from unauthorized access and make networks and computers safer. The article then reviews signature, anomaly, and hybrid intrusion detection systems that apply machine learning techniques. It is observed that hybrid network intrusion detection system may be the most effective. Then, the article examines the characteristics of popular benchmark datasets for evaluating intrusion detection systems such as NSL-KDD, Kyoto 2006 +, and KDD Cup-‘99 and performance metrics to appraise intrusion detection results. Finally, the article discusses research opportunities in the field of intrusion detection.

Keywords Intrusion detection system · Machine learning · Ensemble learning · Feature selection · Datasets · Performance metrics

The original version of this chapter was revised: The original version of this chapter was revised by including a new sections. The correction to this chapter is available at https://doi.org/10.1007/978-3-030-66288-2_13

Thomas Rincy N (✉)
Department of Computer Science and Engineering, University Institute of Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, MP, India
e-mail: rinc_thomas@rediffmail.com

Roopam Gupta
Department of Information Technology, University Institute of Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, MP, India
e-mail: roopamgupta@rgtu.net

H. Chiroma et al. (eds.), *Machine Learning and Data Mining for Emerging Trend in Cyber Dynamics*, https://doi.org/10.1007/978-3-030-66288-2_4

1 Introduction

In recent decades, the internet has greatly facilitated communication between individuals and businesses. Many services are now offered and a lot of important and confidential information is transmitted over the internet and private computer networks. In this context, an important issue is to protect the privacy and integrity of computer networks from intruders and to ensure that networks can continue to operate normally when they are under attack. A recent report by Cybersecurity Ventures estimated that by 2021, cybercrime damage will rise to $6 trillion annually from $3 trillion in 2015 [1]. Moreover, it is predicted that there will be 6 billion internet users worldwide by 2022, and global spending on cyber security will cumulatively exceed $1 trillion over the next five years.

Due to the increasing frequency of sophisticated cyber-attacks, network security has become an emerging area of research for computer networking. To prevent intrusions and protect computers, programs, data, and unauthorized access to systems, intrusion detection systems (IDS) have been designed. An IDS can identify an external or internal intrusion in an organization's computer network and raise an alarm if there exists a security breach in the network [2]. To allow IDS to operate automatically or semi-automatically, machine learning techniques have been employed. Machine learning [3] can detect the correlation between features and classes found in training data, and identify relevant subsets of attributes by feature selection and dimensionality reduction, then use the data to create a model for classifying data to perform predictions.

The rest of this article is organized as follows. Section 2 succinctly introduces machine learning concepts and techniques. Section 3 reviews machine learning based intrusion detection system and in particular discusses the importance of ensemble learning and feature selection techniques. Section 4 concentrates on hybrid intrusion detection systems. Section 5 describes the NSL-KDD, Kyoto 2006 + , and KDD Cup-'99 benchmark datasets and the various performance metrics to evaluate the IDS. Section 6 discusses the research opportunities. Finally, Sect. 7 draws a conclusion.

2 Machine Learning

Machine learning is prominently applied in the area of intrusion detection systems. It is the field through which the various computer algorithms are studied, that improves incrementally through the experience and has been an area of computational learning theory and pattern recognition in artificial intelligence research since the field of its inception [4]. It helps in predictions and decisions on data, by studying algorithms and building a model from the input data. Machine learning is classified into supervised learning, semi-supervised learning, un-supervised learning, and reinforcement learning [5]. Figure 1 shows the classification of machine learning types with their corresponding methods.

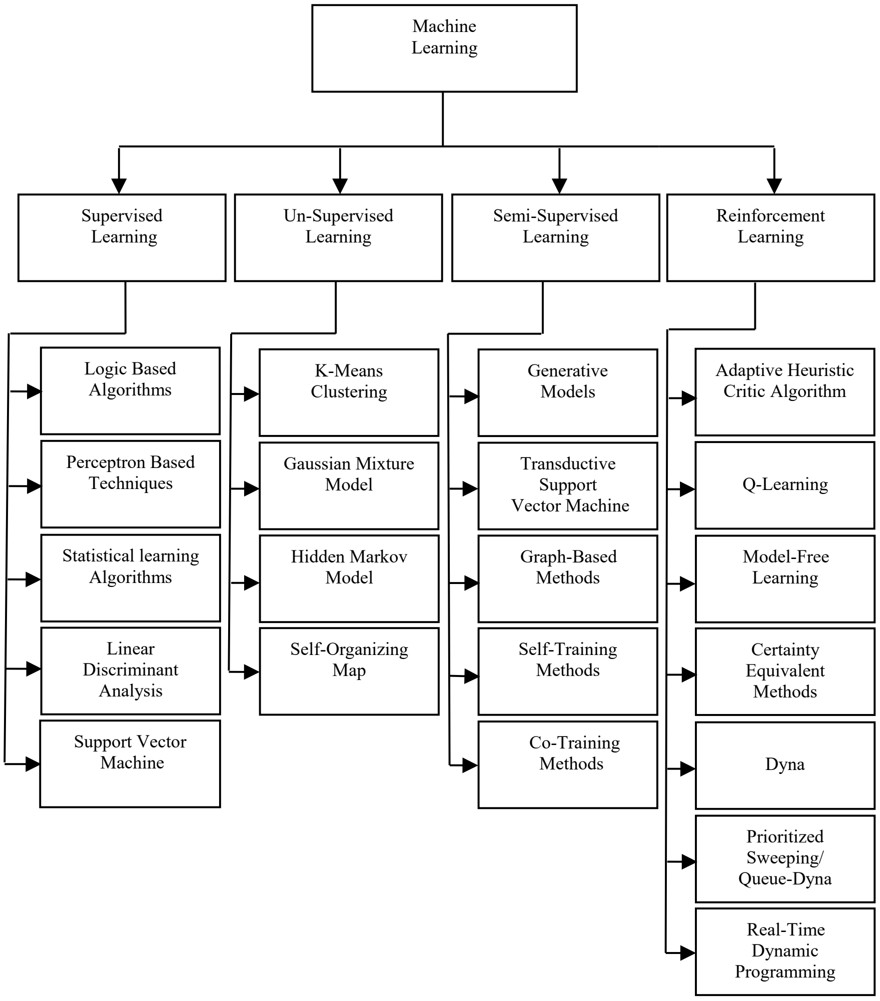

Fig. 1 Classification of machine learning system

2.1 Supervised Learning

Supervised learning is a machine learning task that assumes a function from the labeled training data. In supervised learning, there is an input variable (P) and an output variable (Q). From the input variable, the function of the algorithm is to study the mapping function to the output variable Q = f(P). Spervised learning aims to analyze the training data then produces a complete function that can be utilized to map the new instances. The learning algorithm should be able to analyze and generalize those class labels correctly from the unobserved instances. This section introduces the various algorithms used in supervised learning.

2.1.1 Logic Based Algorithms

This subsection describes the decision trees and rule-based classifiers.

Decision Trees

A decision tree [6] is a directed tree with a root node having no incoming edges. The remaining node has an incoming edge. Each of the leaf nodes is available with a label, non leaf nodes have a feature which is called a feature set. According to the distinct values in the feature set, the decision tree splits the data that falls inside the non leaf node. The testing of the feature is operated from the leaf node and its outcome is achieved until the leaf node has arrived.

Rule-Based Classifiers

Quinlan et al. [7] stated that, by transforming the decision tree into a distinct set of rules, a different path can be created for the set of rules. From the root of the tree to the leaf of the tree, a distinct rule for each path is created the decision tree is altered in to set of rules. Directly, from the training data, the rules can be inducted by different algorithms and apply these rules. The main goal is to construct the smallest set of rules that is similar to the training data.

2.1.2 Perceptron Based Techniques

Perceptron algorithm is applied to the batch of training instances for learning. The main objective is to run the algorithm frequently among the training data, till it discovers a prediction vector that is appropriate on all of the training data. For predicting the labels, this prediction rule is then applied to the test set [8]. The following subsection introduces perceptron based techniques.

Neural Network

The neural network conceptual model was developed in 1943 by McCulloch et al. [9]. It consists of different cells. The cell receives data from other cells, processes the inputs, and passes the outputs to other cells. Since then, there was intensive research to develop the ANNs. A perceptron [10] is a neural network that contains a single neuron that can receive more than one input to produce a single output.

A linear classifier contains a weight vector w and a bias b. From the given instance x, the predicted class label y is obtained according to

$$y = \operatorname{sign}(w^{\mathrm{T}}x + b)$$

With the help of weight vector w, the instance space is mapped into a one-dimensional space, afterwards to isolate the positive instances from negative instances.

2.1.3 Statistical Learning Algorithms

In statistical learning, there is a likelihood of a probability where each instance belongs to its class. The following subsection describes statistical learning algorithms.

Bayesian Network

Bayesian Network is the directed acyclic graph (DAG) [11], in which each edge correlates to the conditional dependency and each node correlates to a distinctive random variable. The Bayesian network performs the following task for learning. Firstly it learns the directed acyclic graph of the network. Secondly, the aim is to find the network parameters. Once the parameters or probabilistic parameters are found, then the parameters are induced in the set of tables, and joint distribution is remodeled by increasing exponentially the tables.

Naive-Bayesian Classifier

Naive Bayesian classifiers [12] are probabilistic classifiers with their relation related to Bayes theorem having a strong assumption of naive independence among its features. Bayes theorem can be stated in mathematical terms:

$$\mathrm{P(A/B) = P(A)(B/A)/P(B)}$$

where A and B are events.

P(A) and P(B) are events.

P(A) and P(B) are the prior probabilities of A and B.P(A/B), is a posterior probability, of the probability of observing an event A, provided that B is true. P (B/A) is known as likelihood, the probability of observing an event B provided that A is true. Naive Bayes classifier has the advantage of requiring the least execution time required for training the data.

K-Nearest Neighbor Classifiers

K-NN [13] is a nonparametric technique applied for regression and classification. In the feature space, the input of k-NN contains the k closest training examples. Then the output will depend on whether k-NN is applied for regression or classification purposes. The output is the property value for the object in k-NN regression.

2.1.4 Linear Discriminant Analysis

Linear discriminant analysis is a method to find the linear combination of the features that separate the two or more classes of objects. The resulting combination can be used as a linear classifier. A linear classifier [14] contains a weight vector *w* and a bias *b*. From the given instance x, the predicted class label *y* is obtained according to

$$y = \mathrm{sign}(w^{\mathrm{T}}x + b)$$

With the help of weight vector *w*, the instance space is mapped into a one-dimensional space, afterward to isolate the positive instances from negative instances.

2.1.5 Support Vector Machine

SVM [15] revolves around the margin on either side of a hyperplane that separates two data classes. To reduce an upper bound on the generalization error, the main idea is to generate the largest available distance between its instance on either side and separating hyperplane. The data points that lie on the margins of the optimum hyperplane are termed as Support Vector Points, and it is characterized as the linear combination of these points. An alternative data point is neglected. The different features available on the training data do not affect the complexity of SVM. This is the main reason the SVM is employed with learning tasks having a large number of features with respect to the number of training data.

Large problems of the SVM cannot be solved as it not only contains the large quadratic operations but also there exist numerical computation which makes the algorithm slow in terms of processing time. There is a variation of SVM called Sequential Minimal Optimization (SMO). SMO can solve the SVM quadratic problem without employing any extra matrix storage and without using any numerical quadratic programming optimization steps [16].

2.2 *Un-Supervised Learning*

In Un-Supervised learning the data is not labeled, more precisely, there are unlabeled data. Un-Supervised learning consists of an input variable (P), but there is no output variable. The representation is seen as a model of data. Un-Supervised learning aims to obtain the hidden structures from unlabeled data or to infer a model having the probability density of input data. This section reviews some of the basic algorithms used in Un-Supervised learning.

2.2.1 *K*- Means Clustering

K-means clustering [17] partitions the *N* observation in space into *K* clusters. The information and nearest mean belongs to this cluster and works as a model of the cluster. As a result, the data space splits into Voronoi cells. *K*-means algorithm is an iterative method, which starts with a random selection of the *k*-means $v_1, v_2 \ldots v_k$. With each number of iteration the data points are grouped by k-clusters, keeping with the closest mean to each of the points, mean is then updated according to the points within the cluster. The variant of *K*-means is termed as *K*-medoids. In *K*-medoids, instead of taking the mean, the larger part of the cluster having the centrally located data point is investigated as a reference point of the corresponding cluster [18].

2.2.2 Gaussian Mixture Model

The Gaussian mixtures were popularized by Duda and Hart in their seminal 1973 text, Pattern Classification and Scene Analysis [19]. A Gaussian Mixture is a function that consists of several Gaussians, each identified by $k \in \{1, \ldots, K\}$, where K represents the number of clusters of a dataset. Each Gaussian is represented by the combination of mean and variance, consisting of a mixture of *M* Gaussian distributions. The weight of each Gaussian will be a third parameter related to each Gaussian distribution in a Gaussian mixture model (GMM).

When clustering is performed applying Gaussian Mixture Model, the goal is to obtain a criterion such as mean and covariance of each distribution and the weights, so that the resulting model fits optimally in the data. To optimally fit the data, one should enlarge the likelihood of the data given in the Gaussian Mixture Model. This can be achieved by using the iterative expectation maximization (EM) algorithm [20].

2.2.3 Hidden Markov Model

HMM [21] is a parameterized distribution for sequences of observations. A hidden Markov model is a Markov process that is divided into two components called observable components and unobservable or hidden components. That is, a hidden Markov model is the Markov process (Y_k, Z_k) k $\geq$ 0 on the state space $C \times D$, where we presume that we have a means of observing Y_k, but not Z_k as the signal process and C as the signal state space, while the observed component Y_k is called the observation process and D is the observation state space.

2.2.4 Self-organizing Map

A self-organizing map (SOM) [22] is a form of an artificial neural network, trained by applying unsupervised learning to yield a low dimensional, discretized representation of input space of the training samples called maps. SOM applies the competing learning while ANN applies the error-correction learning (backpropagation with gradient descent). To retain the topological properties of the input space the SOM applies the neighborhood function. SOM begins with initializing the weight vectors. Sample vectors are chosen randomly and the map of the weight vectors is inspected to find the best weight that represents that sample. The weight vector has neighboring weights that are close to each other. The chosen best weight sample and neighbors of that weight are rewarded and is more likely to become the randomly selected sample vector. This strategy allows the maps to grow and take the different shapes such as rectangular, square, and hexagonal in two-dimensional feature spaces.

2.3 Semi-supervised Learning

Semi-Supervised learning is the sequence of labeled and unlabeled data. The labeled data is very sparse while there is an enormous amount of unlabeled data. The data is used to create an appropriate model of the data classification. Semi-Supervised learning aims to classify the unlabeled data from the labeled data. This section explores some of the most familiar algorithms used in Semi-Supervised learning.

2.3.1 Generative Model

Generative model [23] considers a model $p(x, y) = p(y)p(x/y)$ where $p(x/y)$ is known as mixture distribution. The mixture components can be analyzed when there are large numbers of unlabeled data is available. Let $\{p_\theta\}$ be the distribution family and is denoted by parameter vector θ. θ can be identified only if $\theta_1 \neq \theta_2 \Rightarrow p_{\theta_1} \neq p_{\theta_2}$ to a mixture components transformation. The expectation–maximization (EM) algorithm is applied on the mixture of multinomial for a task of text classification [24].

2.3.2 Transductive Support Vector Machine

TSVM [25] extends the Support Vector Machine (SVM) having the unlabeled data. The idea is to have the maximal margin among the labeled and unlabeled data on its linear boundary by labeling the unlabeled data. Unlabeled data has the least generalization error on a decision boundary. The linear boundary is put away from the dense region by the unlabeled data. With all the available estimation solutions to TSVM, it is curious to understand just how valuable TSVM will be as a global optimum solution. Global optimal solution on small datasets is found in [26]. Overall an excellent accuracy is obtained on a small dataset.

2.3.3 Graph Based Methods

In Graph based methods [27] labeling information of each sample is proliferated to its neighboring sample until a global optimum state is reached. A Graph is constructed with nodes and edges where nodes are defined with labeled and unlabeled samples, while the edges define the resemblance among labeled and unlabeled data. The label of the data sample is advanced to its neighboring points. The graph based techniques are the focus of interest among researchers due to their better performance. Kamal and Andrew [28] propose semi-supervised learning as the graph mincut (known as st-cut) problem. Szummer et al. [29] performed a step Markov random walk on the graph.

2.3.4 Self Training Methods

Self-training is a methodology applied in semi-supervised learning. On a small amount of data, the classifier is trained and then applied to classify the unlabeled data. The most optimistic unlabeled points, together with their labels predicted, are then added to the training set. The classifier is again trained with the training dataset. This procedure goes on repeating itself. For teaching itself, the classifier had its own predictions. This methodology is called bootstrapping or self-teaching [30]. Various natural language processing tasks apply the methodology of self teaching.

2.3.5 Co-Training

Co-Training [31] is applied where there are a large amount of unlabeled data and a small amount of labeled data. In Co-Training, there is a general assumption that the features can be divided into feature subsets. Every feature subset is adequate for training the classifier, and for the given class the feature sets are relatively independent.

There are two classifiers primarily that are trained on a labeled data on the pair of feature subsets. With labeled and unlabeled data, each classifier trains the other classifier to retain the supplementary training sample from another classifier and this process gets repeated often until optimal features are found.

2.4 Reinforcement Learning

In reinforcement learning, the software agent gathers the information from the interaction, with the environment to take actions that would maximize the reward. The environment is formulated as a markov decision process. In reinforcement learning, there is no availability of input/output variables. The software agent receives the input *i*, the present state of environment *s*, then the software agent determines an action *a*, to achieve the output. The software agent is not told which action would be best in terms of long term interest. The software agent needs to gather information about the states, actions, transition, rewards for optimal working. This section reviews algorithms used in reinforcement learning.

2.4.1 Adaptive Heuristic Critic

The adaptive heuristic critic algorithm is a policy iteration, adaptive version [32] is implemented by Sutton's *TD(0)* algorithm [33]. Let *((a, w, t, a'))* be the tuple representing a single transformation in an environment. Let *a*, be the state of an agent before its transition, *w* is the action of its choice, *t* is the award it receives instantaneously, and *a'* is culminating state. The policy value is studied by Sutton's *TD(0)* algorithm which utilizes the update rule given by:

$$V(a) := V(a) + w(t + tV(a') - V(a)).$$

When the state *a* is visited, its estimated value is amended to be adjacent to $(t + tV(a'))$ since *t* is the spontaneous reward received and *V(a')* is the approximate rate of the next appearing state.

2.4.2 Q-Learning

Q-learning [34] is a type of model free reinforcement learning. It may also be known as an approach of asynchronous dynamic programming (DP). Q-learning allows the agents to have the ability to learn and to perform exemplary in the markovian field by recognizing the effects of its actions, which is no longer required by them to build domain maps. Q-learning finds an optimal policy and it boosts the predicted value of the total reward, from the beginning of the current state to any and all successive steps, for a finite markov decision process, given the infinite search time and a partially random policy. An optimal action-selection policy can be associated with Q-learning.

2.4.3 Certainty Equivalent Methods

The certainty equivalent method [35] is studied continuously throughout the agent lifespan. The present model is applied to calculate the optimum policy and functional value. The certainty model effectively uses the available data on rather side, it is computationally costly even for small state areas.

2.4.4 Dyna

Suttons Dyna Architecture [36] applies a yielding approach that tends to be more successful and is more computationally effective than certainty-equivalent methods. Markov decision process is defined by a tuple *(a, w, t, a')* and it acts as follows: The model gets transited state from *a* to *a'* upon on action and getting the reward *t* by taking action *w* in the state *a*. Updated models are then *M* and *P*. Dyna updates the state *a* by applying the rules that are an adaptation of the value iteration updating for the *Q* rules.

2.4.5 Prioritized Sweeping

The limitations with Dyna are proportionately undirected. When the target has just been reached it goes on to update its random state action pairs, instead of concentrating on the curious part of the state space. Prioritized sweeping addresses this problem by choosing the updates at random, the values are associated with space rather than state-action pairs. Additional information has to be stored in the model for having the fitting choices. Every state remembers its previous state, having the non zero transition probability. The priority of each state is set initially to zero. Prioritized sweeping updates the k states having the highest priority, rather than updating the k random state action pairs [37].

2.4.6 Real Time Dynamic Programming

The RTDP [38] applies Q learning. The computing is done on the areas, where the agent is coherent to occupy the state space. RTDP is specific to problems in which the agent tries to attain the particular goal state and the reward everywhere else is 0. RTDP finds the shortest path from the start state to the target state without visiting the unnecessary state space.

2.5 Ensemble Learning

Ensemble learning [39] is an imperative study in the domain of Machine Learning, Data mining, Pattern Recognition, Neural networks, and Artificial Intelligence. The goal of ensemble learning methods is to build the set of learners or classifiers then merge them so that it can make an accurate prediction, in contrast to single learner the accurate prediction may not be good as a set of learners. There are several learners in an ensemble learning called base learners. The goodness of ensemble learning is that they are capable of boosting the base learners. The base learners are sometimes called the weak learners. Therefore the ensemble learning is also called as learning of multiple classifier systems. In this section, few important concepts of ensemble learning techniques are discussed.

2.5.1 Boosting

Boosting [40] is an ensemble learning paradigm, which builds a strong classifier from the number of base learners. Boosting works sequentially by training the set of base learners to combine it for the prediction. The boosting algorithm takes the base learning algorithms repeatedly, having the different distributions or weighting of training data on the base learning algorithms. On running the boosting algorithm, the base learning algorithms will generate a weak prediction rule, until many rounds of steps the boosting algorithm combines the weak prediction rule into a single prediction rule that will be more accurate than the weak prediction rule.

2.5.2 Bagging

Bootstrap AGGregatING [41]. Bagging algorithms combine the bootstrap and aggregation and it represents parallel ensemble methods. Bootstrap sampling [42] is applied by bagging to acquire the subsets of data for training the base learning algorithms. Bagging applies the aggregating techniques such as voting for classification and averaging for regression. Bagging uses a precedent to its base classifiers to obtain its outputs, votes its labels, and then obtains the label as a winner for the prediction. Bagging can be applied with binary classification and multi-class classification.

2.5.3 Stacking

In Stacking [43] the independent learners are combined by the learner. The independent learners can be called cardinal learner, while the combined learner is called Meta learner. Stacking aims to train the cardinal learner with initial datasets to generate the contemporary dataset to be applied to the Meta learner. The output generated by the cardinal learner is regarded as input features, while the original labels are still

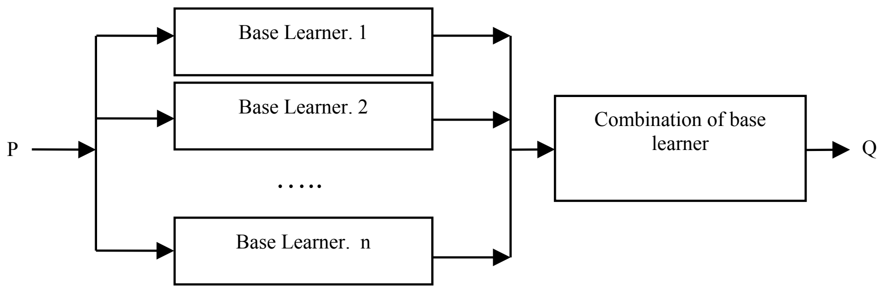

Fig. 2 A basic architecture of ensemble learning

regarded as labels of the contemporary dataset. The cardinal learner is obtained by applying different learning algorithms, often generating composite stacked ensembles, although we can construct uniformly stacked ensembles. The contemporary dataset has to be obtained by the cardinal learner otherwise, the dataset is the same for the cardinal and Meta learner there is speculation of over fitting. Wolpert et al. [44] emphasized various features for the contemporary dataset and categories of learning algorithms for the Meta Learner. Figure 2 shows the basic architecture of ensemble learning.

2.6 Feature Selection

Feature selection is a well-studied research topic in the field of artificial intelligence, machine learning, and pattern recognition. Feature selection removes the redundant, irrelevant, and noisy features from the original features of datasets by choosing the relevant features having the smaller subdivision of the dataset. By applying the various techniques of feature selection to the datasets, results in lower computational costs, higher classifier accuracy, reduced dimensionality, and a predictable model. Existing feature selection methods for machine learning fall into two broad categories. The learning algorithms that classify relevant and useful features applied to the data are termed as the wrapper method and those evaluate the merit of features by using heuristics based on general characteristics of data are termed as filter method [45]. This section focuses on the characteristics of feature selection algorithms, introduces the filter and wrapper approaches.

2.6.1 Characteristics of Feature Selection Algorithms

The feature selection algorithm does its task by performing the exploration in the space having feature subsets, and as a result, should define the four fundamental problems influencing the variety of the search [46].

- *Starting point*. The selection of points in a feature subset space affects the direction of the search. There are different choices to begin the search. The search space can start with no features to successfully adding the attributes. Within the search space, the search moves forward, the subsequent possibility is that the search space begins with all features and adequately removes them. Another alternating technique is to search the space in the middle and then moves outwardly from this point.
- *Search organisation.* Whenever there is data having the features containing the larger number, the searching of the feature subset within the space and time frame is decisive. There exists a 2^N possible subsets for the N number of features. As compared to exhaustive search techniques the heuristic search techniques are more practicable. Greedy hill climbing is a strategy that considers the single feature that can consecutively be added or deleted from the current feature subset. Whenever the algorithm favors the inclusion and elimination of the feature subset then it is termed as forward selection and backward elimination [47].
- *Evaluation strategy*. The biggest challenge among the feature selection algorithms for machine learning is the classification of feature subsets [48]. In this strategy, the algorithm eliminates the undesirable features before the learning starts. The algorithm applies the data having a heuristic based on general characteristics to calculate the goodness of the feature subset. When selecting the features, the bias of the induction algorithm should be considered. This method is called Wrapper, which applies an induction algorithm, with re-sampling techniques to obtain the accuracy of the feature subsets.
- *Stopping criterion.* An important decision of feature selection is to determine when to terminate the searching space of feature subsets. A feature selector should terminate the searching space of feature subsets by summing or eliminating the features when the feature selector finds no one of its substitutes. It enhances the quality of a prevailing feature subset. Concurrently, the algorithm may consider carrying on, altering the feature subset until the quality of the feature subset does not degrade.

2.6.2 Filter Methods

The earlier approaches that were introduced to feature selection are filter methods. Filter methods apply the data that has an examining property in general to calculate the goodness of feature subset, except a learning algorithm that evaluates the quality of the feature subsets. Filter algorithm executes many times faster than wrapper so there is a high probability of scaling the databases having a large number of features. Filter does not require the re-execution for the different learning algorithms. This strategy makes filter approaches quicker than wrapper methods. There are major drawbacks of the filter methods such as some filter algorithm is unable to handle the noisy data, some algorithms need the user to specify the level of noise, and in some

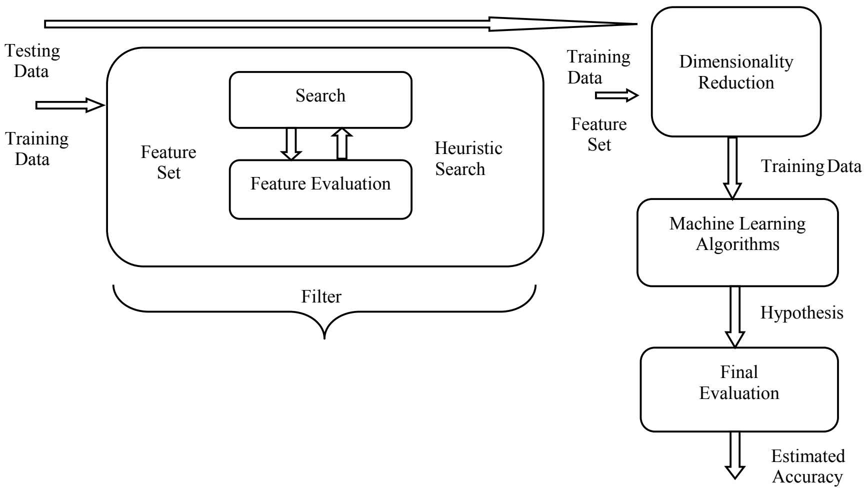

Fig. 3 A filter feature selector

cases, the features are ranked and the final choice is left to the user. In other cases, the users need to specify the required number of features. Filter methods are applied mainly where the data is having highly dimensionality rate. Figure 3 shows the filter feature selector.

2.6.3 Wrapper Methods

For the selection of best feature subsets, a wrapper method applies an induction algorithm for the feature selection.

The idea of wrapper approaches is that since induction methods are using the feature subset it should enhance the accuracy than a separate measure that applies distinct inductive bias [49]. Wrapper approaches are known to obtain better results than filter methods because they are specifically adapted to the interaction between the training data and induction algorithm. Whereas the wrapper methods are much slower than filter methods as the wrapper methods, frequently call the induction algorithm and reruns when a different induction algorithm is applied. Figure 4 shows the wrapper method.

2.6.4 Principal Component Analysis

Principal component analysis (PCA) [50] is an analytical procedure that converts the correlated variables into linearly uncorrelated variables, with the help of an orthogonal transformation. These are named as principal components. The PCA is a multivariate dimensionality reduction tool that extracts the features representing much of

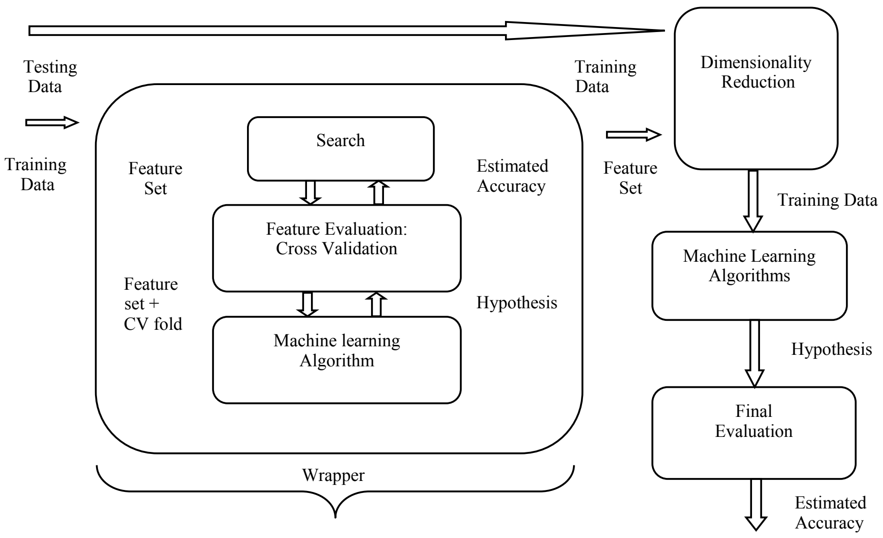

Fig. 4 A wrapper feature selector

the features from the given data and thus removing the unimportant features having less information without losing the crucial information in data. When real data is collected, the random variables describing the data attributes are presumed to be highly correlated. The correlation between random variables can be found in the covariance matrix. The aggregate of the variances will give the overall variability.

2.6.5 Classification and Regression Trees

CART [51] is a binary decision tree that is constructed by splitting a node into two child nodes repeatedly.

The whole learning samples contains at the beginning with the root node. The basic idea behind the tree growth is to choose a split among all the possible splits at each node so that the resulting child nodes obtained are the purest. CART algorithm considers the univariate splits i.e. each split depends on the value of one predictor variable. Classification analysis of the tree is performed when the predicted outcome is the class to which the data belongs, while the regression analysis of the tree is done when the predicted outcome can be considered as a real number.

3 Machine Learning Based Intrusion Detection System

3.1 Intrusion Detection System (IDS)

The IDS literature history starts with a paper by James Anderson [52]. The main challenges of the organization are to protect the information from network threats. A string of actions through which intruder gains control of the system [53]. The main aim of intrusion detection is to detect previously known and unknown attacks, to learn and comply with new attacks and detect intrusions in an appropriate period. IDS are classified according to three main approaches: Implementation, Architecture, and Detection Methods [54]. Figure 5 shows the classification of the Intrusion Detection System.

3.1.1 Implementation Classification

The Intrusion Detection System is mainly classified according to Host-Based Intrusion detection System (HIDS) and Network-Based Intrusion Detection Systems (NIDS).

Host-Based Intrusion Detection Systems

HIDS positions the sensors at an individual host of the network sensor. It collects the relevant information about data, based on various events. The sensor also records the data related to system logs, registry keys, and other logs generated by the operating system process called inspection logs or audit trail and any other unauthorized action. The HIDS rely heavily on inspection logs, degrading the performance of hosts related to bandwidth which can lead to high computational costs. The HIDS is better known

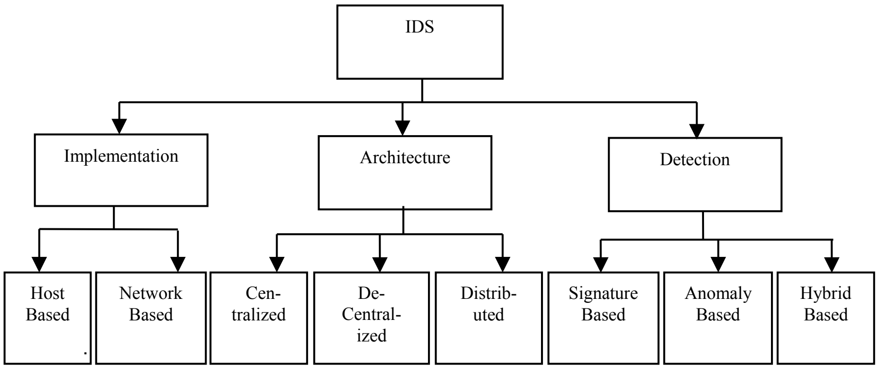

Fig. 5 Classification of intrusion detection system

to trace malicious activities regarding specific users and also keeping a track record of the behavior of individual users within an organization.

Network Based Intrusion Detection Systems

It collects the information from networks and checks for attacks or irregular behavior by examining the contents and header information and flow information of all packets across all networks. NIDS collects the information from networks examines the contents and the header information regarding all packets in various networks and then validates the irregular behavior of the network by verifying the various contents and header information of various packets whether there is an attack or not. The network sensor is having various sensors, a with built-in mechanism with attack signatures that define rules. These sensors also allow the user to define their signature for verifying the new attacks in the networks.

NIDS is computationally costly and also time-consuming, as it has to inspect each flow and header level of packets from various networks, on the other side the NIDS is operating system and platform-independent that does not require any modification when NIDS operates. This makes NIDS more scalable and robust compared with HIDS.

3.1.2 Architectural Classification

The architectural classification of IDS is classified into three categories:

Centralized IDS

The analysis and detection of monitored data are analyzed inside the CPU. There are various networks and different sensors present inside the centralized IDS. Inside the centralized IDS, different sensors monitor the data that are sent from various networks. Whenever there is a need for expansion of the network, the central processing unit is overloaded that is the major disadvantage of centralized IDS.

Decentralized IDS

The CPU overloading problem is solved in decentralized IDS architecture. The multiple sensors and multiple processing units are scattered across the network. The gathered data are send to the nearest processing unit for getting processed, before they are sent to the main processing unit.

Distributed IDS

In distributed IDS the work load of the CPU is distributed among all its peers. The multiple IDS are distributed over the large network and its fundamental principle is that the multiple intrusion detection systems will communicate through peer to peer architecture.

3.1.3 Detection Methods

The detection methods of intrusion detection systems are classified into three major types: Signature-based, Anomaly-based and Hybrid-based. This section discuss various machine learning approaches used in signature, anomaly and hybrid IDS.

Signature-Based IDS

The signature-based intrusion detection system stores the previously known or existing signature in a database and it looks for malignant activities and malignant packets in the network. It measures the similarity and patterns of the attacks within the existing signature that is saved in the database. If the similarity or pattern is found then an alarm is generated. Signature-based IDS are accurate in detecting the previously known attacks, but they cannot recognize the new attacks.

Some of the machine learning algorithms that were used in signature-based IDS are Bayesian network [55], Decision Trees [56], SVM [57], ANN [58], K-NN [59], and Graph-based approaches [60].

Anomaly-Based IDS

In anomaly-based IDS new action profiles are created and are applied to analyze the outliers that diverge from the profiles, Chandola et al. [61]. Anomaly-based intrusion detection system depends on analytical techniques to construct the various attack forecast models. It has the leverage of detecting anonymous attacks that do not have the extant signature in the database, but the major disadvantage is creating new action profiles. Sometimes the outliers that diverge from new action profiles are not always an attack, this creates incorrectly classified as an attack or False Positive. Some of the anomaly-based IDS that uses various machine learning techniques are Probability-based technique [62], Fuzzy rule classifiers using NN technique [63], Neural Network using back propagation technique [64], One class support vector machine [65], and K- NN [66].

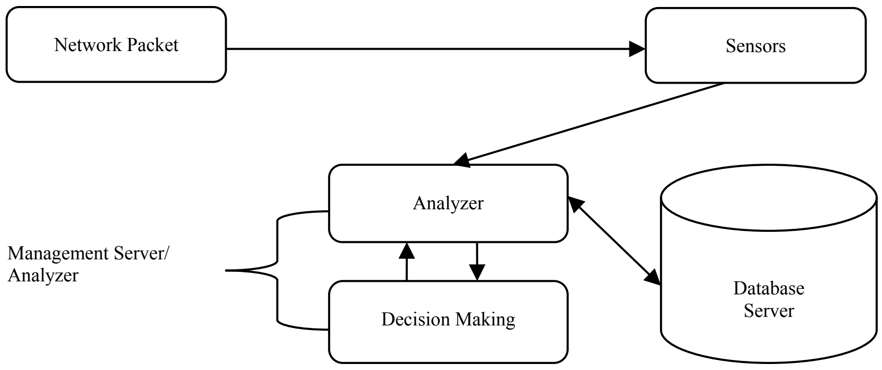

Fig. 6 A basic architecture of intrusion detection system

Hybrid-Based IDS

The hybrid-based IDS combines both signature and anomaly-based detection approaches to detect attacks. The main advantage of a hybrid intrusion detection system is that it combines the existing features from the anomaly and signature-based IDS. The downside of hybrid-based techniques is its increased execution time of applying both matchings of signatures and anomaly detection to evaluate the incoming of the network connections. The techniques that use the hybrid-based IDS using machine learning techniques are combined NN and SOM [67], combining ANN, SVM, and MARS [68], etc. Figure 6 shows the basic architecture of an intrusion detection system.

4 Hybrid Intrusion Detection Systems

Hybrid-based intrusion detection systems were enormously applied as they produce a better result. The authors concluded that the hybrid-based approaches can overcome the one approach over the other [69]. Neha and Shailendra [70] conclude that hybrid-based IDS are relevant methods to get good accuracy and proposed, intelligent water drops (IWD) algorithm described by Shah–Hosseini [71] based feature selection approach. This strategy applies the IWD algorithm, an optimization algorithm that is inspired by the nature of the selection of subset features, combined with SVM as a classifier to evaluate the chosen features. The search of the subset is performed by the IWD algorithm while the evaluation task of the subset is carried by the classifier. Overall this approach reduces the total of 45 features to 10 features from the input dataset. The proposed approach achieved a detection rate, precision, and accuracy of 99% on the KDD Cup-‘99 dataset. Arif et al. [72] introduced a Hybrid approach for intrusion detection systems where PSO is applied to prune the node while pruned decision trees are applied as a classification technique for NIDS. Particle Swarm

Optimization is a population-based stochastic optimization technique based on bird flocking and fish schooling. The PSO algorithm works by having the population called the swarm and the potential solutions called particles. PSO performs the searching through a swarm of particles that updates through repetitions. Each particle advances in the direction to its previous best position and globally best position for finding the most promising solution. The above approach applies the single and multi-purpose particle swarm optimization algorithm. The evaluation is done on the KDD Cup-'99 dataset. Thirty arbitrary samples of the datasets are preferred from 10% of KDD Cup-'99 training and testing datasets. The estimate of records in individual training, as well as testing datasets, is 24,000 and 12,000 jointly for the evaluation purposes. A precision of 99.98% and an accuracy of 96.65% is achieved using the above approaches.

Ahmed and Fatma [73] applied a triple edge strategy to develop a Hybrid Intrusion detection system in which the Naïve Bayes feature selection (NBFS) technique has been applied for dimensionality reduction, Optimized Support Vector Machines (OSVM) is applied for outlier rejection and Prioritized K-Nearest Neighbours (PKNN) classifier is employed for classifying the input attacks. Different datasets are applied such as the KDD Cup-'99 dataset, NSL-KDD dataset, and Kyoto 2006 + dataset. 18 effective features are selected from the KDD Cup-'99 dataset having a detection rate of 90.28%. 24 features are selected from Kyoto 2006 + dataset with a detection ratio of 91.60%. The author has compared with previous work and has a best overall detection ratio of 94.6%. Tirtharaj Dash et al. [74] report two new hybrid intrusion detection methods that are gravitational search (GS) and sequence of gravitational search and particle swarm optimization called (GSPSO). Gravitational search is a population-based heuristic algorithm that is based on mass interactions and the law of gravity. It comprises of search agent who interacts among each other with heavier masses by the gravitational force and its performance is evaluated by its mass. The slow movement of the search agent assures the good solution of the algorithm. The combinational approach has been carried out to train ANN with models such as GS-ANN and GSPSO-ANN. The random selection of 10% features is selected for training purposes, while 15% is used for testing purposes is applied successfully for intrusion detection purposes. The KDD Cup-'99 dataset was applied as a metric for calculation. Normalisation of dataset was done for uniform distribution by MATLAB. An average detection ratio of 95.26% was achieved.

Yao and Wang [75] introduced a hybrid framework for IDS, K-Means algorithm is employed for clustering purposes. In the classification phase, many machine learning algorithms (SVM, ANN, DT, and RF) which are all supervised learning algorithms are compared on different parameters. The supervised learning algorithm has various parameters for different kinds of attacks (DoS, U2R, Probe, and R2L). The proposed hybrid algorithm has achieved an accuracy rate reaching 96.70% on the KDD Cup-'99 dataset. Alauthaman et al. [76] proposed a technique of peer to peer, bot detection based on an adaptive multilayer feed forward neural network in cooperation with decision trees. CART is used as a feature selection technique to prefer the relevant features. With all those features, a multilayer feed-forward NN training model is built applying the resilient back-propagation learning algorithm. Network traffic reduction techniques were applied by using six rules to pick the most relevant features. Twenty-nine features are selected from six rules. The proposed approach has an accuracy rate of 99.20% and a detection rate of 99.08% respectively on ISOT and ISCX datasets.

Sidra and Faheel [77] introduce a genetic algorithm, which is based on vectors. In this technique vector chromosomes are applied. The novelty of this algorithm is to show the chromosomes as a vector and training data as metrics. It grants multiple pathways to have the fitness function. Three feature selection techniques are chosen, linear correlation-based feature (LCFS), modified mutual information-based feature selection (MMIFS), and forward feature selection algorithm (FFSA). The novel algorithm is tested in two datasets (KDD Cup-'99 and CDU-13). Performance metrics show that vector-based genetic algorithm has a high DR of 99.8% and a low FPR of 0.17%. Saud and Fadl [78] introduced an IDS model applying the machine learning algorithm to a big data environment. This paper employs a Spark-Chi-SVM model. ChisqSelector is applied as a feature selection method and constructed an IDS model by applying the SVM as a classifier. The comparison is done with Spark Chi-SVM classifier and Chi- Logistic regression classifier. KDD Cup-'99 dataset is applied for the metrics evaluation process. The result shows Spark Chi-SVM model has good performance having an AUROC of 99.55% and AUPR of 96.24%. Venkataraman and Selvaraj [79] report the hybrid feature selection structure for the efficient classification of data. The symmetrical uncertainty is employed to find the finest features for classification. The Genetic algorithm is applied to find the finest subset of the features with greater accuracy.

The author combined (SU-GA) as a hybrid feature selector. Matlab and Weka tools are applied to develop the proposed hybrid feature selector. Different classification algorithm (J48, NB, SMO, DT, JRIP, Kstar, Rand Frst, Multi Perptn,) are used to classify different attacks. The average learning accuracy with Multi Perpn and SU-GA is highest having 83.83% on the UCI dataset. Neeraj et al. [80] applied knowledge computational intelligence in network intrusion detection systems. This paper introduces an intelligent and Hybrid NIDS model. This model, then integrates the fuzzy logic controller, multilayer perception, adaptive Neuro-Fuzzy interference system, and a Neuro fuzzy genetic. The proposed system has trio modules collector, analyser, and predictor modules for gather and filter network traffic, classifying the data, and to prepare the final decision in assuming knowledge on the accurate attack. The experiment is performed on the KDD Cup-'99 dataset and achieved a false alarm detection accuracy of 99%. Akash and Khushboo [81] implemented a hybrid technique. DBSCAN is applied for choosing the enhanced features for IDS and eliminating the noise from data. Based on Euclidean distance and minimum number of points DBSCAN groups together the points that are close to each other. For grouping data, K means clustering is proposed. SMO (sequential minimal optimization) classifier is implemented for classification purposes. The experimental setup is performed on the KDD Cup-'99 dataset with certain attributes. The approach DBKSMO achieved an accuracy of about 98.1%. MATLAB and WEKA tools are applied to execute the full process.

Rajesh and Manimegalai [82] introduced an Effective IDS applying flawless feature selection outlier detection and classification. The author employs a feature selection technique named intelligent flawless feature selection algorithm (IFLFSA) to select the finest number of features that are effective for identifying the attacks. To identify the outliers from the dataset, an entropy-based weighted outlier detection (EWOD) approach is applied. An intelligent layered technique is employed for efficient classification. The KDD Cup-'99 dataset is applied for experimental purposes. The comprehensive detection accuracy wraps the detection accuracy on four categories of attacks namely DoS, Probe, U2R, and R2L. The detection accuracy of the suggested system has achieved the rate of 99.45%. Unal et al. [83] implemented a hybrid approach for IDS using machine learning approaches. Naive bayes, K-Nearest neighbor is used for classification purposes, while random forest and J-48 algorithm are used as a decision tree to train the algorithm on different types of attack. The author proposed a CfsSubsetEval approach and WrapperSubsetEval as the two feature selection techniques. NSL-KDD dataset is used as an evaluation process. The overall accuracy of 99.86% is obtained on all types of attacks. Pragma et al. [84] introduce a hybrid IDS to the hierarchical filtration of anomalies (HFA). This work proposes an ID3, a decision tree, which is employed for classifying data into corresponding classes. K-NN is used to assign the class label to an unknown data point that is based on its class labels of the k nearest neighbors. Isolation forest is introduced to isolate an anomaly against normal instances. The proposed algorithm (HFA) is performed on the NSL-KDD dataset and KDD Cup-'99 dataset. The overall performance on the KDD Cup-'99 dataset, has an accuracy rate of 96.92%, detection ratio of 97.20% and FPR of 7.49%. The proposed algorithm performance with the NSL-KDD dataset has an accuracy rate of 93.95% has a detection ratio of 95.5%, and an FPR of 10.34%. Sushruta and Chandrakanta [85] report a new approach named BFS-NB hybrid model in IDS. This paper proposes the best first search technique for dimensionality reduction and was employed for an attribute selection technique. The Naïve Bayes classifier is implemented as a classification technique and to improve the accuracy of identifying the intrusions. The BFS-NB algorithm is performed on the KDD dataset gathered from the US air force. The classification accuracy of the BS-NFB is 92.12% while sensitivity and specificity analysis of 97 and 97.5% is achieved. Table 1 depicts the Taxonomy of the latest hybrid intrusion detection methods.

5 Evaluations of Intrusion Detection System

This section describes the features of KDD Cup-'99, NSL-KDD, and Kyoto 2006 + datasets and different metrics to classify the performance of the IDS.

Table 1 Taxonomy of latest hybrid intrusion detection methods

Year	Authors/research papers	Algorithms	Techniques	Dataset	Evaluation criteria	Feature selection	Results
2017	Neha Acharya, Shailendra Singh: An IWD-based feature selection method for intrusion detection system. Soft Computing Springer 2017. pp: 4407–4416. DOI 10.1007/s00500-017-2635-2	IWD, SVM	Feature reduction using IWD (Intelligent water drops), SVM, is used as a classifier	KDD Cup-‘99 Dataset	Precision rate, Detection rate, False Alarm rate, Accuracy rate	Yes	Achieves a precision rate of 99.10%, Detection rate of 99.40%, Accuracy rate of 99.05%, False alarm of 1.40%
2017	Arif Jamal Malik, Farrukh Aslam Khan: A hybrid technique using binary particle swarm optimization and decision tree pruning for network intrusion detection. Cluster Comput Springer 2017. pp: 667–680. DOI 10.1007/s10586-017-0971-8	Particle swarm optimization (PSO)	Particle swarm Optimization (PSO) algorithm/PSO are used for node pruning of Decision Trees and the pruned Decision Trees is used for classification of the network intrusions	KDD Cup-‘99 Dataset	Precision rate, Accuracy rate, IDR, FPR. Time	Yes	A precision of 99.98%, Accuracy of 96.65%, IDR of 92.71%, FPR of 0.136 and Evaluation performance of 383.58 sec. is achieved
2017	Ahmed I. Saleh, FatmaM. Talaat, LabibM. Labib: A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM Classifiers. Artif Intell Springer 2017. pp: 403–443. DOI 10.1007/s10 462-017-9567-1	Naïve based feature selection, Optimized SVM Algorithm, Prioritized KNN Algorithm	Hybrid HIDS strategy (Based on Naïve based feature selection), Naïve based is used as feature selection, OSVM is used for outlier rejection. PKNN is used for classifying input attacks	KDD Cup-‘99 NSL-KDD and Kyoto 2006 + Dataset	Detection Rate, Sensitivity, Specificity, Precision	Yes	An overall detection rate of 94.6%, Sensitivity rate of 53.24%, Specificity of 98.21%, Precision of 56.62%, is achieved on all the datasets

(continued)

Table 1 (continued)

Year	Authors/research papers	Algorithms	Techniques	Dataset	Evaluation criteria	Feature selection	Results
2017	Tirtharaj Dash: A study on intrusion detection using neural networks trained with evolutionary algorithms. Soft Comput Springer 2017. pp: 2687–2700. DOI 10.1007/s00500-015-1967-z	Artificial Neural Network	Gravitational search (GS), and combination of GS and particle Swarm optimization (GSPSO) are implemented to train ANN	NSL-KDD Dataset	Detection rate, Mean Squared Error MSE, Time	No	A detection ratio of 95.26%, MSE of 0.4527% and training time of 103.70 seconds is achieved
2017	Haipeng Yao, Qiyi Wang: An Intrusion Detection Framework Based on Hybrid Multi-Level Data Mining. Int J Parallel Prog. Springer 2017. pp: 1-19. DOI 10.1007/s10766-017-0537-7	Hybrid Multilevel data mining algorithm	MH-DE (Multilevel Hybrid Data Engineering), MH-ML (Multilevel Hybrid Machine Learning, MEM (Micro Expert Module) to train on KDD Cup-‘99 dataset	KDD Cup-‘99 Dataset	Accuracy rate, Detection rate, Precision rate, Recall, F-Value	Yes	Accuracy of 96.70%, Precision of 96.55%, F-Value of 96.60%, Recall of 96.70% and on all types of attack a detection rate of 66.69% is achieved
2018	Mohammad Alauthaman, Nauman Aslam, Li Zhang, Rafe Alasem, M. A. Hossain: A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks. Neural Comput & Applic 29, Springer 2018. pp. 991–1004. DOI 0.1007/s00521-016-25645	Neural Network with resilient back propagation algorithm, CART	Feature reduction is done by CART, Neural Network with resilient back propagation algorithm for updating the weights	ISOT & ISCX dataset.	Accuracy rate, Detection rate, FPR	Yes	Detection of 99.08%, An accuracy rate of 99.20%, and FPR of 0.75% is achieved

(continued)

Table 1 (continued)

Year	Authors/research papers	Algorithms	Techniques	Dataset	Evaluation criteria	Feature selection	Results
2018	Sidra Ijaz, Faheel A. Hashmi: Vector Based Genetic Algorithm to optimize predictive analysis in network security: Applied intelligence, Volume 48, issue 5, Springer 2018. pp: 1086-1096. DOI 10.1007/s10489-017-1026-9	Vector based genetic algorithm	Chromosomes as vector and training data as metrics'	CTU-13 dataset and KDD Cup-'99 dataset	Accuracy rate, FPR	Yes	Accuracy rate for DOS is 99.8% and FPR of 0.17% is achieved
2018	Suad Mohammed, Fadl Mutahe: Intrusion detection model using machine learning algorithm on Big Data environment: In proc. journal of big data. Springer 2018. pp: 1-12. DOI 10.1186/s40537-018-0145-4	Support Vector Machine (SVM)	Chisqselector using SVM classifier for feature reduction	KDD Cup-'99 dataset	AUROC, AUPR, Time	Yes	Area under curve (AUROC) of 99.55%, Area under precision recall curve (AUPR) of 96.24% and training time of 10.79 se
2018	Sivakumar Venkataraman, Rajalakshmi Selvaraj: Optimal and Novel Hybrid Feature Selection Framework for Effective Data Classification: Advances in systems, control and application. Springer 2018. pp: 499–514. DOI 10.1007/978-981-10-4762-6_48	Symmetrical Uncertainty and genetic algorithm (SU-GA) is used as classification algorithm	Symmetric uncertainty is used to obtain best features/genetic algorithm is applied on selected features	UCI dataset.	Accuracy rate	Yes	The average learning accuracy of 83.83% is achieved, and overall evaluation time of 0.23 seconds is achieved on all methods

(continued)

Table 1 (continued)

Year	Authors/research papers	Algorithms	Techniques	Dataset	Evaluation criteria	Feature selection	Results
2018	Neeraj Kumar, Upendra Kumar: Knowledge Computational Intelligence in Network Intrusion Detection Systems. In: Knowledge Computing and Its Applications. Springer 2018. pp: 161-176. DOI 10.1007/978-981-10-6680-1_8	Genetic algorithm	Fuzzy Logic Controller, Neuro- fuzzy inference system, Multi layer perception, Neural Fuzzy genetic for classification of attacks	KDD Cup-'99 dataset	Accuracy rate	Yes	A detection accuracy of 99% is achieved
2019	Akash Saxena, Khushboo Saxena: Hybrid Technique Based on DBSCAN for Selection of Improved Features for Intrusion Detection System: Emerging Trends in Expert Applications and Security. pp: 365377. Springer 2019. DOI 10.1007/978-981-13-2285-3_43	DBSCAN, SMO, K-Means clustering	DBSCAN is used to eliminate noise in data, K-Means is used for grouping of data, and SMO is used for detection of intrusions.	KDD Cup-'99 dataset.	Accuracy rate, Detection rate	Yes	An approx accuracy of 98.1% and approx Detection rate of 70% is achieved

(continued)

Table 1 (continued)

Year	Authors/research papers	Algorithms	Techniques	Dataset	Evaluation criteria	Feature selection	Results
2019	Rajesh Kambattan, Manimegalai Rajkumar: An Effective Intrusion Detection System Using Flawless Feature Selection, Outlier Detection and Classification: Progress in Advanced Computing and Intelligent Engineering. Springer 2019. pp: 203–213. DOI 10.1007/978-981-13-1708-8_19	Intelligent flawless feature selection algorithm (IFLFSA), Entropy Based weighted outlier rejection (EWOD), Intelligent layered classification algorithm	IFLFSA is used as feature selection; EWOD is used to detect outliers in data. Intelligent layered classification algorithm is applied to classify the data	KDD Cup-‘99 dataset	Accuracy rate	Yes	Overall accuracy of 99.45% is achieved
2019	Unal Cavusoglu: A new hybrid approach for intrusion detection using machine learning methods: Applied Intelligence (2019) 49. pp: 2735–2761. Springer 2019. DOI 10.1007/s10489-018-01408-x	Naive bayes, Random forest, J-48, K-Nearest Neighbor (KNN) algorithm	CfsSubsetEval and WrapperSubsetEval as a two feature selection techniques, while Naive bayes, Random forest, J-48, K-Nearest Neighbor (KNN) algorithm is used as a classifier	NSL-KDD Dataset	Accuracy rate, Detection Rate, FP rate, F–measure, MCC, TP rate, Time	Yes	Overall detection ratio of 0.9828%, Overall accuracy rate of 99.86%, Overall TPR of 0.929%, Overall FPR of 0.00042%, Overall F-Measure of 0.9625 %, MCC of 0.955 % and total time to evaluate is 10.625 seconds (Performed on NSL KDD, 25 attributes on all types of attacks)

(continued)

Table 1 (continued)

Year	Authors/research papers	Algorithms	Techniques	Dataset	Evaluation criteria	Feature selection	Results
2019	Pragma Kar, Soumaya Banerjee, Kartick Chandra Mondal, Gautam Mahapatra, Samiran Chattopadhay: A Hybrid Intrusion Detection System for Hierarchical Filtration of Anomalies. In: Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies, vol 106. Springer 2019. pp: 417-426. DOI 10.1007/	ID3, K-Nearest neighbor, Isolation Forest,	ID3 is applied as feature selector, K-nearest neighbor is used to apply a class to unknown data point, Isolation forest is applied to segregate normal data from anomaly	KDD Cup-‘99 dataset, NSL-KDD Dataset	Accuracy rate, Detection rate, False alarm rate	Yes	Performance on NSL-KDD dataset has accuracy of 93.95% has a detection rate of 95.5% and a false alarm rate of 10.34%.The performance with KDD Cup-‘99 dataset has an accuracy of 96.92%, detection rate of 97.20% and false alarm rate of 7.49%
2019	Sushruta Mishra, Chandrakanta Mahanty, Shreela Dash, Brojo Kishore Mishra.: Implementation of BFS-NB Hybrid Model in Intrusion Detection System: Recent Developments in Machine Learning and Data Analytics. Advances in Intelligent Systems and Computing, vol 740. Springer. 2019 pp: 167–175. DOI 10.1007/978-981-13-1280-9_17	Best first search and Naïve bayes, (BFS-NB) algorithm.	Naïve bayes is used as classifier and best search is used as attribute optimization method.	KDD dataset from US Air force	Accuracy, Specificity, Sensitivity	Yes	The classification accuracy of BS-NFB is 92.12%, Sensitivity analysis of 97% and specificity of 97.5% is achieved

5.1 KDD Cup-‘99 Dataset

The KDD Cup-‘99 is an extensively used dataset in the field IDS. KDD Cup-‘99 data is captured from the DARPA 98, IDS evaluation program. The entire KDD Cup-‘99 dataset comprises 4,898,431 single connection records; each contains 41 features classified as normal or attacks. KDD Cup-‘99 dataset classifies the attacks into four types, namely Probe, User to Root, Denial of Service, and Remote to Local [86].

- *Probe.* The intruder gathers the knowledge from the networks or hosts scans the whole networks or hosts that are prone to attacks. An intruder then exploits the system vulnerabilities by looking at the known security breaches so that the whole system is compromised for malicious purposes.
- *Denial of Service (DOS).* This kind of attack results in the unavailability of computing resources to legitimate users. The intruder overloads the resources, by making the computing resources busy, so that genuine users are unable to utilize the full resources of the computer.
- *User to Root (U2R).* The intruder tries to acquire the root access of the system or the administrator privileges by sniffing the passwords. The attacker then looks for the vulnerabilities in the system, to acquire the gain of the administrator authorization.
- *Root to Local (R2L).* The intruder attempts to gain access to the remote machine, which does not have the necessary and legal privilege to access that machine. The attacker then exploits the susceptibility of the remote system, tries gaining access rights to the remote machine. Table 2 describes the categories of attack and attack names.

The KDD Cup-‘99 intrusion datasets mainly consist of three components: The entire KDD Cup-‘99 dataset consists of the examples of normal and attack connections, 10% KDD dataset consists of the data, which aims for training the classifiers, and the KDD test dataset for the testing of classifiers. Table 3 describes the dataset features.

The connection features can be classified into four categories namely:

- *Basic features.* It is accessed from the header of the packet, without inspecting the packet contents, such as (protocol type, duration, flag, service, and the number of bytes that are sent from the source to the destination and contrarily).

Table 2 Four categories of attack

Category of attack	Attack name
Probe	satan, portsweep, nmap, ipsweep
DoS	neptune, back, smurf, land, teardrop, pod
U2R (User to Root)	perl, loadmodule, buffer_overflow, rootkit
R2L (Remote to Local)	warezlient, spy, warezmaster, multihop, guesspasswd, phf, imap, ftp_write

Table 3 KDD Cup-‘99 Dataset Features

Index	Feature name	Description
1	Duration	Connection length
2	protocol_type	Protocol types (TCP, ICMP, and UDP)
3	Service	Mapping of destination port to the service (ftp, telnet, http…)
4	Flag	Connection status
5	src_bytes	Source to destination. (Total number of bytes)
6	dst_bytes	Destination to source. (Total number of bytes)
7	Land	If the address of the source and destination are same from the place, then it is = 1/otherwise 0
8	wrong_fragment	Indicates the number of the wrong fragments
9	Urgent	Indicates the number of the urgent packets
10	Hot	Indicates the number of the hot indicators
11	num_failed_logins	No. of failed attempts at login
12	logged_in	If logged in = 1/if login failed = 0
13	num_compromised	It indicates the compromised states
14	root_shell	With a root account a command interpreter is running then root shell = 1/otherwise its 0
15	su_attempted	su is attempted = 1/otherwise its 0
16	num_root	Number of times the root is accessed
17	num_file_creations	To create new files, the number of operations used
18	num_shells	Number of command interpreters that are active
19	num_access_files	Number of operations to create the file
20	num_outbound_cmds	In the ftp sessions the number of outgoing commands
21	is_host_login	If the login is on the host login file list then, is host login = 1 /otherwise it is 0
22	is_guest_login	If a guest logs into system then, is guest login = 1/ otherwise it is 0
23	Count	At a given interval, the number of connections to the same host as the current connection
24	srv_count	At a given interval, the number of connections to the same service as the current connection
25	serror_rate	SYN errors with proportion of connections
26	srv_serror_rate	SYN errors with proportion of connections
27	rerror_rate	REJ errors with proportion of connections
28	srv_rerror_rate	REJ errors with proportion of connections
29	same_srv_rate	Same services with proportion of connections
30	diff_srv_rate	Different services with proportion of connections
31	srv_diff_host_rate	Different hosts with proportion of connections

(continued)

Table 3 (continued)

Index	Feature name	Description
32	dst_host_count	Same destination having the number of connections
33	dst_host_srv_count	Same destination that apply the same service having the number of connections
34	dst_host_same_srv_rate	Same destination that apply the same service having the number of connections
35	dst_host_diff_srv_rate	Different services having same host, has the number of connections
36	dst_host_same_src_port_rate	System having same source port has the number of connections
37	dst_host_srv_diff_host_rate	Same service incoming from different hosts having the number of connections
38	dst_host_serror_rate	Host having a S0 error has the number of connections
39	dst_host_srv_serror_rate	The host and the specified service having S0 error have the number of connections
40	dst_host_rerror_rate	The host having an RST error has the number of connections
41	dst_host_srv_rerror_rate	The host and the specified service having an RST error have the number of connections

- *Content features.* It is decided by investigating the details of the TCP packet (number of failed attempts to login the computer).
- *Traffic features.* The features are calculated in proportionate to window intervals. It is further divided into the same service features and same host features. The same service features it analysis the connections in last two seconds having the same service as a present connection. The same host features, it analyses the connections in the last two seconds of the host having the same destination as its present connection and then computes the stats of the protocol behavior.
- *Time features.* The same service features and same host features and are also called time-based features. The time features decide the length of time from the source IP address to the destination IP address.

5.2 NSL-KDD Dataset

KDD Cup-'99 datasets are having several redundant records and duplicate records. The redundant record is having an overall 78% and duplicate records have an overall 75%. This redundancy and duplicity prevent it from classifying the other records [87].

A contemporary NSL-KDD dataset was proposed [88] that does not include the redundant and duplicate records in training and testing data [89], which helped in overcoming the redundancy and duplicity issue. There is a total 37 attacks in

Table 4 NSL-KDD: Number of instances in training and test data

Attack Classes	Total number of instances in the training set	Total number of instances in the test set
DoS	land (18), teardrop (892), pod (201), neptune (41,214), smurf (2,646), back (956)	neptune (4,657), pod (41), smurf (665), apache2 (737), worm (2), mailbomb (293), processtable (685), udpstorm (2), teardrop (12), land (7), back (359)
Probe	portsweep (2,931), ipsweep (3,599), nmap (1,493), satan (3,633),	satan (735), ipsweep (141), nmap(73), portsweep (157), mscan (996), saint (319)
R2L	guess_passwd (53), ftp_write (8), imap (11), phf (4), multihop (7), warezmaster (20), warezclient(890), spy(2)	ftp_write (3), phf (2), warezmaster (944), snmpguess (331), httptunnel (133), named (17), guess_passwd (1,231), imap (1), multihop (18), xlock (9), snmpgetattack (178), xsnoop (4), sendmail (14)
U2R	buffer_overflow (30), rootkit (10), perl (3), loadmodule (9)	rootkit (13), loadmodule (2), perl (2), sqlattack (2), xterm (13), ps (15), buffer_overflow(20)
Total	58,630	12,833

the testing dataset out of which 21 different attacks are of training dataset and the remaining attacks are available only for testing the data.

The attack classes are categorized into DoS, Probe, R2L, and U2R categories. In the training data, the normal traffic consists of 67,343 instances having an overall of 125,973 instances. In test data the normal traffic consists of 9711 instances, having an overall of 22,544 instances. Table 4 depicts the total stats of instances in training and test data.

5.3 Kyoto 2006 + Dataset

The Kyoto 2006 + dataset consists of original data that is captured from network traffic data between the years 2006 and 2009. The new version consists of further data collected from the year 2006 to 2015. This dataset is taken by applying the email server, honey pots, dark net sensors, and WebCrawler [90, 91]. The authors have employed various types of darknet, honeypots on five of the networks in indoors and outdoors of the Kyoto University and from the honeypots, the traffic data is possessed.

There are 50,033,015 normal session, 43,043,225 attack sessions and 425,719 sessions relevant to unknown attack. The author's extracted 14 statistical features from the honeypots of 41 features from the KDD Cup-'99 dataset. Table 5 shows the statistical features of the Kyoto 2006 + dataset, which is taken from the dataset of KDD Cup-'99.

Besides, the authors also obtained ten features extra, which facilitate the authors to further examine, what kind of attacks appear on computer networks. The redundant

Table 5 Statistical features in the Kyoto 2006 + dataset derived from the KDD Cup-‘99 dataset

Index	Feature name	Description
1	Duration	Connection length (in number of seconds)
2	Service	Connection service types such as (http, telnet, etc.)
3	Source bytes	The source IP address that sends the number of data bytes
4	Destination bytes	The destination IP address that sends the number of data bytes
5	Count	For past two seconds, the number of connections having same for Source and destination IP address for current connection
6	Same_srv_rate	Identical service having the count feature with proportion of connections
7	Serror_rate	‘SYN’ errors having the count feature with proportion of connections
8	Srv_serror_rate	‘SYN’ errors having the service count with proportion of connections
9	Dst_host_count	Between the past 100 connections whose destination IP address is same to the current connection, the number of connections having source IP address is same to that of the current connection
10	Dst_host_srv_count	Between the past 100 connections whose destination IP address is same to that of the current connection, the number of connections having service type is same to that of the current connection
11	Dst_host_same_src_port_rate	Source port having the same of the current connection in Dst_host_count feature with proportion of connections
12	Dst_host_serror_rate	‘SYN’ errors having Dst_host_count feature with proportion of connections
13	Dst_host_srv_serror_rate	‘SYN’ errors having Dst_host_srv_count feature with proportion of connections
14	Flag	The connections state at the time of which connection was written

and insignificant features, content features are ignored as they are unsuitable and time consuming for NIDS to obtain without domain knowledge. Table 6 depicts the additional features of the Kyoto 2006 + dataset.

5.4 *Performance Metrics*

This section describes various performance metrics used to validate the results in IDS. Those metrics indicate all the measurements that are used by researchers to

Table 6 Additional features in Kyoto 2006 + dataset

Index	Feature name	Description
1	IDS_detection	It indicates that if IDS is started an alarm for the connection. ‘0’ indicates that alarms are not triggered and an Arabic numeral indicates various kinds of alarms. Parenthesis implies the same amount of alarm
2	Malware_detection	It specifies that if malware was noticed at the connection. ‘0’ indicates that malware was not noticed, and string implies the identical malware noticed at the connection. Parenthesis implies the amount of similar malware
3	Ashula_detection	It indicates if exploit codes and shellcode were applied in connection. ‘0’ indicates no exploit code and shellcode were observed, and an Arabic numeral implies the distinctive types of the exploit codes or shellcodes. Parentheses indicate the same figure of exploit code or shellcodes
4	Label	It means that, there was attack in the session or not ‘1’ indicates as normal. ‘-1’ means familiar attack was observed in the session, and ‘-2’ indicates unidentified attack was observed in the session
5	Source_IP_Address	It indicates that source IP address is utilized in session
6	Source_Port_Number	It implies that the source port number is utilized in the session
7	Destination_IP_Address	It implies that the destination IP address is utilized in the session
8	Destination_Port_Number	It implies that the destination port number is utilized in the session
9	Start_Time	It implies that at which moment the session was started
10	Duration	It indicates the total time duration when the session is set

prove the results obtained in any of: True positive (TP), True negative (TN), False negative (FN), False positive (FP) [92].

5.4.1 Confusion Matrix

It is known as error metrics, allowing the correlation between predicted and actual classes. It is important for measuring the accuracy, recall, precision, specificity, and ROC, AUC curve. It helps in the visualisation performance of the algorithms and it is often applied to depict the performance of the classifier on the testing dataset.

Class	Predicted positive class	Predicted negative class
Actual positive class	FN	TP
Actual negative class	TN	FP

5.4.2 Accuracy

The ratio of true predictions, precisely classified instances, calculated by:

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{(\text{TN} + \text{TP} + \text{FN} + \text{FP})}$$

5.4.3 Error Rate

The ratio of all the predictions made, which are maliciously classified: It is calculated by:

$$\text{ER} = 1 - \text{Acc.}$$

5.4.4 True Positive Rate

It is termed as the intrusions that are accurately classified as an attack by the IDS. It is also called recall, sensitivity, or detection rate. It is calculated by:

$$\text{TPR} = \frac{\text{TP}}{\text{FN} + \text{TP}}$$

5.4.5 False Positive Rate

It is the regular patterns that are wrongly identified as an attack. It is also called a false alarm rate is given by:

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

5.4.6 True Negative Rate

It is the regular patterns that are correctly identified as normal. It is also known as specificity given by:

$$\text{TNR} = 1 - \text{FPR.}$$

5.4.7 False Negative Rate

It is termed as the intrusions that are wrongly identified as normal. It is calculated by:

$$\text{FNR} = 1 - \text{TPR}.$$

5.4.8 Precision

It is the amount of behavior precisely classified as an attack. It is calculated by:

$$\text{Precision} = \frac{\text{TP}}{(\text{FP} + \text{TP})}$$

5.4.9 F-measure

It is the measurement of accuracy tests. It is described as the weighted harmonic mean of the precision and recall and it also called as f-value or f-score:

$$\text{FM} = 2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

5.4.10 Matthews's Correlation Coefficient

It is used only in the binary intrusion detection system in which it computes the observed and predicated binary classification. It is calculated by:

$$\text{MCC} = \frac{(\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{TP} + \text{FN})(\text{TP} + \text{FP})(\text{TN} + \text{FN})(\text{TN} + \text{FP})}}$$

The detection achievement of IDS is classified using the area under curve (AUC) and receiver operating characteristic (ROC). ROC reveals the changes in detection ratio with contrast in the internal verge to produce a low or high false positive rate. If AUC values are larger, then the classifier performance will be better. Another important metric that are considered in the evaluation of IDS is CPU consumption, its throughput, and power consumption which may run on different hardware configuration on high speed networks with or without the limited number of the resources.

6 Research Opportunities

The inception of Intrusion Detection Systems has started in the year 1980, numerous literature studies have presented on this topic to date, there exists a vide varieties of research opportunities in this field. Some of the research opportunities are listed below.

- *Trusted IDS.* There is a demand to develop a dependable intrusion detection systems that accurately classifies the novel intrusions, such a degree that it can be efficiently trusted without or little human interventions as possible. The robustness of IDS still needs further investigation to new evasion techniques.
- *Nature of Attacks.* The knowledge information about the nature of new intrusion attacks needs to be continuously studied and updated frequently, as the attackers are motivated enough and try different ways to intrude the IDS systems. Updating the knowledge of new attacks helps in preventing the new attacks that penetrate the IDS systems.
- *The efficiency of IDS.* One of the major concerns about IDS that exists today is its run time efficiency. There is a concern about IDS, that it consumes more resources results in degrading the IDS performance. The need of the hour is to develop efficient IDS that is computationally inexpensive and consumes fewer resources without affecting the performance factors of IDS.
- *Datasets.* NSL-KDD, Kyoto 2006 +, and KDD Cup-'99 are considered as the benchmark datasets in the field of IDS, still one needs to evaluate the performance of IDS on other datasets such as CAIDA, ISCX, ADFA-WD, ADFA-LD, CICIDS, BoT-IoT, etc. Evaluating the performance of IDS in different datasets helps us to know the actual performance of IDS on different environments.

7 Conclusion

In this study, an analysis of various intrusion detection systems that applies the various machine learning techniques was observed. Different hybrid machine learning-based intrusion detection techniques, proposed by various researchers have been analyzed. The outcome of the results and observing various parameters such as accuracy, detection rate, and false positive alarm rates are the most important contributions of this study. The study of the KDD Cup-'99, NSL-KDD, and Kyoto 2006 + dataset, with its different parameters, are discussed in this article. Finally, observation of the various performance metrics used for evaluation of results in intrusion detection systems is analyzed. By observing various parameters in this article it is concluded that "A Network Hybrid Model for Intrusion Detection System" was an efficient one.

Acknowledgements Sincere thanks to Dr. Philippe Fournier-Viger, the founder of the SPMF data mining library for providing us his deep insights into our work.

References

1. https://cybersecurityventures.com/research/
2. Liao, H.-J., Richard, C.-H.: Intrusion detection system a comprehensive review. J. Netw. Appl. 16–24. Elsevier
3. Mohammed, M., Khan, M.B.: Machine Learning Algorithms and Applications. CRC Press Taylor and Francis Group
4. Tsai, C.-F., Hsu, Y.-F.: Intrusion detection by machine learning, a review. Expert Systems with Applications, pp. 11994–12000. Elsevier
5. Kang, M., Jameson, N.J.: Machine learning Fundamentals Prognostics and health management in electronics. Fundamentals. Machine Learning, and Internet of Things. Willey Online Library
6. Quinlan, J.R.: Machine Learning, vol. 1, no. 1
7. Quinlan, J.R.: C4.5: Programs for Machine Learning, vol. 16, pp. 235–240. Morgan Kaufmann Publishers, Inc.
8. Littlestone, N., Warmuth, M.K.: The weighted majority algorithm. Inf. Comput. **108**(2), 212–261
9. McCulloch, W., Pitts, W.: A logical calculus of ideas immanent in nervous activity. Bull. Math. Biophys. **5**(4), 115–133
10. Freund, Y., Schapire, R. E.: Large margin classification using the perceptron algorithm. Mach. Learn. **37**(3), 277–296
11. Pearl, J.: Bayesian networks. A model of self-activated memory for evidential reasoning. In: Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine, CA, pp. 329–334. Accessed 01 May 2009
12. Rish, I.: An empirical study of the Naive Bayes classifier. IJCAI Workshop on Empirical Methods in AI
13. Altman, N. S.: An introduction to kernel and nearest-neighbor nonparametric regression (PDF). The American Statistician, 46 (3), pp. 175–185
14. Yuan, G.-X., Ho, C.-H.: Recent advances of large-scale linear classification. Proceedings of the IEEE, pp. 2584–2603
15. Cortes, C., Vapnik, V.N.: Support-vector networks. Mach. Learn. **20**(3), 273–297
16. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, pp. 61–74. Advances in Large Margin Classifiers, MIT Press
17. MacQueen, J. B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. University of California Press
18. Kaufman, L, Rousseeuw, P.J.: Clustering by means of Medoids. In: Statistical Data Analysis Based on the Norm and Related Methods, pp. 405–416. North-Holland
19. Duda, R.O, Hart, P.E.: Pattern Classification and Scene Analysis. Wiley
20. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. 1–38
21. Baum, L.E., Petrie, T.: Statistical inference for probabilistic functions of finite State Markov Chains. Ann. Math. Stat. 1554–1563
22. Kohonen, T.: The self-organizing map. In: Proceedings of IEEE, pp. 1464–1480
23. Ng, A., Jordan, M.: On discriminative versus generative classifiers. A comparison of logistic regression and naive bayes. Adv. Neural Inf. Process Syst.
24. Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts. In: Proceedings of the 18th International Conference on Machine Learning
25. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceeding of the 16th International Conference on Machine Learning (ICML), pp. 200–209. Morgan Kaufmann, San Francisco (1999)
26. Chapelle, O., Schölkopf, B., Zien, A.: Semi-supervised Learning. MIT Press
27. Zhu, X.: Semi-supervised Learning Literature Survey. University of Wisconsin, Madison
28. Nigam, K., Mccallum, A.K.: Text classification from labeled and unlabeled documents using EM. Machine Learning, vol. 39, pp. 103–134. Springer

29. Szummer, M., Jaakkola, T.: Partially labeled classification with Markov random walks. Advances in Neural Information Processing Systems
30. Yu, N.: Domain adaptation for opinion classification, a self training approach. J. Inf. Sci. Theory Pract
31. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: COLT: Proceedings of the Workshop on Computational Learning Theory
32. Barto, A.G, Sutton, R.S., Anderson, C.W.: Neuron like adaptive element that can solve difficult learning control problems. IEEE Trans. Syst. Man Cybern. 834–846
33. Sutton, R.S.: Learning to predict by the method of temporal differences. Mach. Learn. 9–44
34. Watkins, C.J, C.H, Dayan, P.: Q learning. Mach. Learn. 279–292
35. Kumar, P.R, Variya, P.P.: Stochastic System: Estimation, Identification and adaptive control. Prentice Hall, Englewood Cliffs, NJ
36. Sutton, R.S.: Integrated architectures for learning and planning and reacting based on the approximating dynamic programming. In: Proceedings on Seventh International Conference on Machine Learning, Austin, T.X Morgan Kaufmann
37. Moore, A.W., Atkeson, C.G.: Prioritized sweeping. Reinforcement learning with less data and less time. Mach. Learn.
38. Barto, A.G, Bradke, S.J, Singh S.P.: Learning to act using real time dynamic programming. Artif. Intell. 81–138
39. Lior, R.: Ensemble learning. Pattern classification using ensemble methods. Ser. Mach. Perception Artif. Intell. **85**
40. Schapire, R.E.: The strength of weak learnability. Mach. Learn. 197–227
41. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996d)
42. Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. Chapman & Hall, New York, NY (1993)
43. Smyth, P., Wolpert, D.: Stacked density estimation. In: Jordan, M.I, Kearns, M.J., Solla, S.A. (eds.), Advances in Neural Information ProcessingSystems, vol. 10, pp. 668–674. MIT Press, Cambridge, MA (1998)
44. Wolpert, D.H.: Stacked generalization. Neural Netw. **5**(2), 241–260 (1992)
45. Kohavi, R., John, G.: Wrappers for feature subset selection. Artif. Intell. Spec. Issue Relev. 273–324
46. Langley, P.: Selection of relevant features in machine learning. In: Proceedings of the AAAI Fall Symposium on Relevance. AAAI Press
47. Miller, A. J.: Subset Selection in Regression. Chapman and Hall, New York
48. Kohavi, R.: Wrappers for Performance Enhancement and Oblivious Decision Graphs. Ph.D. thesis, Stanford University
49. Langley, P.: Selection of relevant features in machine learning. In: Proceedings of the AAAI Fall Symposium on Relevance. AAAI Press
50. Pearson, K.: On Lines and Planes of Closest Fit to Systems of Points in Space, pp. 559–572, Philosophical Magazines
51. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees, Monterey, CA, Wadsworth & Brooks/Cole Advanced books & Software
52. Anderson, J.P.: Computer society threat monitoring and surveillance. Fort Washington, PA Computer Security Research Centre
53. Halme, L., R.: AIN'T misbehaving-A taxonomy of anti-intrusion techniques. Comput. Secur. **14**(7), 606 (1995)
54. Nisioti, A., Mylonas, A.: From intrusion detection to attacker attribution. Comprehensive survey of unsupervised methods. IEEE Commun. Surv. Tutor. **20**, 3369–3388
55. Sebyala, AA, Olukemi T, Sacks L.: Active platform security through intrusion detection using Naive Bayesian network for anomaly detection. In: The London Communications Symposium. Citeseer, London
56. Fan, W, Miller, M, Stolfo, S, Lee, W, Chan P.: Using artificial anomalies to detect unknown and known network intrusions. Knowl. Inf. Syst. **6**(5), 507–527
57. Vapnik, V.: The Nature of Statistical Learning Theory, 2nd edn. Springer, New York

58. Williams, G., Baxter, R., He, H., Hawkins, S., Gu, L.: A comparative study of ANN for outlier detection in data mining. In: Proceedings of IEEE International Conference on Data Mining (ICDM'02), Maebashi City, Japan, pp. 709–712. IEEE
59. Liao, Y, Vemuri V,R.: Use of K-nearest neighbor classifier for intrusion detection. Comput. Secur. **21**(5), 439–448
60. Gruschke, B.: Integrated event management. Event correlation using dependency graphs. In: Proc. of the 9th IFIP/IEEE International Workshop on Distributed Systems, pp. 130–141. Operations & Management (DSOM 98), Newark, DE, USA
61. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection. A survey ACM Computure Surveys, vol. 41, no. 3, pp. 1–72 (2009)
62. Tylman, W.: Anomaly-based intrusion detection using bayesian networks. In: Third International Conference on Dependability of Computer Systems Szklarska, Poreba, Poland, pp. 211–218, 26–28 June 2008
63. Botha, M, Von, Solms, R.: Utilising fuzzy logic and trend analysis for effective intrusion detection. Comput. Secur. **22**(5), 423–434
64. Cha, B.R., Vaidya, B., Han, S.: Anomaly intrusion detection for system calls using the soundex algorithm and neural networks. In: 10th IEEE Symposium on Computers and Communications (ISCC'05), Cartagena, Spain, pp. 427–433. IEEE
65. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection. Detecting intrusions in unlabeled data. In: Proceedings of the Conference on Applications of Data Mining in Computer Security, pp. 78–100. Kluwer Academics
66. Fangfei, W., Qingshan, J., Lifei, C., Zhiling, H.: Clustering ensemble based on the fuzzy KNN algorithm. In: Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'07), Qingdao, July 30, 2007–Aug 1, 2007, vol 3, pp. 1001–1006 (2007)
67. Idris, NB., Shanmugam, B.: Artificial intelligence techniques applied to intrusion detection. In: IEEE India Conference Indicon (INDICON'05), Chennai, India, pp. 52–55, 11–13 Dec 2005
68. Mukkamala, S., Sung, AH., Abraham, A.: Intrusion detection using an ensemble of intelligent paradigms. J. Netw. Comput. Appl. **28**(2), 167–182
69. Agrawal, S., Agrawal, J.: Survey on anomaly detection using data mining techniques. In: Proceeding with 19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, vol. 60, pp. 708–713. Elsevier (2015)
70. Acharya, N., Singh, S.: An IWD-based feature selection method for intrusion detection system. Soft Computing, pp. 4407–4416, Springer (2017). https://doi.org/10.1007/s00500-017-2635-2
71. Shah-Hosseini, H.: Optimization with the nature-inspired intelligent water drops algorithm. Dos Santos, W.P. (ed.) Evolutionary computation. I-Tech, Vienna, pp. 298–320. ISBN 978–953–307–008–7
72. Malik, A.J., Khan, F.A.: A hybrid technique using binary particle swarm optimization and decision tree pruning for network intrusion detection. Cluster Computing, pp. 667–680, Springer (2017). https://doi.org/10.1007/s10586-017-0971-8
73. Saleh1, A.I., Talaat1, F.M.: A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers. Artificial Intelligence Review, pp. 403–443. Springer (2017). https://doi.org/10.1007/s10462-017-9567-1
74. Dash, T.: A study on intrusion detection using neural networks trained with evolutionary algorithms. Soft Computing, pp. 2687–2700. Springer (2017). https://doi.org/10.1007/s00500-015-1967-z
75. Yao, H., Wang: An Intrusion Detection Framework Based on Hybrid Multi-Level Data Mining, International Journal of Parallel Programming, pp. 1–19. Springer (2017). https://doi.org/10.1007/s10766-017-0537-7
76. Alauthaman, M., Aslam, N.: A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks. Neural Computing & Applications, pp. 991–1004. Springer (2018). https://doi.org/10.1007/s00521-016-2564-5.

77. Ijaz, S., Hashmi, F.A.: Vector based genetic algorithm to optimize predictive analysis in network security. Applied intelligence, vol. 48, issue 5, pp. 1086–1096. Springer (2018). https://doi.org/10.1007/s10489-017-1026-9
78. Mohammed, S., Mutaheb, F.: Intrusion detection model using machine learning algorithm on Big Data environment, proceedings. J. Big Data 1–12. Springer (2018). https://doi.org/10.1186/s40537-018-0145-4
79. Venkataraman, S., Selvaraj, R.: Optimal and Novel Hybrid Feature Selection Framework for Effective Data Classification, Proceedings with: Advances in Systems, Control and Application, pp. 499–514. Springer (2018). https://doi.org/10.1007/978-981-10-4762-6_48
80. Kumar, N., Kumar, U.: Knowledge Computational Intelligence in Network Intrusion Detection Systems, Knowledge Computing and Its Applications, pp.161–176. Springer (2018). https://doi.org/10.1007/978-981-10-6680-1_8
81. Saxena, A., Saxena, K.: Hybrid Technique Based on DBSCAN for Selection of Improved Features for Intrusion Detection System, Emerging Trends in Expert Applications and Security, pp. 365–377. Springer (2019). https://doi.org/10.1007/978-981-13-2285-3_43
82. Kambattan, R., Rajkumar, M.: An effective intrusion detection system using flawless feature selection, outlier detection and classification. Progress in Advanced Computing and Intelligent Engineering, pp. 203–213. Springer (2019). https://doi.org/10.1007/978-981-13-1708-8_19
83. Cavusoglu, U.: A new hybrid approach for intrusion detection using machine learning methods. Applied Intelligence, pp. 2735–2761. Springer 2019. https://doi.org/10.1007/s10489-018-01408-x
84. Kar, P., Banerjee, S., Mondal, K.C., Mahapatra G., Chattopadhyay S.: A hybrid intrusion detection system for hierarchical filtration of anomalies. Information and Communication Technology for Intelligent Systems, Smart Innovation Systems and Technologies, vol. 106, pp. 417–426. Springer (2019). https://doi.org/10.1007/978-981-13-1742-2_41
85. Mishra, S., Mahanty, C., Dash, S., Mishra, B.K.: Implementation of BFS-NB hybrid model in intrusion detection system, recent developments in machine learning and data analytics. Advances in Intelligent Systems and Computing, vol. 740, pp. 167–175. Springer (2019). https://doi.org/10.1007/978-981-13-1280-9_17
86. Al-Dhafian, B., Ahmad, I, Al-Ghamid, A.: An Overview of the Current Classification Techniques, International Conference on Security and Management, Las Vegas, USA, pp. 82–88, July 27–30
87. Revathi, S., Malathi, A.: A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection. Int. J. Eng. Res. Technol. **2**(12), 1848–1853
88. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, Ali, A.: A detailed analysis of the KDD Cup-'99 data set. In: Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications, Ottwa, Canada, July 8–10
89. Kavitha, P., Usha, M.: Anomaly based intrusion detection in WLAN using discrimination algorithm combined with Naïve Bayesian classifier. J. Theor. Appl. Inf. Technol. **62**(1), 77–84
90. Singh, R., Kumar, H., Singla, R.K.: An intrusion detection system using network traffic profiling and online sequential extreme learning machine. Expert Syst. Appl. **42**(22), 8609–8624
91. Song, J., Takakura, H., Okabe, Y., Eto, M., Inoue, D., Nakao, K.: Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation. In: Proceedings of the 1st Workshop on Building Anal. Datasets and Gathering Experience Returns for Security, Salzburg, pp. 29–36 (2006)
92. Hindy, H., Brosset, D.: A Taxonomy and Survey of Intrusion Detection System Design Techniques, Network Threats and Datasets, pp. 1–35, 9, June 2018. arXiv. 1806.03517v1 [cs.CR]
93. Fournier-Viger, P., Lin, C.W., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., Lam, H. T.: The SPMF open-source data mining library version 2. In: Proceedings of the 19th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2016) Part III, pp. 36–40. Springer LNCS 9853 (2016). https://www.philippe-fournier-viger.com/spmf/