A Multi-objective Genetic Algorithm Based Approach for Effective Intrusion Detection Using Neural Networks

Gulshan Kumar and Krishan Kumar

Abstract In this paper, a novel multi-objective genetic algorithm (MOGA) based approach is proposed for effective intrusion detection based on benchmark datasets. The proposed approach can generate a pool of non-inferior individual solutions and ensemble solutions thereof. The generated ensembles can be used to detect the intrusions accurately. For intrusion detection problem, the proposed MOGA based approach could consider conflicting objectives simultaneously like detection rate of each attack class, error rate, accuracy, diversity etc. The proposed approach can generate a pool of non-inferior solutions and their ensemble thereof having optimized trade-offs values of multiple conflicting objectives. In this paper, a three phase MOGA based approach is proposed to generate solutions with a simple chromosome design in first phase. In first phase, a Pareto front of non-inferior individual solutions is approximated. In the second phase of the proposed approach, entire solution set is further refined to determine effective ensemble solutions considering solution interaction. In this phase, another improved Pareto front of ensemble solutions over that of individual solutions is approximated. The ensemble solutions in improved Pareto front reported improved detection results based on benchmark datasets for intrusion detection. In third phase, a combination method like majority voting method is used to fuse the predictions of individual solutions for determining prediction of ensemble solution. Benchmark datasets namely KDD cup 1999 and ISCX 2012 dataset are used to demonstrate and validate the performance of the proposed approach for intrusion detection. The proposed approach can discover individual solutions and ensemble solutions thereof with good support and detection rate from benchmark datasets (in comparison with well-known ensemble methods like bagging and boosting). In addition, the proposed approach is a generalized classification approach that is applicable to the problem of any field having multiple conflicting objectives and a dataset can be represented in the form of labeled instances in terms of its features.

G. Kumar (✉) · K. Kumar
Shaheed Bhagat Singh State Technical Campus, Ferozepur, Punjab, India
e-mail: gulshanahuja@gmail.com

K. Kumar
e-mail: k.salujasbs@gmail.com

R.R. Yager et al. (eds.), *Intelligent Methods for Cyber Warfare*,
Studies in Computational Intelligence 563, DOI 10.1007/978-3-319-08624-8_8

1 Introduction

The industry faces the challenges of a fast changing trends of attacking the Internet resources, inability of conventional techniques to protect the Internet resources from a variety of attacks, and biases of individual techniques towards specific attack class(es). Developing effecting techniques is necessary for securing valuable Internet resources from attacks. Nowadays, conventional protection techniques such as firewalls, user authentication, data encryption, avoiding programming errors and other simple boundary devices are used as the first line of defense for security of the systems. Some attacks are prevented by the first line of defense where as some bypass them. Such attacks must be detected as soon as possible so that damage may be minimized and appropriate corrective measures may be taken. Several techniques from different disciplines are being employed for the accurate intrusion detection systems (IDSs). Detection Rate (DR) and False Positive Rate (FPR) are two key indicators to evaluate the capability of an IDS. Many efforts are being done to improve DR and FPR of the IDSs [47]. In beginning, the research focus was on rule based and statistical IDSs. But, with large datasets, the results of these IDSs become un-satisfactory. Thereafter, a lot of Artificial Intelligence (AI) based techniques have been introduced to solve the problem due to their advantages over the other techniques [41, 60]. The AI based techniques have reported certain improvements in the results to detect the intrusions. Many researchers analyzed various AI based techniques empirically and compared their performance for detection of intrusions. Findings of representative empirical comparative analysis are as follows: Most of the existing techniques strive to obtain a single solution that lacks classification trade-offs [22]; Low detection accuracy and high false alarm rate; No single technique is capable enough to detect all classes of attacks to an acceptable level of false alarm rate and detection accuracy [41, 49]; Some of the existing techniques fall into local minima. For global minima, these techniques are computationally expensive; The existing techniques are not capable to model correct hypothesis space of the problem [20]; Some existing techniques are unstable in nature such as neural networks show different results with different initializations due to the randomness inherent in the training procedure; Different techniques trained on the same data may not only differ in their global performances, but they may show strong local differences also. Each technique may have its own region in the feature space where it performs the best [30]; Delay in the detection of intrusions due to the processing of a large size of high dimensional data [9, 60]; and NB, MLP and SVM techniques are found to be most promising in detecting the intrusions effectively [35]. It is also noticed from the literature of AI based techniques that most of the existing intrusion detection techniques report poor results in terms of DR and FPR towards some specific attack class(es). Even, Artificial Neural Networks (ANNs), Naive Bayes (NB) and Decision Trees (DT) have been popularly applied to intrusion detection (ID), but these techniques have provided poor results, particularly towards the minor attack class(es) [10, 31]. The poor results may be due to an imbalance of instances of a specific class(es) or the

inability of techniques to represent a correct hypothesis of the problem based on available training data.

In order to improve the low DR and high FPR, focus of the current research community in the field of intrusion detection (ID) is on ensemble based techniques. Because, there is a claim in the literature that ensemble based techniques generally outperform the best individual techniques. Moreover, several theoretical and empirical reasons including statistical, representational and computational reasons exist that also advocate the use ensemble based techniques over the single techniques [19]. This paper aims to develop a multi-objective genetic algorithm (MOGA) based approach for intrusion detection to generate a pool of non-inferior individuals solutions and combine them to generate ensemble solutions for improved detection results. The pool of solutions provides classification trade-offs to the user. Out of pool of solutions, the user can select an ideal solution as per application specific requirements.

Paper Overview: Sect. 2 presents the related work and identifies the research gaps in the field. A novel MOGA based approach for effective intrusion detection is proposed in Sect. 3. This section also explains implementation detail of the proposed approach including brief description of multi layer perceptron (MLP), benchmark datasets, performance metrics followed by experimental setup, results of the proposed approach using MLP as a base classifier. Finally, the concluding remarks along with the scope for future work are listed at the end of this paper in Sect. 4.

2 Related Work

Ensemble techniques/classifiers have been recently applied to overcome the limitations of a single classifier system in different fields [19, 34, 42]. Such attention is encouraged by the theoretical [19] and experimental [21] studies, which illustrate that ensembles can improve the results of traditional single classifiers. In general, an ensemble construction of base classifiers involves generating a diverse pool of base classifiers [6], selecting an accurate and diverse subset of classifiers [57], and then combining their outputs [42]. These activities correspond to ensemble generation, ensemble selection and ensemble integration phases of ensemble learning process [38]. Most of the existing ensemble classifiers aim at maximizing the overall detection accuracy by employing multiple classifiers. The generalizations made concerning ensemble classifiers are predominantly suitable in the field of ID. As Axelsson [4] notes, “In reality there are many different types of intrusions, and different detectors are needed to detect them”. Use of multiple classifiers is supported by the statement that if one classifier fails to detect an attack, then another should detect it [43]. However, to create an efficient ensemble, we are still facing numerous difficulties: How can we generate diverse base classifiers? Then, once these base classifiers have been generated, should we use all of them or should we select a sub-group of them? If we decide to select a subgroup, how do we go about it? Then, once the sub-group has been selected, how can we combine the outputs of these classifiers?

Previous studies in the field of intrusion detection have attempted various techniques to generate effective ensembles such as bagging, boosting, and random sub-space etc. The researchers proposed a multi classifier based system of Neural Networks (NNs) [24]. The different neural networks were trained using different features of KDD cup 1999 dataset. They concluded that a multi strategy combination technique like belief function outperforms other representative techniques. Multi classifier system of NNs was also advocated by Sabhnani and Serpen [50]. The authors reported improved results over single techniques. The researchers used weighted voting to compute the output of ensemble of CART and BN and reported improved results for intrusion detection [1, 11]. Perdisci et al. [48] proposed a clustering based fusion method that reduces the volume of alarms produced by the IDS. The reduced alarms provides a concise high level description of attacks to system administrator. The proposed method uses correlation between alarms and meta alarms to reduce the volume of alarms of the IDSs. A hierarchical hybrid system was also proposed in [61]. But, the proposed system leads to high false positive rate. Chen et al. [12] used different features of dataset to generate ensemble solutions based on evolutionary algorithms. Toosi and Kahani [56] proposed a neuro-fuzzy classifier to classify instances of KDD cup 1999 dataset into five classes. But, a great time consuming is a big problem. Hu and Damper [28] proposed a adaBoosting ensemble method that uses different features to generate diverse set of classifiers. No doubt, the proposed method reported improved performance but it suffers from limitation of incremental learning. It requires continuous retraining for changing environment. Zainal et al. [62] proposed a heterogeneous ensemble of different classifiers and used weighted voting method for combining their predictions. Wang et al. [58] proposed an approach based on NN and fuzzy clustering. Fuzzy clustering helps to generate homogeneous training subsets from heterogeneous training dataset which are further used to train NN models. They reported improved performance in terms of detection precision and stability. Clustering based hybrid system was also advocated by Muda et al. [45] for intrusion detection. The system was unable to detect the intrusions of U2R and R2L attack classes. Khreich et al. [33] proposed a iterative boolean combination (IBC) technique for efficient fusion of the responses from any crisp or soft detector trained on fixed-size datasets in the ROC space. However, IBC does not allow to efficiently adapt a fusion function over time when new data becomes available, since it requires a fixed number of classifiers. The IBC technique was further improved as incremental Boolean combination (incrBC) by the authors in [34]. The incrBC is a ROC-based system to efficiently adapt ensemble of HMM (EoHMMs) over time, from new training data, according to a learn-and-combine approach without multiple iterations. Govindarajan and Chandrasekaran [26] suggested a hybrid architecture of NNs for intrusion detection. They used weighted voting method compute the final prediction of system.

However, the models developed based on these techniques attempted to obtain a single solution. They lack in providing classification trade-offs for application specific requirements. Most of the models provided biased results towards specific attack class(es).

In contrast, genetic algorithm (GA) is the most widely used technique in data mining and knowledge discovery [23]. Applying GA is valuable for its robustness in performing global search in search space compared with other representative techniques. Several researchers employed single and multiple objective genetic algorithms for finding a set of non-inferior solutions for the problem of ID. Such initiative was carried by Parrott et al. [46] by suggesting an evaluation function which was later known as Parrot function. They proposed to use accuracy of each target class as a separate objective in their evaluation function for MOGA. Here, accuracy of each class refers to correctly classified instances of that class. The Parrot function was further adopted in [2] and [3] to generate an ensemble of base classifiers. The generation of the ensemble was completed in two stages using modified NSGA-II [18]. In the first stage, a set of base classifiers was generated. Second stage optimized the combination of base classifiers using a fixed combining method. Both of these methods differ in their function evaluation. The former study proposed to optimize the classifiers by minimizing the aggregated error of each class and maximize diversity among them. Since, the error on each class is not treated as separate objectives, this is similar to a general error measure such as MSE (mean square error), which have the same issues as the implementation of Parrot function, being biased towards the major class(es). In the second phase of the approach proposed in [2] and [3], the objectives are to minimize the size of the ensemble and maximize the accuracy. Consequently, the drawback of their approach is to create a single best solution based on general performance metrics. The same concept was further extended by Engen [22] by conducting similar experiments with different evaluation functions for creating ensemble of ANNs as base classifiers in the presence of imbalanced datasets using NSGA-II. He used 3-class classification by using ANNs and MOGA. He proved that MOGA based approach is an effective way to train the ANN which works well for minority attack classes in imbalanced datasets. He proposed two phase process for intrusion detection. In the first phase, he generated a set of base classifiers of ANNs by optimizing their weights assuming a fixed number of hidden layers and the number of neurons per hidden layer in ANN. The second phase generates improved non-dominated front of ensemble solutions based upon base ANN solutions optimized in phase 1. However, the performance of NSGA-II degrades for the real world problems having more than three objectives and large population [55].

3 MOGA Based Approach for Effective Intrusion Detection

A novel MOGA based approach for intrusion detection is proposed. The concept of two tier fitness assignment mechanism consisting of domination rank and diversity measure of solutions (as proposed in [53]) are used to improve the solutions from benchmark datasets. Generally, intrusion detection problem encounters a trade-offs between multiple conflicting criteria such as detection rate of attack classes, accuracy and diversity etc. Considering the multiple criteria of intrusion detection problem, GAs can be used in two ways. The first way to solve a multi-objective problem is

to convert multiple objectives into a single objective [13]. The single objective is further optimized by GA to produce a single solution. Generally, prior knowledge about the problem, or some heuristics guide the GA to produce a single solution. By changing the parameters of the algorithm and executing the algorithm repeatedly, more solutions can be produced. This approach has several limitations for multi objective optimization problems. The second way to solve multi objective optimization problems by using GA produces a set of non-inferior solutions. This set of non-inferior solutions represents trade-offs between multiple criteria which is identified as a Pareto optimum front [22, 39]. By incorporating domain knowledge, the user can select a desired solution. Here, GA has produced a set of solutions in Pareto front in a single run without incorporating any domain knowledge or any other heuristic about the problem. Some of the important researches in developing MOGAs are Strength Pareto Evolutionary Algorithm (SPEA2) [63], Pareto-Envelope based Selection Algorithm (PESA-II) [15], Non-dominated Sorting Genetic Algorithm (NSGA-II) [17], Archive based Micro Genetic Algorithm 2 [54] and many more. A comprehensive review of various MOGAs can be further referred in [13, 14, 16].

The proposed approach is developed with particular attention to enhance the detection rate of majority as well as minority attack class(es). A chromosome encoding scheme is proposed to represent the individual classifiers. Further more, the proposed approach is used to find an improved Pareto front consisting of ensemble solutions. The MOGA used in this paper is Archive based Micro Genetic Algorithm 2 (AMGA2) [54], which is an effective algorithm for finding optimal trade-offs for multiple criteria. AMGA2 is a generational algorithm that works with a very small population size and maintains a large external archive of good solutions obtained. Using an external archive that stores a large number of solutions provides useful information about the search space as well as tends to generate a large number of Pareto points at the end of the simulation. At every iteration, a small number of solutions are created using the genetic variation operators. The newly created solutions are then used to update the archive. The strategy used to update the archive relies on the domination level and the diversity of the solutions, and the current size of the archive, and is based on the non-dominated sorting concept borrowed from NSGA-II [18]. This process is repeated until the allowed number of function evaluations is exhausted. We used differential evolution (DE) operator as crossover operator for mating the population. Because, DE has advantage of not requiring a distribution index and it is self-adaptive in that the step size is automatically adjusted depending upon the distribution of the solutions in the search space. After mating the population with crossover operator, it is followed by mutation operator. Modified polynomial mutation operator is used to mutate the offsprings solutions.

3.1 The Proposed Approach

This section describes the proposed approach based on MOGA to create a set of base classifiers and ensembles thereof. The proposed approach follows an overproduce

and choose approach that focus on generation of a large number of base classifiers and later on choose the subset of the most diverse base classifiers to generate ensembles. The proposed approach is a three phase approach as described in subsequent paragraphs.

Phase 1 and phase 2 are multi-objective in nature and use MOGA to generate a set of base classifiers and ensembles thereof respectively. These phases of the proposed approach evolve a set of solutions to formulate diverse base classifiers and ensembles thereof using MOGA. The set of base classifiers and their ensembles exhibit classification trade-offs for the user. The diversity among individual solutions and their ensembles is maintained implicitly. The detection rate for each class is treated as a separate objective in both the phase. Here, the MOGA is real-coded, uses cross-over and mutation operators and an elitist replacement strategy.

Phase 1 of the proposed approach is capable to find the optimal Pareto front of non-dominated solutions (depicted in Fig. 1). These solutions formulate the base classifiers as candidate solutions for the ensemble generation in Phase 2. In phase 1, the values in chromosome and its size depends upon the type of base classifier and corresponding encoding scheme. The output of phase 1 is a set of optimized real values for classifiers that formulate the base classifiers of ensembles. The population size is equal to the number of desired solutions input by the user.

Phase 2 generates another improved approximation of optimal Pareto front consisting of a set of non-dominated ensembles based on a set of non-dominated solutions as base classifiers (output of phase 1) which also exhibit classification trade-offs (depicted in Fig. 2). It takes input in the form of archive of non-dominated solutions produced by phase 1 that formulates the base classifiers of the ensembles. The

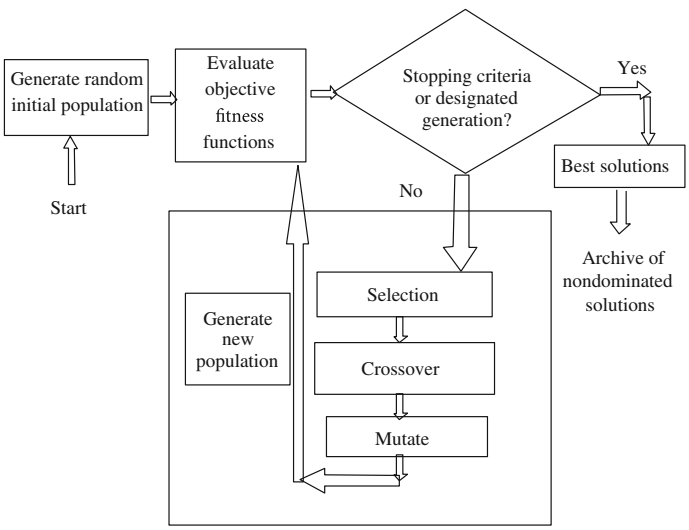

Fig. 1 Phase 1 of the proposed approach

phase evolves ensembles by combining the Pareto front of non-dominated solutions instead of the entire population like other studies [29]. Here, we are interested in those solutions which are non-inferior and exhibit classification trade-offs. The predictions of the base classifiers are combined using the 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 real coded having values from 0 to 1. Value $\geq$0.5 signifies the participation of base classifier in the ensemble and $<$0.5 signifies non-participation concerned base classifiers in creating the ensembles. The output of phase 2 is an archive of the ensembles of the base classifiers in terms of chromosomes in the range of 0 and 1 (depicted in Fig. 2). Here, value $\geq$0.5 signifies the participation of base classifier in ensemble and $<$0.5 signifies its non-participation. The set of ensembles provides the classification trade-offs for the user for different objective functions.

Phase 3 of the proposed approach integrates the predictions of base classifiers to get prediction of the final ensemble. As depicted in Fig. 3, the phase takes two inputs (1) archive of non-dominated base solutions (output of phase 1); and (2) one chromosome from the archive of ensembles as chosen by the user depending on requirements (output of Phase 2). The user may adopt static or a dynamic strategy to choose an appropriate ensemble from a pool of ensembles (evolved in Phase 2). Here in this work, we selected the ensemble classifier using a static strategy based on its performance on the training data in terms of pre-defined performance metrics. Based on the values of the chromosome, corresponding predictions of base classifiers

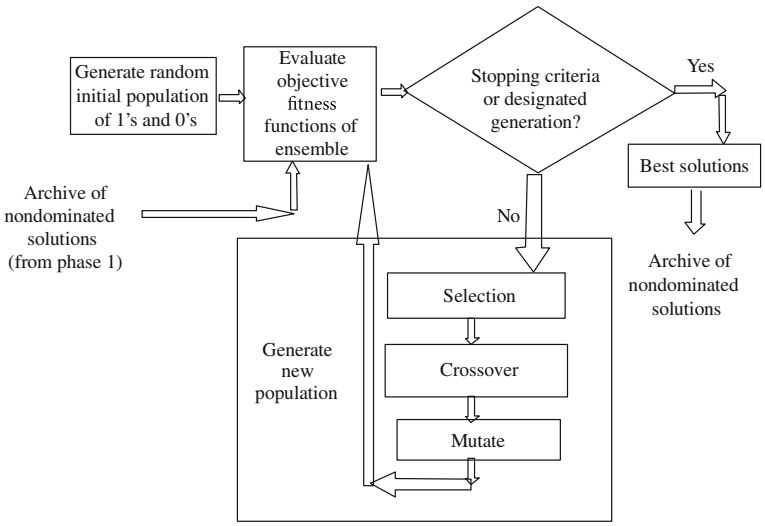

Fig. 2 Phase 2 of the proposed approach

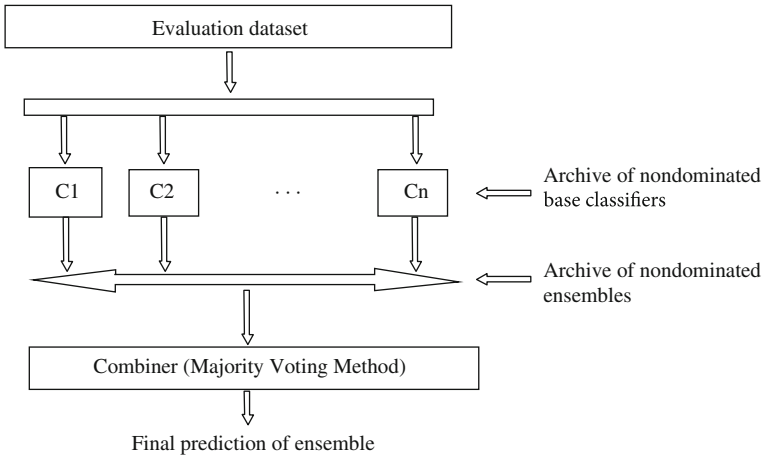

Fig. 3 Phase 3 of the proposed approach

are integrated to get a final prediction of the ensemble. In order to test the proposed approach, test dataset is directly fed to different base classifiers. Their predictions are combined in this phase to give the final output of the ensemble. In this work, we computed the final prediction of ensemble by using the majority voting method because of its popularity as depicted in Fig. 3.

The phases of the proposed approach address key issues of the current research in the field of ensembles. The issues addressed are (1) generation of a set of non-inferior solutions that exhibit classification trade-offs to formulate base classifiers of the ensemble; (2) generation of a set of non-inferior ensemble solutions that exhibit classification trade-offs; and (3) integration of predictions of the base classifiers to get final prediction of the ensemble.

3.2 Implementation

To evaluate the proposed approach, it is implemented in VC++. MLP is used as a base classifier as per finding of state of art literature in the field of ID. The performance of the proposed technique is evaluated based on benchmark datasets for ID namely KDD cup 1999 and ISCX 2012 dataset. During the optimization of multiple criteria by AMGA2, detection rate of each attack class in the dataset is used as a separate objective. Majority voting method is used to integrate the predictions of base classifiers to get prediction of the final ensemble. The results of experiments are computed on a Windows PC with Core i3-2330M 2.20 GHz CPU and 2 GB RAM. Following sub-sections describes the details of GA, MLP, benchmark dataset, and performance metrics used in the experiments.

3.2.1 Genetic Algorithm

GA are population based search techniques that have been identified to perform better that than the classical heuristics or gradient approaches [25]. GAs provides better solutions particularly for multi models, non-differentiable, or discontinuous functions. Generally, GA experiences following steps:

1. Generate a random population of individuals that represents solution to the underlying problem.
2. Evaluate the population by computing their fitness function of each individual.
3. Elevate high quality individuals by selecting them from entire population.
4. Generate new population containing individuals created by applying variation operators of cross-over and mutation.
5. Repeat the above steps till termination criteria is satisfied.

A large number of methods have been developed to implement steps for GAs. However, major issues consist of representation of individuals, fitness evaluation mechanism, variation operators of cross-over and mutation, and deciding the termination criteria.

3.2.2 Multi Layer Perceptron

An MLP is a network of simple neurons called perceptrons [5]. 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 Artificial Neural Networks (ANNs) that may be trained with the standard back propagation algorithm [5] or by using other alternative techniques. 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 MLP used in this paper is composed of three neuron layers, namely, the input layer, the output layer and the hidden layer as shown in Fig. 4. Although the MLP can have more than one hidden layer, having more than one hidden layer is rarely beneficial and can lead to gross over parametrization [22]. For a particular instance i of training/test dataset, the input layer of the MLP used for intrusion detection receives the input vector T from training dataset. The input vector T has general format

$$T_i = (t_{i,1}, t_{i,2} \ldots\ldots\ldots\ldots\ldots\ldots t_{i,n}) \tag{1}$$

Here, is the jth feature of ith instance of training/test dataset. Total number of input neurons in input layer is equal to total features of training/test dataset for intrusion

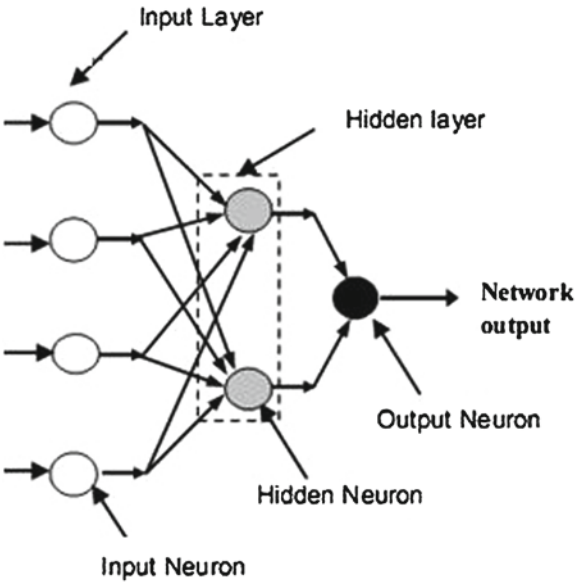

Fig. 4 Structure of MLP

detection. The output layer contains the output neurons. The output neurons are equals 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 Eq. (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

$$Y_{i,k} = f(\sum_{j=1}^{N} W_{j,k} * T_{i,j} + b_k) \tag{2}$$

where $Y_{i,k}$ is the output of kth neuron in hidden layer for ith instance of dataset, f() is an activation function, is the connection weight assigned to kth hidden neuron and jth neuron in input layer and is the bias of kth hidden neuron. In literature, many activation functions are proposed [22]. The most widely used activation functions is the sigmoid function which can be expressed as

$$Y_{i,k} = \frac{1}{1 + \exp(-\sum_{j=1}^{N} W_{j,k} * T_{i,j} - b_k)} \tag{3}$$

The neurons in output layer produce the final network output. These output neurons receives an input array in form of Eq. 4.

$$Z_i = (Y_{i,1}, Y_{i,2}.....Y_{i,n}) \tag{4}$$

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

3.2.3 Benchmark Datasets

The performance of the proposed approach is measured based on benchmark datasets. In the literature, various benchmark datasets are proposed for validation of the IDSs. As per statistics of a survey of 276 papers published between 2000 and 2008 conducted by Tavallaee [52], most of the researchers used publicly available benchmark datasets for evaluating their network based approaches. It is observed that KDD cup 1999 [32] data set is the most widely data set used for validation of an IDS [41, 52] in spite of many criticisms [7, 22, 44]. The raw training dataset contains about 4 GB of TCP connection data in the form of 5 million connection records. Similarly, test data set contains about 2 million records. KDD cup 1999 dataset utilizes TCP/IP level information and embedded with domain-specific heuristics, to detect intrusions at the network level. KDD dataset contains four major classes of attacks: Probe, Denial of Service (DoS), User-to-Root (U2R) and Remote-to-Local (R2L) attacks The labeled connection records consist of 41 features and 01 attack type. The labeled connection records consist of 22 different attack types categorized into 04 classes whereas unlabeled dataset consist of 20 known and 17 unknown attack types. The 41 features can be divided into three categories viz: Basic features of individual TCP connections, Content features within a connection suggested by domain knowledge and Traffic features computed using a two-second time window.

In a thorough study of KDD cup 1999 dataset, Tavallaee [52] observed that there are some inherent problems. He refined the KDD cup 1999 dataset and named it as NSL-KDD dataset. As the number of connection records in training and test NSL-KDD data set is very large, so it's practically very difficult to use the whole data set. Thus, in order to conduct unbiased learning and testing of the proposed approach, we used subsets of the dataset containing different proportions of normal and attack instances. The statistics of selected subsets of NSL-KDD datasets used in our experiments is as depicted in Table 1. Here, we selected 10 most prominent features in ITFS data subset by applying feature selection technique described in [36, 37].

In order to overcome the limitations of KDD cup 1999 dataset, [51] presented a new dataset for validation of an IDS at Information Security Center of eXcellence (ISCX). The dataset is available in the packet capture form. Features are extracted from the packet format by using tcptrace utility (downloaded from www.tcptrace.org) and applying the following command.

tcptrace csv −l filename1.7z > filename1.csv

Table 1 Statistics of subsets of KDD cup 1999 dataset as Training and Test data subsets

Dataset	Mode	Number of features	Class	Number of instances	Total instances
KDD 1	Training	41	Normal	1000	
			Probe	100	
			DoS	100	
			U2R	11	
			R2L	100	1311
	Test	41	Normal	500	
			Probe	75	
			DoS	75	
			U2R	50	
			R2L	50	750
KDD 2	Training	41	Normal	13449	
			Probe	2289	
			DoS	9234	
			U2R	11	
			R2L	209	25192
	Test	41	Normal	2152	
			Probe	2402	
			DoS	4342	
			U2R	200	
			R2L	2754	11850
ITFS	Training	41, 10	Normal	10000	
KDD			Probe	32316	
			DoS	23467	
			U2R	52	
			R2L	1126	66961
	Test	41, 10	Normal	5000	
			Probe	4166	
			DoS	17761	
			U2R	228	
			R2L	13448	40603

where filename is the name of the 7z (packet capture) file. From resulting csv files, we selected features which are most widely used features in the literature as proposed by Brugger [8]. The data instances including normal as well as attack instances are randomly selected to create a subset of the benchmark dataset for our experiments. The selected dataset is further preprocessed by converting discrete feature values to numeric ones as described in [40]. The statistics of selected ISCX 2012 data subset are depicted in Table 2.

Table 2 Statistics of subset of ISCX 2012 dataset as Training and Test data subset

Dataset	Mode	Number of features	Class	Number of instances	Total instances
ISCX	Training	9	Normal	4125	
2012			Attack	578	4703
	Test	9	Normal	64127	
			Attack	577	4704

3.2.4 Performance Metrics

In order to evaluate the effectiveness of the 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 the IDS [27]. Measuring this ability of the IDS is important to both industry as well as research community. It helps us to tune the IDS in a better way as well as compare different IDSs. There exist many metrics that measure different aspects of the IDS, but no single metric seems sufficient to objectively measure the capability of the IDS. Most widely used metrics by intrusion detection research community are True Positive Rate (TPR) and False Positive Rate (FPR). Or False Negative rate FNR $=$ $1 -$ TPR and True Negative Rate TNR $= 1 -$ FPR can also be used alternatively. 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 values TPR and FPR are given, then it is very difficult to determine the better IDS. To solve this problem, Gu et al. [27] proposed a new objective metric called Intrusion Detection Capability (CID) considering base rate, TPR and FPR collectively. 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) [4], Negative Predictive Value (NPV), and base rate (the probability of intrusions); (2) it objectively provides an essential measure of intrusion detection capability; and (3) it is very sensitive to IDS operation parameters such as base rate, FPR and FNR. Detail of CID can be further studied in [27]. Keeping these points in view, we computed TPR, FPR and CID to evaluate the performance of the proposed technique and compare it with other representative techniques in the field.

3.2.5 Design of Experiments

In this investigation, we used AMGA2 as a multi objective genetic algorithm because of its benefits over other representative algorithms [54]. The implementation of AMGA2 algorithm takes following input parameters.

- Number of function evaluations
- Number of desired solutions

- Random seed
- Output file

Rest of parameter like mutation rate, crossover rate, etc. is automatically tuned by the AMGA2 algorithm.

The proposed approach involves three phases to create the ensemble as described in Sect. 3.1. In phase 1 (ensemble generation phase), AMGA2 optimizes an archive of diverse base classifiers that exhibit classification trade-offs. The values in chromosome represent the weights of MLP. The size of chromosome is equal to the number of weights of MLP which is further dependent structure of the MLP (i.e. input nodes, hidden layers, number of hidden nodes per layer and output nodes). Each chromosome represents a MLP classifier in terms of its weights. The output of phase 1 is a set of optimized real values of the weights of MLPs that formulate the base classifiers for the ensembles. In phase 2 (ensemble selection phase), AMGA2 is again used to create an archive of the ensembles that also exhibits classification trade-offs. In phase 3 (ensemble integration phase), the predictions of selected base classifiers are combined to compute the final prediction of the ensemble using the majority voting method. The parameters used as input by the user to AMGA2 are depicted in Table 3. Other simulation parameters tuned automatically by AMGA2 for KDD cup 1999 dataset and the ISCX 2012 dataset are presented in Tables 4 and 5 respectively. For investigation of MLP as a base classifier, the structure of MLP used is as depicted in Table 6.

Table 3 Parameters of AMGA2 input by the user

Number of function evaluations	25000
Number of desired solutions	100
Random seed	0.1

Table 4 Simulation parameters tuned by AMGA2 for KDD cup 1999 dataset

Parameter	Value
Maximum allowed size of archive	Number of desired solutions input by the user
Size of initial population	Number of desired solutions input by the user
Size of working population	20
Maximum number of function evaluations	Number of function evaluations input by the user
Probability of crossover	0.1
Probability of mutation	0.01
Index for crossover	0.5
Index for mutation	15

Table 5 Simulation parameters tuned by AMGA2 for ISCX 2012 dataset

Parameter	Value
Maximum allowed size of archive	Number of desired solutions input by the user
Size of initial population	Number of desired solutionsinput by the user
Size of working population	8
Maximum number of function evaluations	Number of function evaluations input by the user
Probability of crossover	0.1
Probability of mutation	0.111111
Index for crossover	0.5
Index for mutation	15

Table 6 Configuration of MLP

Input nodes	Number of features of dataset
Hidden layer	1
Number of hidden nodes	30
Output nodes	5

3.3 Results and Discussion

Here, for investigation of MLP as a base classifier, ensemble generation is done by using random initial values of the weights of MLPs. As an output of this phase, we obtained an archive of MLP having optimized values of their weights. In the ensemble selection phase, we selected the MLP classifiers for the final ensemble based upon their performance during the training process (overproduce-and-choose strategy). Finally, the ensemble integration phase involves fusion strategies to combine the predictions of the selected classifiers. We used majority voting method to solve the purpose for its popularity.

In our experiments, we selected the solution for comparison with the other classifiers having a better value of the CID. Alternate solutions from the pool may provide different values of performance metrics. The results of the proposed intrusion detection approach using MLP as a base classifier and the other representative techniques are computed based upon benchmark datasets in terms of confusion matrices and other defined performance metrics. We computed average DR, Average FPR, CID and DR of each target class from the confusion matrices. The representative techniques used in this investigation are MLP trained with back propagation method, their ensembles using bagging and boosting. We utilized WEKA software package [59] to compute the results of MLP trained with back propagation, its ensembles (bagging and boosting). We used default parameters of WEKA for computing the results using MLP and its ensembles.

3.3.1 Results of KDD Cup 1999 Dataset

The proposed approach is applied to various data subsets of KDD cup 1999 dataset that produces a set of non-inferior MLP based ensemble solutions. The performance of ensemble solutions for training and test data of KDD 1 dataset is depicted in Fig. 5.

The performance of ensemble solutions for training and test data of KDD 2 dataset is shown in Fig. 6. The performance of ensemble solutions for training and test data of ITFS-KDD (41 features and 10 features) data subsets is portrayed in Figs. 7 and 8 respectively.

The overview of the classification results of KDD subsets obtained with MLP and its ensembles (using bagging and boosted methods) and our proposed approach (AMGA2-MLP) with respect to different evaluation criteria is as shown in Table 7.

The results indicate that MLP and its ensembles using bagging and boosting demonstrate comparable performance. But, these techniques are more biased towards majority classes and reported poor performance for the minority classes like U2R and

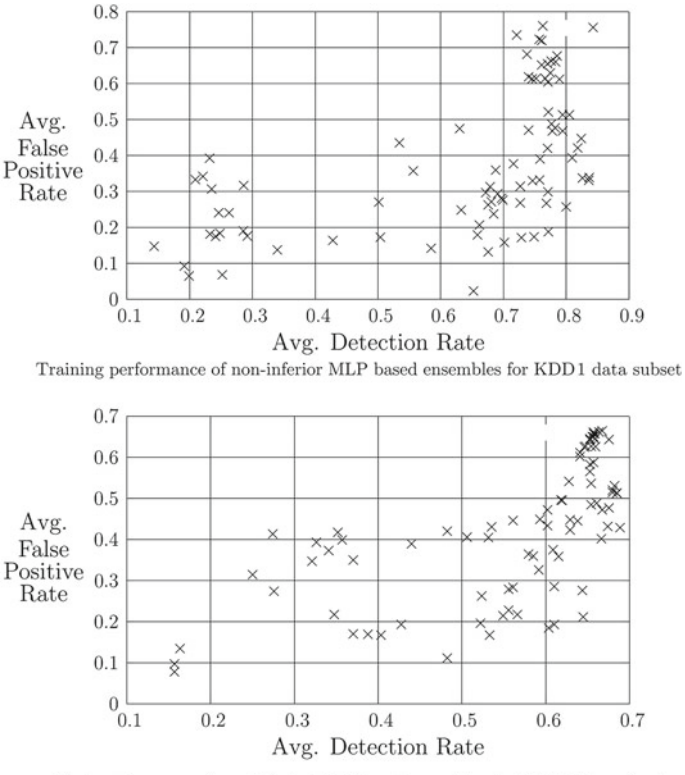

Fig. 5 Training and Test performance of non-inferior MLP based ensembles for KDD 1 data subset

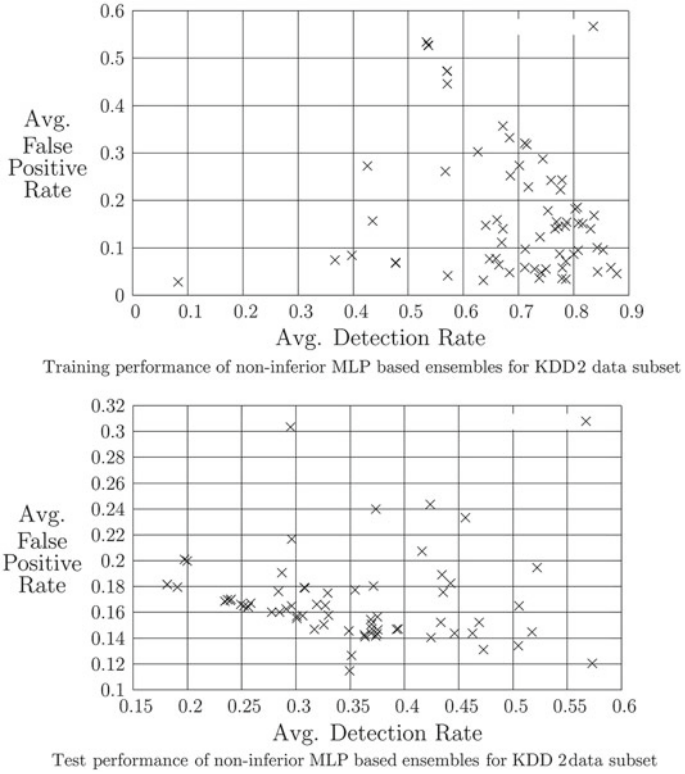

Fig. 6 Training and Test performance of non-inferior MLP based ensembles for KDD 2 data subset

R2L. MLP trained with our proposed approach is less biased and reported improved results than others for minority as well as majority classes. In case of KDD1 data subset, AMGA2-NB improved the detection of R2L attack class up to 52 % which was detected up to 2 % by the MLP and boosted MLP and 6 % by bagging based ensemble of MLP. Similarly, detection of U2R attack class is also enhanced by 66 % than MLP and its conventional ensemble techniques. In case of KDD2 data subset, MLP and its ensembles based upon bagging and boosting fails to detect U2R and R2L attack classes whereas AMGA2-MLP reported the detection of U2R and R2L attack classes up to 16.5 and 68.5 % respectively. Whereas, detection of the other classes is comparable with the other conventional ensemble techniques. In case of other data sets, the performance of the proposed technique is also comparable to the other representative techniques. Higher values of CID of our proposed technique revealed in Table 7 (in most of the cases) indicate that it outperformed the other techniques considered in this investigation.

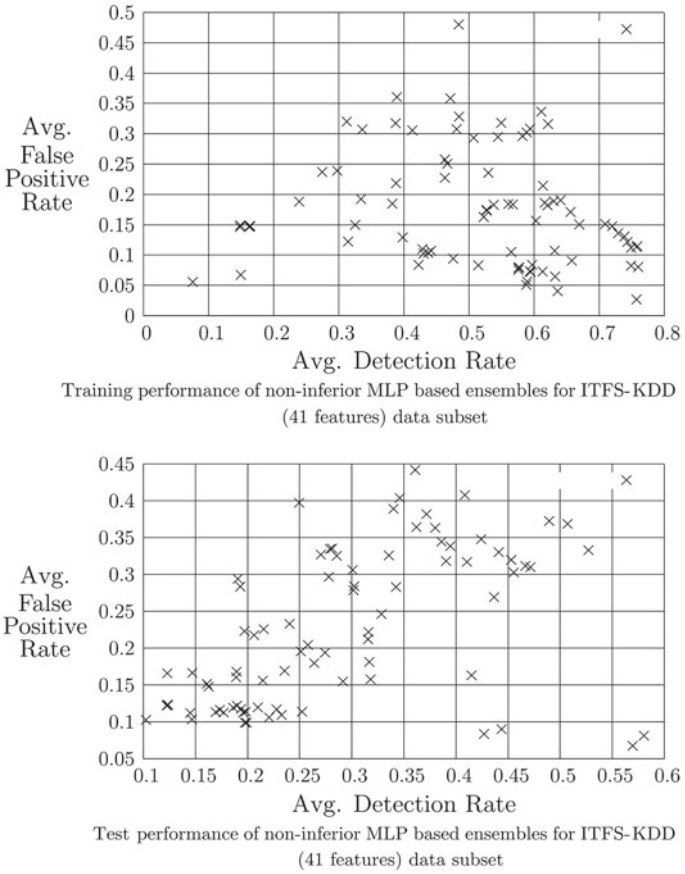

Fig. 7 Training and Test performance of non-inferior MLP based ensembles for ITFS (41 features) data subset

3.3.2 Results of ISCX 2012 Dataset

The performance of ensemble solutions for training and test data of ISCX 2012 dataset is depicted in Fig. 9. The detection results of the techniques are presented for the subset of ISCX 2012 dataset in Table 8. It can also be observed from the reporting results that AMGA2-MLP (The MLP trained with the proposed approach) reported superior performance than MLP and its bagging based ensemble and comparable performance that of boosting based ensemble of MLP. AMGA2-MLP reported the detection of normal and attack classes upto 96.9 and 97.7 % respectively. Higher value of CID indicates that our proposed approach outperformed the other techniques for the ISCX 2012 dataset considered in this investigation.

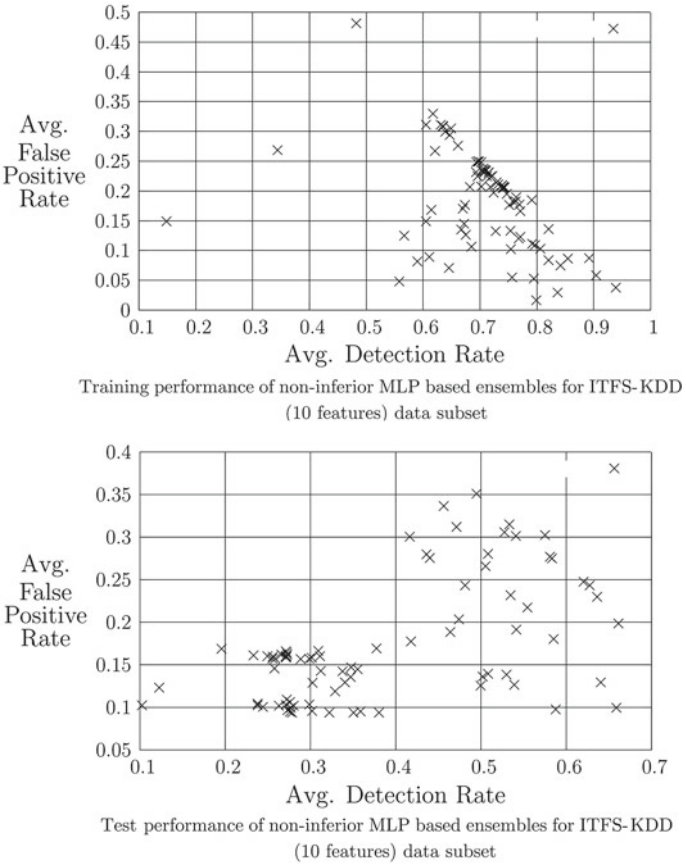

Fig. 8 Training and Test performance of non-inferior MLP based ensembles for ITFS (10 features) data subset

3.3.3 Discussion

The results obtained in this paper highlight clearly the benefits of training the MLP and its ensembles by using the proposed multi-objective genetic algorithm based approach. The proposed technique helps to improve the detection results especially for minority attack classes than that of other conventional ensemble approaches. The percentage improvement of the results of the proposed approach over other approaches is depicted in Table 9. The reporting results indicate that the proposed approach helps to enhance the average detection rate, reduce average false positive rate and overall increase in CID values over the other approaches.

In case of KDD cup 1999 dataset, MLP trained with the proposed approach helps to enhance the detection of minority attack classes like U2R and R2L attack classes which was very poorly detected by MLP trained using back propagation method.

Table 7 Overview of classification results of KDD cup 1999 subsets using MLP as a base classifier

Dataset	Technique	Avg. DR	Avg. FPR	CID	Normal	Probe	DoS	U2R	R2L
KDD1	MLP	0.720	0.374	0.086	0.898	0.627	0.507	0.100	0.020
	Bagged-MLP	0.736	0.334	0.116	0.894	0.827	0.520	0.020	0.060
	Boosted-MLP	0.720	0.374	0.086	0.898	0.627	0.507	0.100	0.020
	AMGA2-MLP	0.645	0.212	**0.142**	0.726	0.760	0.000	0.760	0.520
KDD2	MLP	0.447	0.135	0.073	0.897	0.453	0.526	0.000	0.000
	Bagged-MLP	0.435	0.138	0.066	0.908	0.414	0.507	0.000	0.000
	Boosted-MLP	0.447	0.135	0.073	0.897	0.453	0.526	0.000	0.000
	AMGA2-MLP	0.573	0.120	**0.143**	0.566	0.697	0.456	0.165	0.685
ITFS-KDD	MLP	0.510	0.071	0.134	0.971	0.958	0.360	0.136	0.405
(41 features)	Bagged-MLP	0.437	0.437	0.000	0.000	0.000	1.000	0.0000	0.000
	Boosted-MLP	0.510	0.071	0.134	0.971	0.958	0.360	0.136	0.405
	AMGA2-MLP	0.570	0.068	**0.170**	0.835	0.911	0.299	0.031	0.733
ITFS-KDD	MLP	0.580	0.202	0.087	0.859	0.818	0.652	0.101	0.314
(10 features)	Bagged-MLP	0.591	0.192	0.097	0.863	0.876	0.679	0.013	0.294
	Boosted-MLP	0.580	0.202	0.087	0.859	0.818	0.652	0.101	0.314
	AMGA2-MLP	0.659	0.099	**0.199**	0.744	0.836	0.884	0.118	0.284

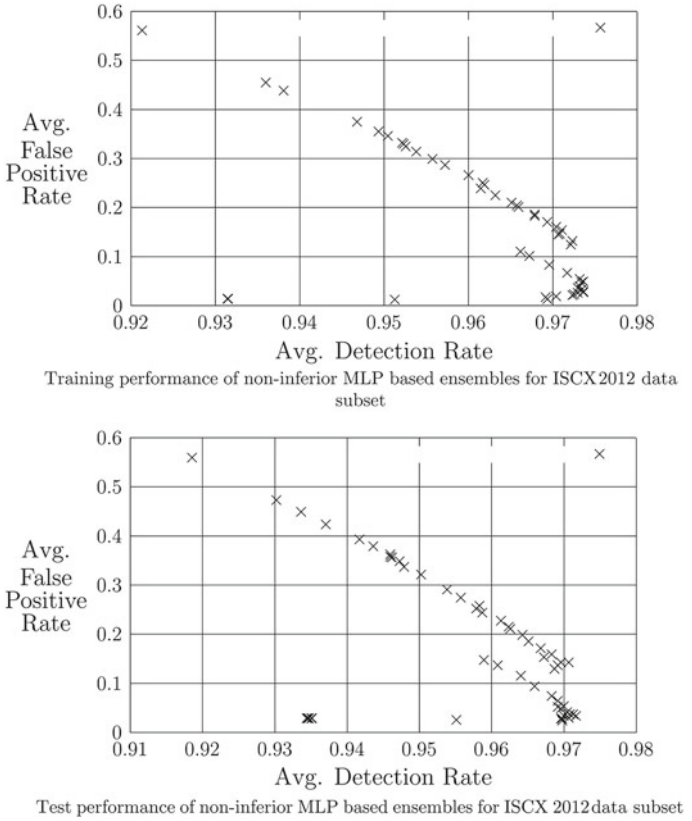

Fig. 9 Training and Test performance of non-inferior MLP based ensembles for ISCX 2012 data subset

Table 8 Overview of classification results of ISCX 2012 subset using MLP as a base classifier

Dataset	Technique	Avg. DR	Avg. FPR	CID	Normal	Attack
ISCX 2012	MLP	0.906	0.660	0.049	0.998	0.248
	Bagged MLP	0.906	0.660	0.049	0.999	0.246
	Boosted MLP	0.947	0.083	0.560	0.952	0.912
	AMGA2-MLP	0.970	0.024	**0.778**	0.969	0.977

MLP and its ensemble based on the conventional techniques like bagging and boosting are biased towards majority classes, so reported poor results for minority attack classes. Whereas, the findings of the proposed approach are that they are less biased towards majority attack classes. Thus, the proposed approach is applicable where there is class imbalance and detection of all classes especially minority attack classes is equally important, as expected in many application domains including intrusion

Table 9 Percentage improvement of the results of the proposed approach using MLP as a base classifier

Classifier	MLP			Bagged	MLP		Boosted	MLP	
Dataset	DR	FPR	CID	DR	FPR	CID	DR	FPR	CID
KDD1	−10.42	−43.32	65.12	−12.36	−36.53	22.41	−10.42	−43.32	65.12
KDD2	28.19	−11.11	95.89	31.72	−13.04	116.67	28.19	−11.11	95.89
ITFS 41	11.76	−4.23	26.87	30.43	−84.44	–	11.76	−4.23	26.87
ITFS 10	13.62	−50.99	128.74	0.12	−0.48	1.05	13.62	−50.99	128.74
ISCX 2012	7.06	−96.36	1487.76	7.06	−96.36	1487.76	2.43	−71.08	38.93

detection. It is observed from the literature that MLPs trained with back propagation methods are often used for classification tasks as they are universal approximation algorithms. But, the results of this investigation indicate that back propagation method and other similar methods for training are not appropriate in all scenarios especially where detection of majority as well as minority attack classes is equally important. In case of ISCX 2012 dataset, results similar to KDD cup 1999 dataset are also obtained. MLP trained with back propagation method and its bagging based ensemble demonstrated poor results for detection of attack class. Whereas, AMGA2-MLP enhanced average DR to 0.97 (0.906 in case of MLP) and reduced average FPR to 0.024 (0.66 in case of MLP) approximately. It is also observed that most of the conventional techniques provide a single solution and lacks in providing classification trade-offs. Whereas, the proposed approach provides a pool of solutions to the problem. Out of this pool, the user can select any one solution based on its better value for CID and his/her application specific requirements. Other solutions with different values of CID may offer different detection results for the same problem that helps to exhibit the different classification trade-offs. Hence, the results depicted above sections proved the superiority of the proposed multi-objective genetic algorithm based approach and validated its applicability for proper training of the MLP for intrusion detection.

In a nut shell, the empirical investigation and comparison of the results indicate the following:

- The proposed approach outperforms the individual representative techniques in terms of identified performance metrics.
- There are indications in the literature that bagging and boosting learn better from imbalanced data. However, the experiments here have demonstrated that these algorithms remain biased towards the majority class(es).
- Using MLP as a base classifier, the proposed approach is able to enhance DR by 28 % , reduce FPR by 51 % approximately over the results of MLP trained using back propagation method and its ensemble using boosting technique based on KDD cup 1999 dataset. However, an improvement of results is noticed upto 30 % in DR and 84 % in FPR approximately over bagging based ensemble of MLP for KDD cup 1999 dataset. For ISCX 2012 dataset, the results of the proposed

technique are improved upto 7 % in DR and 96 % in FPR approximately over MLP and its ensemble using bagging technique.
- The ensembles evolved with the proposed technique provides better solutions, and also achieves a higher detection accuracy.
- Higher values of CID for the proposed technique proved the superiority over the existing individual techniques and their ensembles using bagging and boosting.
- The proposed approach is capable to produce a pool of solutions that address the limitations of the existing techniques, striving to obtain a single solution in which there is no control on classification trade-offs (for application specific requirements).
- The proposed approach is a generalized classification approach that is applicable to the problem of any field having multiple conflicting objectives and a dataset can be represented in the form of labeled instances in terms of its features.

4 Concluding Remarks

In this paper, a novel multi objective genetic algorithm based approach is proposed for effective intrusion detection. The proposed approach is capable of producing a pool of non inferior individual solutions and ensemble solutions thereof which exhibit classification trade-offs for the user. By using certain heuristics or prior domain knowledge, a user can select an ideal solution as per application specific requirements. The proposed approach attempts to tackle the issues of low DR, high FPR and lack of classification trade-offs in the field of ID. The proposed approach consists of encoding of chromosomes that provides optimized values of weights of MLPs. AMGA2 is employed to build multi objective optimization model that generates individual solutions and ensemble solutions thereof with simultaneous consideration of detection rate of each attack class in the dataset. A three phased multi-objective genetic algorithm based approach can rapidly generate numerous individual solutions and ensemble solutions thereof with simple chromosome design in first phase of the proposed approach. The entire solutions are further refined to obtain ensemble solutions in second phase of the approach. The predictions of individual solutions are fused together to compute final prediction of the ensemble using majority voting method in phase 3 of proposed approach.

Benchmark datasets namely KDD cup 1999 and ISCX 2012 dataset for intrusion detection are used to demonstrate and validate the performance of the proposed approach based on MLP as a base classifier. The proposed approach can discover an optimized set of individual MLPs and ensemble of MLPs thereof with good support and detection rate from benchmark datasets (in comparison with well-known ensemble methods like bagging and boosting). The optimized set of MLPs and ensemble of MLPs exhibit the classification tradeoffs for the users. The user may select an ideal solution as per application specific requirements. Using MLP as a base classifier, the proposed approach is able to enhance DR by 28 % , reduce FPR by 51 % approximately over the results of MLP trained using back propagation method and

its ensemble using boosting technique based on KDD cup 1999 dataset. However, an improvement of results is noticed upto 30 % in DR and 84 % in FPR approximately over bagging based ensemble of MLP for KDD cup 1999 dataset. For ISCX 2012 dataset, the results of the proposed technique are improved upto 7 % in DR and 96 % in FPR approximately over MLP and its ensemble using bagging technique. Higher values of CID for the proposed approach proved the superiority over the existing individual techniques and their ensembles using bagging and boosting.

The major issue in the proposed approach is that it takes long time to compute fitness functions in various generations. It may be overcome by computing the function values in parallel. Here, we computed the results by limiting the population size and number of generations of MOGA. More experiments may be conducted by using different values of these parameters. The proposed approach is validated using small subsets of benchmark datasets only, whereas its applicability can be tested by conducting more experiments with real network traffic in the field of ID. The proposed approach utilized static method for selecting an appropriate ensemble solution whereas dynamic selection method may lead to more fruitful results.

References

1. Abraham, A., Thomas, J.: Distributed intrusion detection systems: a computational intelligence approach. Applications of Information Systems to Homeland Security and Defense, pp. 105–135. Idea Group Inc., Publishers, USA (2005)
2. Ahmadian, K., Golestani, A., Analoui, M., Jahed, M.: Evolving ensemble of classifiers in low-dimensional spaces using multi-objective evolutionary approach. In: Proceedings of 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS), pp. 217–222. IEEE (2007)
3. Ahmadian, K., Golestani, A., Mozayani, N., Kabiri, P.: A new multi-objective evolutionary approach for creating ensemble of classifiers. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics (ISIC), pp. 1031–1036. IEEE (2007)
4. Axelsson, S.: Intrusion detection systems: a survey and taxonomy. Technical report (2000)
5. Bishop, C.: Pattern Recognition and Machine Learning, vol. 4. Springer, New York (2006)
6. Breiman, L.: Bias, variance, and arcing classifiers (technical report 460). Department of statistics. University of California at Berkeley (1996)
7. Brown, C., Cowperthwaite, A., Hijazi, A., Somayaji, A.: Analysis of the 1999 darpa/lincoln laboratory ids evaluation data with netadhict. In: Proceedings of IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), pp. 1–7. IEEE (2009)
8. Brugger, S.: Data mining methods for network intrusion detection. University of California at Davis (2004). www.citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.3127&rep=rep1&type=pdf
9. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. (CSUR) **41**(3), 15 (2009)
10. Chawla, N.: C4. 5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In: Proceedings of the ICML Workshop on Learning from Imbalanced Datasets II, vol. 3 (2003)
11. Chebrolu, S., Abraham, A., Thomas, J.: Feature deduction and ensemble design of intrusion detection systems. Comput. Secur. **24**(4), 295–307 (2005)
12. Chen, Y., Abraham, A., Yang, B.: Hybrid flexible neural-tree-based intrusion detection systems. Int. J. Intell. Syst. **22**(4), 337–352 (2007)

13. Coello, C.: An updated survey of ga-based multiobjective optimization techniques. ACM Comput. Surv. (CSUR) **32**(2), 109–143 (2000)
14. Coello, C., et al.: A comprehensive survey of evolutionary-based multiobjective optimization techniques. Knowl. Inf. syst. **1**(3), 129–156 (1999)
15. Corne, D., Jerram, N., Knowles, J., Oates, M., et al.: Pesa-ii: Region-based selection in evolutionary multiobjective optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001). Citeseer (2001)
16. Deb, K.: Multi-objective optimization. Multi-objective Optimization using Evolutionary Algorithms, pp. 13–46. Wiley, New York (2001)
17. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. Lect. Notes Comput. Sci. **1917**, 849–858 (2000)
18. Deb, K., Anand, A., Joshi, D.: A computationally efficient evolutionary algorithm for real-parameter optimization. Evol. Comput. **10**(4), 371–395 (2002)
19. Dietterich, T.: Ensemble methods in machine learning. Multiple Classifier Systems, pp. 1–15. Springer, Heidelberg (2000)
20. Dietterich, T., Bakiri, G.: Error-correcting output codes: a general method for improving multiclass inductive learning programs. In: Proceedings of Santa fe Institute Studies in the Sciences of Complexity, vol. 20, pp. 395–395. Citeseer (1994)
21. Dos Santos, E.M.: Static and dynamic overproduction and selection of classifier ensembles with genetic algorithms. Ph.D. thesis, Montreal (2008)
22. Engen, V.: 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 from imbalanced data. Ph.D. thesis, Bournemouth University (2010)
23. Fung, K., Kwong, C., Siu, K., Yu, K.: A multi-objective genetic algorithm approach to rule mining for affective product design. Expert Syst. Appl. **39**(8), 7411–7419 (2012)
24. Giacinto, G., Roli, F.: An approach to the automatic design of multiple classifier systems. Pattern Recogn. Lett. **22**(1), 25–33 (2001)
25. Giannopoulos, N., Moulianitis, V., Nearchou, A.: Multi-objective optimization with fuzzy measures and its application to flow-shop scheduling. Eng. Appl. Artif. Intell. **25**, 1381–1394 (2012)
26. Govindarajan, M., Chandrasekaran, R.: Intrusion detection using neural based hybrid classification methods. Comput. Netw. **55**(8), 1662–1671 (2011)
27. Gu, G., Fogla, P., Dagon, D., Lee, W., Skorić, B.: Measuring intrusion detection capability: An information-theoretic approach. In: Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, pp. 90–101. ACM (2006)
28. Hu, R., Damper, R.: A no panacea theorem for classifier combination. Pattern Recogn. **41**(8), 2665–2673 (2008)
29. Ishibuchi, H., Nojima, Y.: Evolutionary multiobjective optimization for the design of fuzzy rule-based ensemble classifiers. Int. J. Hybrid Intell. Syst. **3**(3), 129–145 (2006)
30. Jain, A., Duin, R., Mao, J.: Statistical pattern recognition: a review. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 4–37 (2000). doi:10.1109/34.824819
31. Jo, T., Japkowicz, N.: Class imbalances versus small disjuncts. ACM SIGKDD Explor. Newsl. **6**(1), 40–49 (2004)
32. KDD: Kdd cup 1999 dataset (1999). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
33. Khreich, W., Granger, E., Miri, A., Sabourin, R.: Iterative boolean combination of classifiers in the roc space: an application to anomaly detection with hmms. Pattern Recogn. **43**(8), 2732–2752 (2010)
34. Khreich, W., Granger, E., Miri, A., Sabourin, R.: Adaptive roc-based ensembles of hmms applied to anomaly detection. Pattern Recogn. **45**(1), 208–230 (2012)
35. Kumar, G., Kumar, K.: Ai based supervised classifiers: an analysis for intrusion detection. In: Proceedings of International Conference on Advances in Computing and Artificial Intelligence, pp. 170–174. ACM (2011)

36. Kumar, G., Kumar, K.: A novel evaluation function for feature selection based upon information theory. In: Proceedings of 24th Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 000,395–000,399. IEEE (2011)
37. Kumar, G., Kumar, K.: An information theoretic approach for feature selection. Secur. Commun. Networks **5**(2), 178–185 (2012). doi:10.1002/sec.303
38. Kumar, G., Kumar, K.: The use of artificial-intelligence-based ensembles for intrusion detection: a review. Appl. Comput. Intell. Soft Comput. **2012**, 1–20 (2012). doi:10.1155/2012/850160
39. Kumar, G., Kumar, K.: The use of multi-objective genetic algorithm based approach to create ensemble of ann for intrusion detection. Int. J. Intell. Sci. **2**(24), 115–127 (2012). doi:10.4236/ijis.2012.224016
40. Kumar, G., Kumar, K., Sachdeva, M.: An empirical comparative analysis of feature reduction methods for intrusion detection. Int. J. Inf. Telecommun. Technol. **1**(1), 44–51 (2010)
41. Kumar, G., Kumar, K., Sachdeva, M.: The use of artificial intelligence based techniques for intrusion detection: a review. Artif. Intell. Rev. **34**(4), 369–387 (2010)
42. Kuncheva, L.I.: Combining pattern classifiers: methods and algorithms (kuncheva, li; 2004)[bibbookreview]. IEEE Trans. Neural Netw. **18**(3), 964–964 (2007)
43. Lee, W., Stolfo, S., Mok, K.: Adaptive intrusion detection: a data mining approach. Artif. Intell. Rev. **14**(6), 533–567 (2000)
44. McHugh, J.: Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. ACM Trans. Inf. Syst. Secur. **3**(4), 262–294 (2000)
45. Muda, Z., Yassin, W., Sulaiman, M., Udzir, N., et al.: A k-means and naive bayes learning approach for better intrusion detection. Inf. Technol. J. **10**(3), 648–655 (2011)
46. Parrott, D., Li, X., Ciesielski, V.: Multi-objective techniques in genetic programming for evolving classifiers. In: Proceedings of IEEE Congress on Evolutionary Computation, vol. 2, pp. 1141–1148. IEEE (2005)
47. Patcha, A., Park, J.M.: An overview of anomaly detection techniques: existing solutions and latest technological trends. Comput. Netw. **51**(12), 3448–3470 (2007). doi:10.1016/j.comnet.2007.02.001. http://www.sciencedirect.com/science/article/pii/S138912860700062X
48. Perdisci, R., Giacinto, G., Roli, F.: Alarm clustering for intrusion detection systems in computer networks. Eng. Appl. Artif. Intell. **19**(4), 429–438 (2006)
49. Re, M., Valentini, G.: Integration of heterogeneous data sources for gene function prediction using decision templates and ensembles of learning machines. Neurocomputing **73**(7–9), 1533–1537 (2010)
50. Sabhnani, M., Serpen, G.: Application of machine learning algorithms to kdd intrusion detection dataset within misuse detection context. In: Proceedings of International Conference on Machine Learning: Models, Technologies, and Applications, vol. 1, pp. 2009–215 (2003)
51. Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput. Secur. **31**(3), 357–374 (2012)
52. Tavallaee, M.: An adaptive hybrid intrusion detection system. Ph.D. thesis, University of new brunswick (2011)
53. Tiwari, S.: Development and integration of geometric and optimization algorithms for packing and layout design. Ph.D. thesis, Clemson University (2009)
54. Tiwari, S., Fadel, G., Deb, K.: Amga2: improving the performance of the archive-based micro-genetic algorithm for multi-objective optimization. Eng. Optim. **43**(4), 377–401 (2011)
55. Tiwari, S., Koch, P., Fadel, G., Deb, K.: Amga: an archive-based micro genetic algorithm for multi-objective optimization. In: Proceedings of Genetic and Evolutionary Computation conference (GECCO-2008), Atlanta, USA, pp. 729–736 (2008)
56. Toosi, A.N., Kahani, M.: A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. Comput. Commun. **30**(10), 2201–2212 (2007). doi:10.1016/j.comcom.2007.05.002. http://www.sciencedirect.com/science/article/pii/S0140366407001855

57. Tsoumakas, G., Angelis, L., Vlahavas, I.: Selective fusion of heterogeneous classifiers. Intell. Data Anal. **9**(6), 511–525 (2005)
58. Wang, G., Hao, J., Ma, J., Huang, L.: A new approach to intrusion detection using artificial neural networks and fuzzy clustering. Expert Syst. Appl. **37**(9), 6225–6232 (2010)
59. Witten, I., Frank, E., Hall, M.: Data Mining: Practical Machine Learning Ttools and Techniques. Morgan Kaufmann, San Francisco (2011)
60. Wu, S., Banzhaf, W.: The use of computational intelligence in intrusion detection systems: a review. Appl. Soft Comput. **10**(1), 1–35 (2010)
61. Xiang, C., Yong, P., Meng, L.: Design of multiple-level hybrid classifier for intrusion detection system using bayesian clustering and decision trees. Pattern Recogn. Lett. **29**(7), 918–924 (2008)
62. Zainal, A., Maarof, M., Shamsuddin, S., et al.: Ensemble classifiers for network intrusion detection system. J. Inf. Assur. Secur. **4**, 217–225 (2009)
63. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. Evol. Comput. **8**(2), 173–195 (2000)