

The Use of Multi-Objective Genetic Algorithm Based Approach to Create Ensemble of ANN for Intrusion Detection

Gulshan Kumar¹, Krishan Kumar²

¹Department of Computer Application, Shahed Bhagat Singh State Technical Campus, Punjab, India

²Department of Computer Science & Engineering, Punjab Institute of Technology, Punjab, India

Email: gulshanahuja@gmail.com, k.salujapitk@gmail.com

Received June 8, 2012; revised July 18, 2012; accepted July 31, 2012

ABSTRACT

Due to our increased dependence on Internet and growing number of intrusion incidents, building effective intrusion detection systems are essential for protecting Internet resources and yet it is a great challenge. In literature, many researchers utilized Artificial Neural Networks (ANN) in supervised learning based intrusion detection successfully. Here, ANN maps the network traffic into predefined classes *i.e.* normal or specific attack type based upon training from label dataset. However, for ANN-based IDS, detection rate (DR) and false positive rate (FPR) are still needed to be improved. In this study, we propose an ensemble approach, called MANNE, for ANN-based IDS that evolves ANNs by Multi-Objective Genetic Algorithm to solve the problem. It helps IDS to achieve high DR, less FPR and in turn high intrusion detection capability. The procedure of MANNE is as follows: firstly, a Pareto front consisting of a set of non-dominated ANN solutions is created using MOGA, which formulates the base classifiers. Subsequently, based upon this pool of non-dominated ANN solutions as base classifiers, another Pareto front consisting of a set of non-dominated ensembles is created which exhibits classification tradeoffs. Finally, prediction aggregation is done to get final ensemble prediction from predictions of base classifiers. Experimental results on the KDD CUP 1999 dataset show that our proposed ensemble approach, MANNE, outperforms ANN trained by Back Propagation and its ensembles using bagging & boosting methods in terms of defined performance metrics. We also compared our approach with other well-known methods such as decision tree and its ensembles using bagging & boosting methods.

Keywords: Ensemble Classifiers; Intrusion Detection System; Intrusion Detection; Multi-Objective Genetic Algorithm

1. Introduction

With coming age of Internet and dependence of business applications on it, network security has become key foundation. Information access through Internet provides various ways of attacking the computer system. More and more organizations have become vulnerable to Internet attacks/intrusions. An intrusion or attack can be defined as “any set of actions that attempt to compromise the security objectives”. The important security objectives include Availability, Integrity, Confidentiality, Accountability and Assurance [1]. Intrusion detection attempts to detect computer attacks/intrusions by analyzing audit data of the network [2,3]. Intrusion detection system (IDS) is one of the most important components among six anti-intrusion systems namely prevention, preemption, deterrence, deflection, detection, and countermeasures [4]. Detection rate (DR) and False Positive rate (FPR) are two key indicators to evaluate the capability of IDSs. Many efforts have done to improve DR and FPR of

IDS [5]. In the beginning, the research focus was on rule based and statistical IDS. But, with large dataset, the results of these IDS become un-satisfactory. Thus, a lot of AI based intrusion detection techniques have been introduced to solve the problem [1,5,7]. Among these techniques, Artificial Neural Networks (ANN) is one of the most widely and robust techniques used to solve complex problems. Many researchers have successfully utilized ANN for IDS [3,8-11] due to its advantages like: 1) High tolerance to noisy data; 2) Ability to classify untrained patterns; 3) Well-suited for continuous-valued inputs and outputs; 4) Successful on a wide array of real-world data; 5) Algorithms are inherently parallel [12]. But, major limitations of ANN based IDSs are: 1) Lower value of DR and high value of FPR, especially for monitoring attack classes like U2R and R2L classes; 2) Long training time; 3) Require a number of parameters typically best determined empirically, e.g., the network topology and weights; 4) Poor interpretability: difficult to

interpret the symbolic meaning behind the learned weights and of hidden units in the network [12]. The major cause of low results for minority attack classes is that the distribution of different types of attacks is imbalanced [13]. This imbalance results difficulty in learning the features of minority attack classes by ANN. The minority attack classes are also equally important as majority attack classes. The minority attack may cause serious damage if succeeded [9]. For example, if the R2L attacks succeeded, the attacker can get the authority of root user remotely. The attacker can do the whole thing he likes to the targeted computer systems or network device. Although prior research has proposed some approaches for effective intrusion detection systems. But, for improvement of DR and FPR, these approaches still need to be researched to become IDS more effective [5,14]. One limitation with majority of approaches is the lack of control over classification tradeoff the solution they obtains. This issue is identified as general issue while creating classifiers. Striving to create a single best classifier that obtains the highest accuracy may give unfruitful results when used under different scenarios [15].

To solve these problems, we propose an ensemble approach for ANN-based IDS evolved by Multi-objective genetic algorithm (NSGA II) [16], called MANNE (Multi-objective genetic algorithm based Artificial Neural Network Ensemble) to improve the DR and FPR of intrusion detection. The procedure for MANNE approach has following phases.

Phase 1: it generates an optimal Pareto front consisting of a set of non-dominated ANN solutions using MOGA, which formulate the base classifiers from training dataset.

Phase 2: it generates another optimal Pareto front consisting of a set of non-dominated ensembles based upon pool of non-dominated ANN solutions as base classifiers (output of phase 1) which exhibit classification tradeoffs.

Phase 3: Prediction aggregation—It gets the final prediction of ensemble from predictions of base classifiers.

In phase 1, we utilized multi objective Genetic algorithm (MOGA) based approach to evolve weights of ANN as base classifiers. It yields an improved Pareto front of solutions. Here, DR of each attack class is treated as separate objective in multi objective approach. Further in phase 2, base classifiers are selected to create ensemble of ANNs to optimize the objective functions. Phase 3 aggregates the predictions of base classifiers to get final prediction of ensemble. To illustrate the applicability and capability of the new approach for intrusion detection, the results of experiments on KDD CUP 1999 dataset [17] are computed. The results demonstrated better performance of MANNE in comparison to widely used intrusion detection approaches. We selected ANN trained using Back Propagation method and most widely

used methods for creating ensembles namely Bagging [18], Boosting [19] (ANN as base classifier) to compare MANNE. The results are also compared to other well-known method such as decision tree, and its ensembles based upon bagging & Boosting in terms of defined performance evaluation metrics.

Article Overview: The rest of the paper is organized as follows. Section 2 presents the literature analysis. Section 3 presents the description of framework and explains the working of MANNE. Section 4 gives the implementation details of MANNE. It highlights the basics of techniques *i.e.* ANN and MOGA used in the experiments. Section 5 gives details of dataset used to evaluate, evaluation criteria adopted and discuss the results of the experiments. Finally, concluding remarks and future directions are highlighted in Section 6.

2. Literature Analysis

IDS can be categorized into number of classes based on different criteria [1]. One way to classify IDS is based upon method to process the audit data. Using this criteria, the IDSs can be categorized into two classes namely signature based systems and anomaly based systems [1-3]. Signature based IDS identifies the intrusions by matching the patterns to known attack scenarios/signatures. Whereas, anomaly based IDS works to search for malicious behavior that deviates from established normal patterns.

Another criterion to classify IDS is source of audit data. Based upon this criteria, IDSs can be categorized into two classes namely host based IDS and network based IDS [1]. Host based IDS collects the data from a host to be protected. They collect the data generally from system calls, operating system log files, NT events log file, CPU utilization, application log files, etc. Advantage of Host based IDS is that they are operating system dependent & are very efficient to detect attacks like buffer overflow. These systems become inefficient in case of encrypted data and switched network. Whereas, network based IDS collects the data from network directly in form of packets. These IDSs are operating system independent and easy to deploy to various systems. In this paper, our interest is in network based intrusion detection.

In order to detect the intrusions, various approaches have been developed and proposed over the last decade [1,5,9,20,21]. In the beginning, the research focus was on rule based IDS and statistical IDS. Rule based IDS reported high classification results for known attacks whose signatures are available in database. But, these systems fail to detect novel attacks. The systems require continuous updating of signature database which is very complex and laborious task. Statistical based IDS apply different statistical techniques to design a model to establish threshold values. These IDS requires collection of

enough data to build complicated model which is computationally expensive for complicated network traffic [9]. In order to meet deficiency of these methods, a lot of AI based intrusion detection techniques have been introduced [1,6,7,21]. Among these techniques, Artificial Neural Networks (ANN) is one of the most widely and robust techniques used to solve complex problems [9]. Many researchers have successfully utilized ANN for IDS [3,8-11] due to its advantages like: 1) High tolerance to noisy data; 2) Ability to classify untrained patterns; 3) Well-suited for continuous-valued inputs and outputs; 4) Successful on a wide array of real-world data; and 5) Algorithms are inherently parallel [1,12]. ANN can be used in different modes. The modes are: 1) Supervised ANN based IDS; 2) Unsupervised ANN based IDS; and 3) Hybrid ANN based IDS [9]. In supervised ANN based IDS, labeled training data is required to build its model [8,22,23]. Whereas it is not required in case of unsupervised ANN based IDS. In case of hybrid ANN based IDS, there is a combination of supervised ANN and unsupervised ANN, or a combination of ANN with other data mining techniques to detect intrusions [24,25].

Many researchers used ANN as single classifier as well as ensemble classifier for intrusion detection. They reported improvement of detection results by using ensemble of ANN over single ANN. Many researchers utilized ANN in supervised mode for intrusion detection. The main focus was on multi-layer feed-forward (MLFF) neural networks and recurrent neural networks [22,23]. Ryan *et al.* (1998) used MLFF neural networks for anomaly detection based on user behaviors based upon UNIX commands [8]. The approach was to build a profile for each user based upon the commands executed. The 100 most commonly used commands were selected, thus, giving a 100-dimensional vector of command frequency intervals for each user, which is used as input to the MLP. On average, the system correctly identifies the normal users 93% of the time and 63% of the randomly generated vectors were classified as intrusive. But, practically, the number of training set is very large and distribution is imbalanced leading ANN reaches some local minima and thus lower value of DR.

Ghosh and Schwartzbard (1999) utilized MLP for application based anomaly and misuse detection [26]. Along with MLP, they applied a leaky bucket algorithm to facilitate some form of a memory mechanism since it is necessary to classify sequences of events. One network was trained for each process/program. They used DARPA98 data to evaluate their approach. They reported best results of 77.3% true positives and 3.6% false positives for the anomaly detection. For misuse based detection, the results were nearly 20% false positives at approximately the same true positive rate. They justified the results due to limited amount of intrusive data used for

learning.

Han and Cho (2005) proposed an intrusion detection technique based on evolutionary neural networks in order to determine the structure and weights of the call sequences [24]. Chen, Abraham and Yang (2007) proposed hybrid flexible neural-tree-based IDS based on flexible neural tree, evolutionary algorithm and particle swarm optimization (PSO) [27]. Empirical results indicated that the proposed method is efficient.

Engen *et al.* (2009) presented Multi Objective Evolution of Artificial Neural Network Ensembles (MABLE) to achieve high DR and FPR for minority attack classes [11]. They used multi objective approach to yield Pareto front of ANN solutions. The non-dominated front of solutions is further utilized to create the ensembles of ANN. The approach was evaluated using KDD cup 1999 dataset.

Wang *et al.* (2010) presented an approach called FC-ANN, based on ANN and fuzzy clustering [9]. They claimed to achieve higher detection rate, less false positive rate and stronger stability especially for low frequent attacks. The proposed approach generated different training sets by using fuzzy clustering to train different ANN to formulate different base classifiers. The results obtained on KDD cup 1999 dataset [17] were better than decision tree and naïve Bayes methods in terms of precision and stability of detection.

Hansen and Salamon's work shows that the generalization ability of a neural network system can be significantly improved through ensembling a number of neural networks, *i.e.* training many neural networks and then combining their predictions [28]. For ANN-based intrusion detection, hybrid/ensemble of ANN has been the trend [9-11,27,29]. But different ways to construct hybrid/ensemble ANN will highly influence the performance of intrusion detection. Different hybrid ANN models should be properly constructed in order to serve different aims.

Empirical results of above cited studies indicated that the ANN based intrusion detection is efficient. Following this stream, we propose an ensemble of ANN, called MANNE, to solve the drawbacks of current ANN-based IDS mentioned in Section 1, *i.e.* lower DR and high FPR. MANNE approach introduces multi-objective genetic algorithm to evolve ANN as base classifiers. These base classifiers are further used as ensemble members to improve DR and FPR, especially for minority attack classes. The detailed framework of MANNE is described in Section 3. Exploiting a number of views of a same problem with classifier ensembles has been shown to improve the overall accuracy and reliability for a wide range of applications [30]. However, generating an accurate pool of base classifiers and selecting an ensemble among that pool that maximizes prediction accuracy are challenging

tasks. One key element in the success of classifier ensembles that has attracted a great deal of interest in recent years is classifier diversity measures. Since diversity is difficult to assess in the input feature space, these measures compute the disagreement between classifiers in the decision space, over several predictions. Through bias-variance error decomposition, it has been shown empirically that considering diversity for ensemble selection improves the generalization capabilities of ensembles [30]. Diversity among the base classifiers mostly lies on “varying” the parameters related to the design and to the training of ANN [31]. In particular, the main methods in the literature can be included in one of the following categories:

- Varying the initial random weights: an ensemble of nets can be created by varying the initial random weights, from which each network is trained;
- Varying the network architecture: the nets forming the ensemble exhibit different architectures;
- Varying the network type: different net types (e.g., multilayer perceptrons, radial basis functions neural networks, and probabilistic neural networks) can be used to create the ensemble members;
- Varying the training data: an ensemble of nets can be created by training each network with a different learning set. This can be done in a number of different ways. For example, “sampling” the training data to obtain different learning sets, using learning sets extracted from different data “sources” (e.g., data from different imaging sensors), or by different pre-processing phases (e.g., sets formed by samples characterized by different “features”).

Many researchers presented various approaches to design ANN based ensembles. These approaches may be categorized as: 1) direct approach; 2) overproduce and choose approach. The former approach is designed to create an ensemble of diverse ANN directly. For example ASSEMUP, here authors utilized genetic algorithm (GA) to search for an ensemble of diverse ANNs [32].

Whereas later approach (overproduce and choose approach) focuses on creation of an initial large pool of ANNs and the later on choose the subset of the most diverse ANNs to create ensembles. Sharkey and Sharkey also described a design method that follows the “overproduce and choose” strategy [33]. Their choice algorithm is basically guided by a heuristic based on the evaluation of error correlation between pairs of ANNs.

In ensemble classifiers, the predictions of base classifiers can be combined to produce final prediction by using different approaches namely fusion and selection. Fusion approach involves the combination of the predictions obtained by the different classifiers in the ensemble to obtain the final prediction. The important method used for fusion are: 1) Majority voting method; 2) Threshold

plurality vote method; 3) Naïve Bayes method; 4) Fuzzy theory method etc. whereas the selection approach, especially its dynamic form, selects one (or more) classifiers from the ensemble according to the prediction performance of these classifiers on similar data from the validation set. The important methods used for selection are: 1) The test and select method; 2) Cascading classifiers method; 3) Dynamic Classifier Selection method; 4) Clustering based selection method etc. Hansen and Salamon (1990) showed that ANN combined by the majority voting method can provide increases in classification accuracy, only if the ANNs make independent errors (diverse in nature) [28].

Therefore, our approach to the design of ANN based ensembles is to create a set of ANN evolved by MOGA exhibiting the highest possible degree of error diversity and ensemble of ANNs thereof using multi-objective approach. The motivation for using multi-objective approach is its advantages over single objective approaches. Single objective approach deals with multiple evaluation functions by transforming them into one objective [16]. Transformation of multiple objectives into single objective often requires prior knowledge about the problem or heuristic that leads the approach to single objective [15]. This means there is requirement of expert in concerned method or approach and specific problem. Even then it is not sure that all the solutions in optimum Pareto front could be found. But multi-objective approaches offer a set of non inferior solutions that reveals different tradeoff between different objectives. This tradeoff is known as Pareto optimum. For example, MOGA may present the Pareto front in single execution without requiring any expert about the method or problem.

3. Framework of MANNE

This section elaborates our approach, MANNE. The MANNE works in following phases: namely, 1) evolution of pool of ANN as base classifiers; and 2) creation of pool of ensemble of ANNs giving classification trade-offs; 3) combiner. Phases 1 and 2 are multi-objective. These phases present the user an approximation of the Pareto front of non-dominated solutions of both base classifiers and ensembles. The detection rate of each class is adopted as separate objectives in these phases. Each ANN is evolved by using MOGA to exhibit different classification trade-offs. It helps to maintain diversity among the base classifiers implicitly in MANNE. In phase 2, MANNE optimizes the selection of base classifiers to obtain a Pareto front of ensembles that exhibit different classification trade-offs to the users. Phase 3 combines the predictions of base classifiers to get the final results of ensemble classifier. The details are as follows.

Phase 1: Evolution of pool of diverse ANNs

This phase evolves the weights for pool of ANNs to formulate diverse base classifiers using MOGA. The diversity among base classifiers is maintained implicitly by varying initial random weights used for multi objective GA (MOGA) e.g. NSGA II [16]. The detection rate of each class is treated as separate objective. Here, The MOGA is real-coded, uses crossover and mutation operators, an elitist replacement strategy. To enable optimization of multiple objectives, non-dominated sorting is adopted, additionally, an archive function is adopted similar to that of NSGA-II [16] and SPEA [35]. This maintains a set of all non-dominated solutions found from the start of the evolutionary process. This phase of MOGA is able to find optimal Pareto front of non-dominated ANN solutions (**Figure 1**). These ANN solutions formulate the base classifiers as candidate solutions for ensemble generation in phase 2.

Phase 2: Evolution of pool of ensembles of ANNs

Phase 2 is also multi objective in nature. Phase 2 takes input in form of archive of non-dominated ANN solutions produced by phase 1. The phase evolved ensemble classifiers by combining the Pareto front of nondominated solutions instead of the entire population like other studies [15,36]. The detection rate of each class is treated as separate objective. Here, we are interested in those solutions with non dominated trade off. The predictions of base classifiers are combined using majority voting method. In case of a tie, the winner is randomly chosen. The MOGA method discussed in phase 1 is again applied in phase 2. Here, MOGA is binary coded where 0 represents the non-participation and 1 represents participation of concerned base classifier in creating ensembles. The

detection rate of each class is treated as separate objective. The output of phase 2 is an archive of ensemble of ANNs in terms of chromosomes (sequences) of 1's and 0's (**Figure 2**). Here, 1 represents incorporation of corresponding ANN whereas 0 represents absence of corresponding ANN in formation of ensemble. The set of ensembles exhibit the classification trade-off for different objective functions.

Phase 3: Prediction aggregation

This phase is responsible for aggregating the predictions of different ANNs base classifiers to get final prediction of ensemble classifier. The phase takes two inputs: 1) archive of non-dominated ANN solutions (output of phase 1); 2) one chromosome (sequence of 1's and 0's) from archive of ensembles (output of phase 2). An ensemble classifier can be selected by using static or dynamic strategy. Here in this work, we selected the ensemble classifier using static strategy based from its performance on the training data in terms of pre-defined performance metric. Based upon the values of chromosome, corresponding ANNs predictions are aggregated to get final prediction of the ensemble. In order to test the proposed approach, the test dataset is directly fed to different ANNs and get the outputs. Final prediction of ensemble is computed by using majority voting method in prediction aggregation module. Here, we used majority voting method to combine the predictions of base classifiers (**Figure 3**).

These three phases of MANNE framework highlights three major issues of current research. The issues are: 1) creation of pool of non-dominated ANN solutions to formulate base classifiers; 2) creation of pool of non-dominated ensemble solutions that exhibit classification trade-

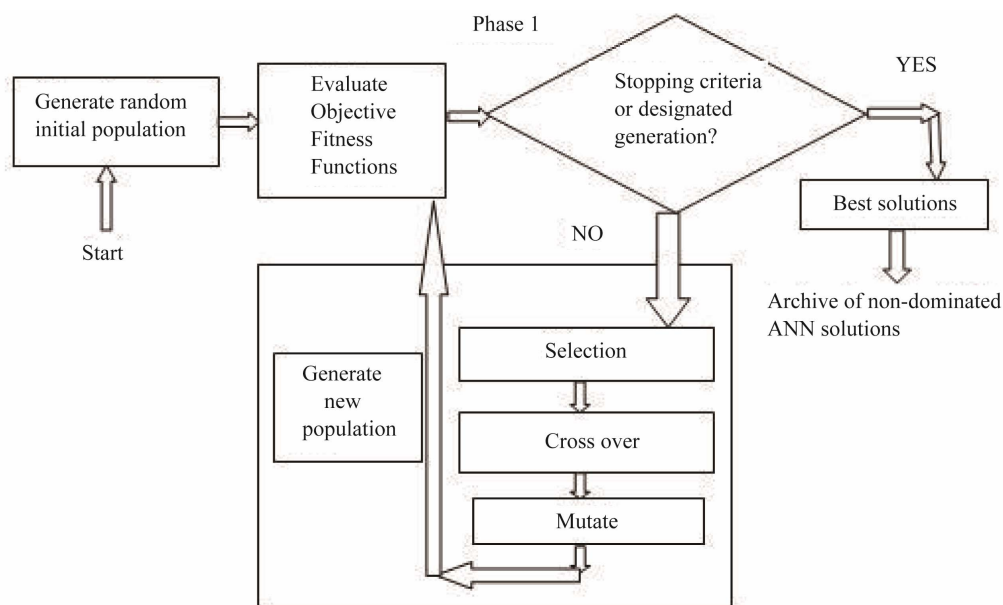


Figure 1. Phase 1 of MANNE approach.

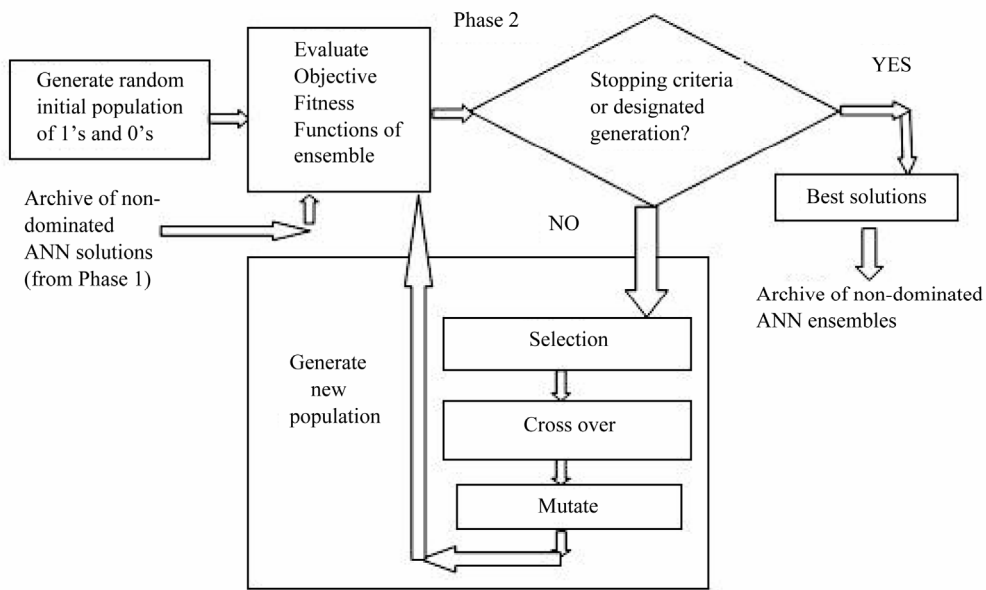


Figure 2. Phase 2 of MANNE approach.

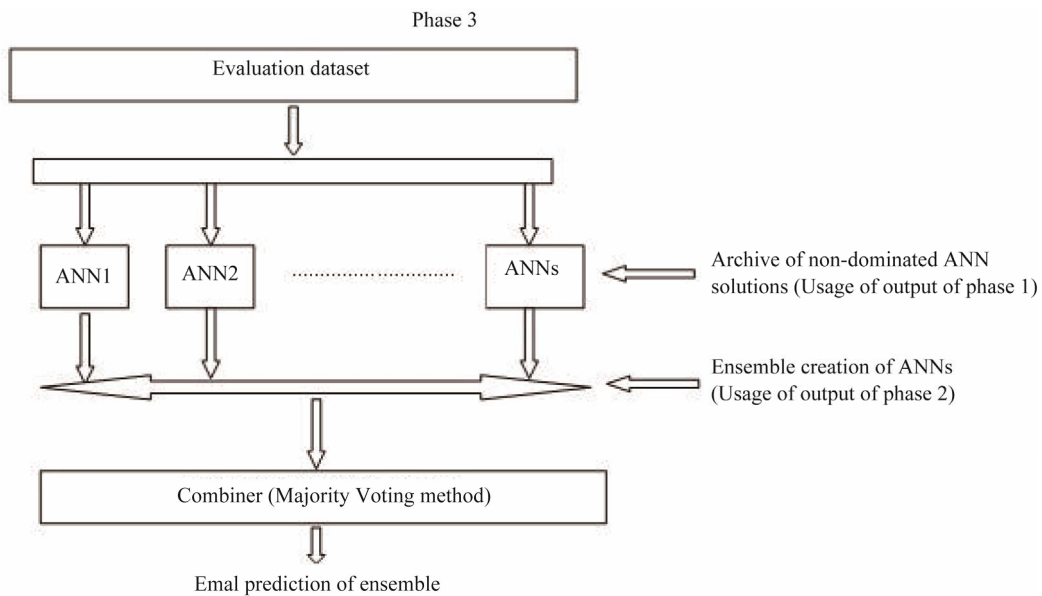


Figure 3. Phase 3 of MANNE approach.

off; 3) efficient combination of predictions of base classifiers to get final prediction of ensemble.

4. Implementation

To evaluate the performance of MANNE approach, a series of experiments were conducted on KDD CUP 1999 dataset. In these experiments, we implemented and evaluated the proposed methods in VC++ on a Windows PC with Core i3-2330M 2.20 GHz CPU and 2 GB RAM. We used MLPFF ANN as base classifiers. NSGA-II [16], the multi objective genetic algorithm is used to evolve the base classifier and their ensembles. Majority voting

method is used as method for combining the predictions of base classifiers. Following sub-sections describe the details of MLPFF ANN and NSGA-II used in this set of experiments.

4.1. MLP ANN

An MLP is a network of simple neurons called perceptrons [37]. The perceptron computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights and then possibly putting the output through some non-linear activation function. In other words, MLPs are feed forward ANNs may

be trained with the standard back propagation algorithm [37] or by using other alternative techniques as depicted in **Figure 4**. They are supervised networks, so they require a desired response to be trained. They learn how to transform input data into a desired response, so they are widely used for pattern classification. With one or two hidden layers, they can approximate virtually any input-output map. They have been shown to approximate the performance of optimal statistical classifier in difficult problems. The MLPNN used in this study is composed of three neuron layers, namely, the input layer, the output layer and the hidden layer as shown in **Figure 4**. Although the MLPNN can have more than one hidden layer, having more than one hidden layer is rarely beneficial and can lead to gross over-parameterization [38].

For a particular instance i of training/test dataset, the input layer of the MLP ANN used for intrusion detection receives the input vector T from training dataset. The input vector T has general format

$$T_i = (t_{i,1}, t_{i,2}, \dots, t_{i,n}) \quad (1)$$

Here, $t_{i,j}$ is the j th feature of i th instance of training/test dataset. Total number of input neurons in input layer is equal to total features of training/test dataset for intrusion detection. The output layer contains the output neurons. The output neurons are equal to number of classes in dataset.

A hidden layer is a middle layer. This layer adds a degree of flexibility to the performance of the ANN that enables it to deal efficiently with complex nonlinear problems. Each neuron in the single hidden layer receives the same input vector of N elements from the neurons of the input layer, as defined by Equation (1), and produces the output. The input-output transformation in each hidden neuron is achieved by a mathematical non-linear transfer (or activation) function. The general form of activation function is

$$Y_{i,k} = f\left(\sum_{j=1}^N w_{j,k} * T_{i,j} + b_k\right) \quad (2)$$

where $Y_{i,k}$ is the output of k th neuron in hidden layer for i th instance of dataset, $f(\)$ is activation function, $w_{j,k}$ is the connection weight assigned to k th hidden neuron and j th neuron in input layer and b_k is the bias of k th hidden neuron. In literature, many activation functions are proposed [38]. The most widely used activation function is the sigmoid function which can be expressed as

$$Y_{i,k} = \left[1 + \exp\left(-\sum_{j=1}^N w_{j,k} * T_{i,j} - b_k\right)\right]^{-1} \quad (3)$$

The neurons in output layer produce the final network output. These output neurons receive an input array in form of Equation (4).

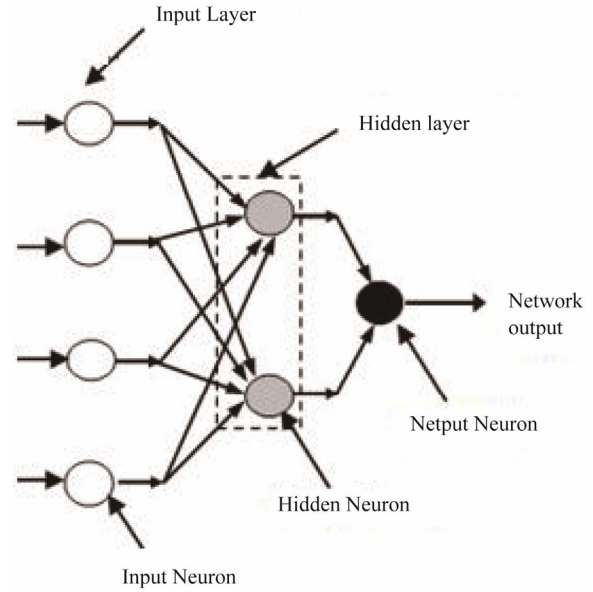


Figure 4. Structure of MLP ANN.

$$Z_i = (Y_{i,1}, Y_{i,2}, \dots, Y_{i,n}) \quad (4)$$

The input-output transformation for this output neuron is similar to that of the hidden neurons.

4.2. MOGA

For multiple-objective problems like intrusion detection, there exist objectives which generally conflict with each other. For example, the objective may be to minimize cost, maximize detection rate, minimize false positive rates etc. Genetic algorithm (GA) (population based approach) is a popular meta-heuristic that is particularly suited for such class of problems [39]. GA can be applied in two ways to solve multi objective problem. 1) The first way combine the individual objective functions into a single composite function or move all but one objective to the constraint set. Such single objective function can be determined by using methods like utility theory, weighted sum method, etc. But, major limitations are to select the weights of different objectives and proper single objective functions. This approach returns a single solution that lack in examining the trade-offs between objectives [15]; 2) The second way determine a set of entire Pareto optimal solutions which contains solutions that are non-dominated with respect to each other. Pareto optimal solution sets are generally favored to single solution because these solutions can be more useful since the final solution of the decision-maker is always a trade-off between objectives. Thus Multi objective GA may sample the Pareto front in a single run without incorporating any domain knowledge about the problem. These features of MOGA motivated us to apply it to evolve weights of MLPFF ANN.

Many researchers proposed various MOGAs; details can be further explored in [40]. Among many MOGAs proposed in literature, NSGA-II is a fast, elitist and generational algorithm which is widely used for multi-objective optimization problems [16]. The important features of NSGA II are: 1) A full elite preservation strategy and diversity preserving mechanism using crowding distance as distance measure without setting any parameter; 2) Elitism is used to provide the means to keep good solutions among the generations and diversity preserving mechanism is used to allow a better spread among solutions over Pareto front [52]. NSGA-II preserves diversity among non-dominated solutions by use of crowding (instead of sharing). NSGA-II utilizes the concept of dominance to determine the new population after each generation (at which point selection and recombination have taken place to produce an offspring population of the same size as the old population). Before classifying individuals according to the dominance concept, parent and offspring populations of size n are combined to form a global population of size $2n$. Non-dominated sorting is then applied to this global population, which identifies sets of non-dominated individuals. A new population of size n is then created from the sets according to their rank. If all individuals of a set cannot fit into the population, niching is used to select individuals in the least crowded region of the set. Consecutive sets, if any, are discarded [15,16,40]. Further details can be found in [16].

4.3. Combining MOGA with ANN

Combination of MOGA and ANN remains focus of research community since its first combination in late 80 s. Combination of these autonomous computing methods attempts to solve the problem of ANN. The problem with ANN is that a number of parameters have to be set before any training can begin [41]. However, there are no clear rules how to set these parameters. Yet these parameters determine the success of the training. GA is used to find these parameters. The motivation for this concept of combining MOGA with ANN comes from nature. As in real life, the success of an individual is not only dependent by the knowledge and skills, which he gained through, experience (like ANN training), but it also depends on his genetic heritage (set by MOGA). MOGA combined with ANN proved to be very successful for most of complex problems. In these experiments, we used MLP ANN and NSGA-II with following parameters.

5. Experiments and Results

In order to evaluate the performance of MANNE approach, we conducted the experiments to evolve ANN using NSGA-II initiated with parameter described in **Table 1** based upon KDD CUP 1999 dataset for intrusion detection.

Table 1. Initialization parameters of MLP ANN and NSGA-II.

MLP ANN	
Input nodes	41
Hidden layer	1
Hidden nodes	30
Output node	5
Activation function	Sigmoid function
NSGA-II	
No. of generations	500
Population size	100
Cross over rate	0.9
Mutation rate	0.3

5.1. Dataset

KDD CUP 1999 dataset [17] for intrusion detection is used to evaluate MANNE approach. The KDD CUP 1999 dataset contains about five million connection records as training data and about two million connection records as test data which are derived from 1998 DARPA intrusion detection evaluation program by MIT Lincoln labs. The connection records have 41 features of continuous, discrete and symbolic features. Each connection record is labeled as either normal or attack type. These attack types can be categorized into four classes namely: 1) Probe; 2) DoS; 3) U2R; and 4) R2L. Although KDD99 data-set might have been criticized for its potential problems [34,42,43], but many researchers give the priority to KDD dataset over other publicly available dataset as benchmark dataset for evaluation of IDS [1, 44]. Tavallae *et al.* (2009) proposed a refined form of KDD 99 dataset and called it as NSL KDD dataset [42]. They claimed certain improvement of KDD dataset deficiencies as criticized by McHugh 2000 [43]. Since the number of connection records in NSL KDD dataset is very large, so we randomly selected connection records from training and test dataset to reduce the size of datasets. Total number of instances in reduced training and test dataset are as depicted in **Table 2**. The training and test datasets are further preprocessed to make it compatible with ANN as described in [45].

5.2. Evaluation Criteria

In order to evaluate the effectiveness of IDS, we measure its ability to correctly classify events as normal or intrusive along with other performance objectives, such as economy in resource usage, resilience to stress and ability to resist attacks directed at IDS [46]. Measuring this ability of IDS is important to both industry as well as research community. It helps us to tune the IDS in better way as well as compare different IDSs. There exist many

Table 2. Summary of training and test dataset used.

Dataset	Class	Training instances	Test instances
Instances	Normal	1000	500
	Probe	100	75
	DoS	100	75
	U2R	11	50
	R2L	100	50
	Total		1311

metrics that measure different aspects of IDS, but no single metric seems sufficient to objectively measure the capability of intrusion detection systems. Most widely used metrics by intrusion detection research community are True Positive Rate (TPR) and False Positive Rate (FPR). False Negative rate $FNR = 1 - TPR$ and True Negative Rate $TNR = 1 - FPR$ can be used as an alternate. Based upon values of these two metrics only, it is very difficult to determine better IDS among different IDSs. For example, one IDS reporting, $TPR = 0.8$; $FPR = 0.1$, while at another IDS, $TPR = 0.9$; $FPR = 0.2$. If only the metrics of TPR; FPR are given, it is very difficult to determine the better IDS. To solve this problem, Guofoei *et al.* (2006) proposed a new objective metric called Intrusion Detection Capability (CID) considering base rate, TPR and FPR collectively [46]. CID possesses many important features. For example, 1) it naturally takes into account all the important aspects of detection capability, *i.e.* FPR, FNR, positive predictive value (PPV [47]), negative predictive value (NPV), and base rate (the probability of intrusions); 2) it objectively provides an essential measure of intrusion detection capability; 3) it is very sensitive to IDS operation parameters such as base rate, FPR and FNR. Detail of CID can be further studied in [46]. Keeping these points in view, we compute TPR, FPR and CID to evaluate the performance of MANNE approach and compare it with other intrusion detection approaches.

5.3. Results & Discussion

We performed experiments by randomly selecting instances from KDD cup 1999 dataset as described in Subsection 5.1. We compared the results with MLP trained with Back Propagation method (MLP-BP), and other well-known methods such as decision tree (J48). We also compared the results of our approach MANNE with two most widely used ensemble methods *i.e.* Bagging [18] and Boosting [19]. While creating ensembles using Bagging and Boosting, we used MLP and Decision tree (J48) as base classifiers. These techniques were executed using WEKA Data Mining tool [48]. The results of experiments are shown in **Tables 3-8** and **Figures 5** and **6**.

As described in [46], it is difficult to compare the re-

Table 3. Performance comparison of various methods (Normal class).

	TPR	FPR	CID
MLP	0.898	0.544	0.1218
Bagged MLP	0.894	0.484	0.1538
Boosted MLP	0.898	0.544	0.1218
J48	0.864	0.516	0.1077
Bagged J48	0.866	0.54	0.0967
Boosted J48	0.892	0.572	0.10099
MANNE	0.848	0.32	0.2195

Table 4. Performance comparison of various methods (Probe class)

	TPR	FPR	CID
MLP	0.627	0.079	0.2424
Bagged MLP	0.827	0.084	0.4036
Boosted MLP	0.627	0.079	0.2424
J48	0.64	0.059	0.2891
Bagged J48	0.64	0.062	0.2830
Boosted J48	0.68	0.056	0.3285
MANNE	0.8	0.08	0.3852

Table 5. Performance comparison of various methods (DoS class).

	TPR	FPR	CID
MLP	0.507	0.028	0.2596
Bagged MLP	0.52	0.025	0.2790
Boosted MLP	0.507	0.028	0.2596
J48	0.493	0.068	0.1682
Bagged J48	0.507	0.061	0.1882
Boosted J48	0.547	0.027	0.2944
MANNE	0.4267	0.0533	0.1490

Table 6. Performance comparison of various methods (U2R class).

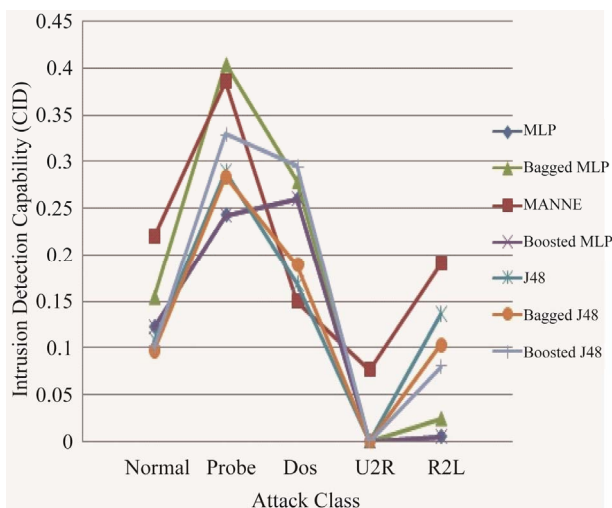
	TPR	FPR	CID
MLP	0	0.544	0
Bagged MLP	0	0.484	0
Boosted MLP	0	0.544	0
J48	0	0.516	0
Bagged J48	0	0.54	0
Boosted J48	0	0.572	0
MANNE	0.022857	0.32	0.07628

Table 7. Performance comparison of various methods (R2L class).

	TPR	FPR	CID
MLP	0.02	0.003	0.0047
Bagged MLP	0.06	0.004	0.0238
Boosted MLP	0.02	0.003	0.0047
J48	0.24	0.006	0.1357
Bagged J48	0.18	0.004	0.1027
Boosted J48	0.14	0.003	0.0797
MANNE	0.42	0.027	0.1898

Table 8. Performance comparison of various methods (based on weighted average of TPR & FPR of all classes).

	TPR	FPR	CID
MLP	0.72	0.374	0.0857
Bagged MLP	0.736	0.334	0.1163
Boosted MLP	0.72	0.374	0.0857
J48	0.708	0.357	0.088
Bagged J48	0.705	0.373	0.0787
Boosted J48	0.732	0.39	0.0841
MANNE	0.727	0.23	0.1825

**Figure 5. Class-wise comparison of different methods based upon CID.**

sults of different intrusion detection approaches only by observing the values of TPR and FPR. So, we compare different approaches based upon CID. As shown by above **Tables 3-7**, we can clearly observe the difference of evaluation criteria (CID) under different attacks types, *i.e.* Normal, Probe, DoS, U2R and R2L. While MANNE gets comparable results for majority attack classes *i.e.* Probe and DoS. MANNE reports higher values of CID than Decision tree (J48), MLP and their ensemble approaches for Normal and minority attack classes *i.e.* U2R and R2L.

As depicted in **Figure 5**, all other approaches except Bagged MLP get similar results for Normal, DoS, U2R and R2L classes. For Probe attack class, Bagged MLP shows comparable performance to MANNE. Whereas, for Normal, U2R and R2L classes, MANNE shows better performance than other six intrusion detection approaches used in these experiments. **Table 8** and **Figure 6** present the results of different approaches based upon weighted average of TPR and FPR for all classes. The results clearly indicate that MANNE is better approach for intrusion detection in terms of high TPR, low FPR and more importantly high value of CID. On average results, Bagged MLP shows comparable TPR, high FPR and low CID than MANNE. It may also be observed that bagged MLP has reported improvement over single MLP classifier. However Boosted MLP does not attain significant improvements. This validates the conclusions of prior research that are: 1) performance of bagging and boosting ensemble methods is affected by the training size [49,50]; 2) boosting is the best for large training sample sizes and bagging is useful for critical training sample sizes [50,51]. Due to small size of training dataset adopted in this study, performance of bagged MLP is superior to boosted MLP.

Therefore, we can draw the conclusion that our proposed approach, MANNE can get higher detection TPR, low FPR and high CID especially for Normal and minority classes. But the major limitation of MANNE is the long time to find Pareto front of non-dominated ANN solutions using MOGA. The computation of fitness functions of different ANN consumes is computationally expensive. The limitation can be overcome by computing fitness function of ANNs in parallel.

6. Concluding Remarks and Future Directions

In this study, we propose a new intrusion detection approach, called MANNE based upon ensemble of ANN evolved by Multi Objective Genetic Algorithm (MOGA). The MANNE generates Pareto front of non-dominated ANN solutions and their ensembles thereof to exhibit classification trade-offs in different phases. Majority voting method is used to determine the final prediction of ensemble from predictions of base classifiers. The MANNE is validated by using KDD cup 1999 dataset for intrusion detection. The results of experiments conducted validate the usage feasibility of our approach MANNE. The efficiency of MANNE approach over single classifier and their conventional ensembles (Bagging and Boosting) is proved.

Since, calculation of fitness functions in various generations by MOGA is computationally expensive, so MANNE take long time to find Pareto optimal ANN solutions and their ensemble thereof. This is major limita-

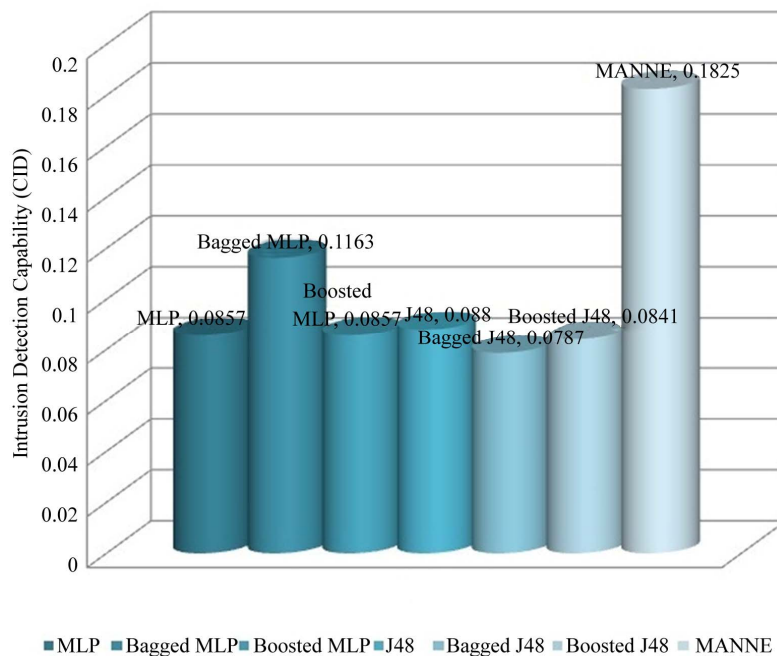


Figure 6. Comparison of different methods based upon CID.

tion of proposed approach that may be overcome by computing the function values in parallel. Here, we compute the results by limiting the population size and number of generations of MOGA. The results can be further improved by choosing appropriate values of these parameters. We validate our approach using a small subset of KDD CUP 1999 dataset only. The applicability of our approach can be tested for different intrusion detection datasets.

So, our future research will be to carry out more experiments by choosing different values of population size as well as number of generations upon different intrusion detection datasets to improve the classification results. Another direction for research work is to explore different techniques to reduce the training time of ANNs.

REFERENCES

- [1] G. Kumar, K. Kumar and M. Sachdeva, "The Use of Artificial Intelligence Based Techniques for Intrusion Detection—A Review," *Artificial Intelligence Review*, Vol. 34, No. 4, 2010, pp. 369-387. [doi:10.1007/s10462-010-9179-5](https://doi.org/10.1007/s10462-010-9179-5)
- [2] J. P. Anderson, "Computer Security Threat Monitoring and Surveillance," Technical Report, James P. Anderson Company, Fort Washington, 1980.
- [3] C. Endorf, E. Schultz and J. Mellander, "Intrusion Detection and Prevention," McGraw-Hill, New York, 2004.
- [4] L. R. Halme and R. K. Bauer, "AINT Misbehaving: A Taxonomy of Anti-Intrusion Techniques," *Computers and Security*, Vol. 14, No. 7, 1995, p. 606. [doi:10.1016/0167-4048\(96\)81669-5](https://doi.org/10.1016/0167-4048(96)81669-5)
- [5] A. Patcha and J. M. Park, "An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends," *Computer Networks*, Vol. 51, No. 12, 2007, pp. 3448-3470. [doi:10.1016/j.comnet.2007.02.001](https://doi.org/10.1016/j.comnet.2007.02.001)
- [6] P. Dokas, L. Ertoz, A. Lazarevic, J. Srivastava and P. N. Tan, "Data Mining for Network Intrusion Detection," *Proceedings of NSF Workshop on Next Generation Data Mining*, November 2002, pp. 21-30.
- [7] S. Wu and E. Yen, "Data Mining-Based Intrusion Detectors," *Expert Systems with Applications*, Vol. 36, No. 3, 2009, pp. 5605-5612. [doi:10.1016/j.eswa.2008.06.138](https://doi.org/10.1016/j.eswa.2008.06.138)
- [8] J. Ryan, M. Lin and R. Miiikkulainen, "Intrusion Detection with Neural Networks," Springer, Cambridge, 2002.
- [9] G. Wang, J. Hao, J. Ma and L. Huang, "A New Approach to Intrusion Detection Using Artificial Neural Networks and Fuzzy Clustering," *Expert Systems with Applications*, Vol. 37, No. 9, 2010, pp. 6225-6232. [doi:10.1016/j.eswa.2010.02.102](https://doi.org/10.1016/j.eswa.2010.02.102)
- [10] M. Govindarajan and R. M. Chandrasekaran, "Intrusion Detection Using Neural Based Hybrid Classification Methods," *Computer Networks*, Vol. 55, No. 8, 2011, pp. 1662-1671. [doi:10.1016/j.comnet.2010.12.008](https://doi.org/10.1016/j.comnet.2010.12.008)
- [11] V. Engen, J. Vincent, A. C. Schierz and K. Phalp, "Multi-Objective Evolution of the Pareto Optimal Set of Neural Network Classifier Ensembles," *Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC)*, Baoding, 2009, pp. 74-79.
- [12] J. Han, M. Kamber and J. Pei, "Data Mining: Concepts and Techniques," 3rd Edition, Morgan Kaufmann, Burlington, 2011.
- [13] V. Engen, J. Vincent and K. Phalp, "Exploring Discrepancies in Findings Obtained with the KDD Cup '99 Data

- Set," *Intelligent Data Analysis*, Vol. 15, No. 2, 2011, pp. 251-276.
- [14] D. Joo, T. Hong and I. Han, "The Neural Network Models for IDS Based on the Asymmetric Costs of False Negative Errors and False Positive Errors," *Expert Systems with Applications*, Vol. 25, No. 1, 2003, pp. 69-75. doi:10.1016/S0957-4174(03)00007-1
- [15] V. Engen, "Machine Learning for Network Based Intrusion Detection: An Investigation into Discrepancies in Findings with the KDD Cup '99 Data Set and Multi-Objective Evolution of Neural Network Classifier Ensembles for Imbalanced Data," Ph.D. Thesis, School of Design, Engineering and Computing, Bournemouth University, Bournemouth, 2010.
- [16] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 2002, pp. 182-197. doi:10.1109/4235.996017
- [17] KDD, "KDD Cup 1999 Dataset," 1999. <http://archive.ics.uci.edu/ml/datasets/KDD+Cup+1999+Data>
- [18] L. Breiman, "Bagging predictors," *Machine Learning*, Vol. 24, No. 2, 1996, pp. 123-140. doi:10.1007/BF00058655
- [19] Y. Freund and R. E. Shapire, "A Decision-Theoretic Generalization of on Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, Vol. 55, No. 1, 1997, pp. 119-139. doi:10.1006/jcss.1997.1504
- [20] O. Depren, M. Topallar, E. Anarim and M. K. Ciliz, "An Intelligent Intrusion Detection System (IDS) for Anomaly and Misuse Detection in Computer Networks," *Expert Systems with Applications*, Vol. 29, No. 4, 2005, pp. 713-722. doi:10.1016/j.eswa.2005.05.002
- [21] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez and E. Vazquez, "Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges," *Computers & Security*, Vol. 28, No. 1, 2009, pp. 18-28. doi:10.1016/j.cose.2008.08.003
- [22] S. Mukkamala, G. Janoski and A. Sung, "Intrusion Detection Using Neural Networks and Support Vector Machines," *Proceedings of the IEEE International Joint Conference on Neural Networks*, 2002, pp. 1702-1707.
- [23] R. Cunningham and R. Lippmann, "Improving Intrusion Detection Performance Using Keyword Selection and Neural Networks," *Computer Networks*, Vol. 34, No. 4, 2000, pp. 597-603. doi:10.1016/S1389-1286(00)00140-7
- [24] S. J. Han and S. B. Cho, "Evolutionary Neural Networks for Anomaly Detection Based on the Behavior of a Program," *IEEE Transactions on Systems, Man and Cybernetics (Part B)*, Vol. 36, No. 3, 2005, pp. 559-570. doi:10.1109/TSMCB.2005.860136
- [25] C. Jirapummin, N. Wattanapongsakorn and P. Kanthamanon, "Hybrid Neural Networks for Intrusion Detection System," *Proceedings of ITC-CSCC*, July 2002, pp. 928-931.
- [26] A. Ghosh and A. Schwartzbard, "A Study in Using Neural Networks for Anomaly and Misuse Detection," *Proceedings of the 8th USENIX Security Symposium*, Washington DC, 1999, pp. 141-152.
- [27] Y. H. Chen, A. Abraham and B. Yang, "Hybrid Flexible Neural-Tree-Based Intrusion Detection Systems," *International Journal of Intelligent Systems*, Vol. 22, No. 4, 2007, pp. 337-352. doi:10.1002/int.20203
- [28] L. K. Hansen and P. Salamon, "Neural Network Ensembles," *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 12, No. 10, 1990, pp. 993-1001. doi:10.1109/34.58871
- [29] S. Sahin, M. R. Tolun and R. Hassanpour, "Hybrid expert Systems: A Survey of Current Approaches and Applications," *Expert Systems with Applications*, Vol. 39, No. 4, 2012, pp. 4609-4617. doi:10.1016/j.eswa.2011.08.130
- [30] J. Francois Connolly, E. Granger and R. Sabourin, "Evolution of Heterogeneous Ensembles through Dynamic Particle Swarm Optimization for Video-Based Face Recognition," *Pattern Recognition*, Vol. 45, 2012, pp. 2460-2477. doi:10.1016/j.patcog.2011.12.016
- [31] G. Giacinto and F. Roli, "Design of Effective Neural Network Ensembles for Image Classification Purposes," *Image and Vision Computing*, 2001, Vol. 19, No. 9-10, pp. 699-707. doi:10.1016/S0262-8856(01)00045-2
- [32] D. W. Opitz and J. W. Shavlik, "Actively Searching for an Effective Neural Network Ensemble," *Connection Science*, Vol. 8, No. 3-4, 1996, pp. 337-353. doi:10.1080/095400996116802
- [33] A. J. C. Sharkey and N. E. Sharkey, "Combining Neurals Nets," *The Knowledge Review*, Vol. 12, No. 3, 1997, pp. 231-247. doi:10.1017/S0269888997003123
- [34] C. Brown, A. Cowperthwaite, A. Hijazi and A. Somayaji, "Analysis of the 1999 DARPA/Lincoln Laboratory IDS Evaluation Data with Netadict," *Proceedings of the 2nd IEEE International Conference on Computational Intelligence for Security and Defense Applications*, Piscataway, 2009, pp. 1-7.
- [35] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, 1999, pp. 257-271. doi:10.1109/4235.797969
- [36] H. Ishibuchi and Y. Nojima, "Evolutionary Multiobjective Optimization for the Design of Fuzzy Rulebased Ensemble Classifiers," *International Journal of Hybrid Intelligent Systems*, Vol. 3, No. 3, 2006, pp. 129-145.
- [37] C. M. Bishop, "Neural Networks for Pattern Recognition," Oxford University Press, Oxford, 1995.
- [38] A. Y. Shamseldin, K. M. O. Connor and A. E. Nasr, "A Comparative Study of Three Neural Network Forecast Combination Methods for Simulated River Flows of Different Rainfall—Runoff Models," *Hydrological Sciences Journal*, Vol. 52, No. 5, 2007, pp. 896-916. doi:10.1623/hysj.52.5.896
- [39] A. Konak, D. W. Coit and A. E. Smith, "Multi-Objective Optimization Using Genetic Algorithms: A Tutorial," *Reliability Engineering and System Safety*, Vol. 91, No. 9, 2006, pp. 992-1007. doi:10.1016/j.res.2005.11.018
- [40] A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan and Q. Zhang, "Multiobjective Evolutionary Algorithms:

- A Survey of the State of the Art,” *Swarm and Evolutionary Computation*, Vol. 1, No. 1, 2011, pp. 32-49. [doi:10.1016/j.swevo.2011.03.001](https://doi.org/10.1016/j.swevo.2011.03.001)
- [41] P. Koehn, “Combining Genetic Algorithms and Neural Networks: The Encoding Problem,” MS Thesis, The University of Tennessee, Knoxville, 1994.
- [42] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set,” *Proceedings of IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA)*, 2009. [doi:10.1109/CISDA.2009.5356528](https://doi.org/10.1109/CISDA.2009.5356528)
- [43] J. McHugh, “Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory,” *ACM Transactions on Information and System Security*, Vol. 3, No. 4, 2000, pp. 262-294. [doi:10.1145/382912.382923](https://doi.org/10.1145/382912.382923)
- [44] C. F. Tsai, Y. F. Hsu, C. Y. Lin and W. Y. Lin, “Intrusion Detection by Machine Learning: A Review,” *Expert Systems with Applications*, Vol. 36, No. 10, 2009, pp. 11994-12000. [doi:10.1016/j.eswa.2009.05.029](https://doi.org/10.1016/j.eswa.2009.05.029)
- [45] G. Kumar, K. Kumar and M. Sachdeva, “An Empirical Comparative Analysis of Feature Reduction Methods for Intrusion Detection,” *International Journal of Information and Telecommunication*, Vol. 1, No. 1, 2010, pp. 44-51.
- [46] G. Gu, P. Fogla, D. Dagon, W. Lee and B. Skoric, “Measuring Intrusion Detection Capability: An Information-Theoretic Approach,” *Proceedings of ACM Symposium on InformAction, Computer and Communications Security (ASIACCS’06)*, March 2006, pp. 90-101.
- [47] S. Axelsson, “The Base-Rate Fallacy and Its Implications for the Difficulty of Intrusion Detection,” *ACM Transactions on Information and System Security (TISSEC)*, Vol. 3, No. 3, 2000, pp. 186-205.
- [48] I. H. Witten and E. Frank, “Data Mining: Practical Machine Learning Tools and Techniques,” 2nd Edition, Morgan Kaufmann, San Francisco, 2005.
- [49] S. Sun, C. Zhang and D. Zhan, “An Experimental Evaluation of Ensemble Methods for EEG Signal Classification,” *Pattern Recognition Letters*, Vol. 28, No. 15, 2007, pp. 2157-2163.
- [50] R. Das and A. Sengur, “Evaluation of Ensemble Methods for Diagnosing of Valvular Heart Disease,” *Expert Systems with Applications*, Vol. 37, No. 7, 2010, pp. 5110-5115.
- [51] M. Skurichina and R. P. W. Duin, “Bagging, Boosting and the Random Subspace Method for Linear Classifiers,” *Pattern Analysis and Applications*, Vol. 5, No. 2, 2002, pp. 121-135. [doi:10.1007/s100440200011](https://doi.org/10.1007/s100440200011)
- [52] E. M. D. Santos, R. Sabouirn and P. Maupin, “Overfitting Cautions Selection of Ensembles with Genetic Algorithms,” *Information Fusion*, Vol. 10, 2009, pp. 150-162.