DOI:10.24996.ijs.2017.58.1C.16

ISSN: 0067-2904

A Multi-Objective Evolutionary Algorithm based Feature Selection for Intrusion Detection

Dhuha I. Mahmood*, Sarab M. Hameed

Department Computer Science, College of Science, University of Baghdad, Baghdad, Iraq.

Abstract

Nowad ays, with the development of internet communication that provides many facilities to the user leads in turn to growing unauthorized access. As a result, intrusion detection system (IDS) becomes necessary to provide a high level of security for huge amount of information transferred in the network to protect them from threats. One of the main challenges for IDS is the high dimensionality of the feature space and how the relevant features to distinguish the normal network traffic from attack network are selected. In this paper, multi-objective evolutionary algorithm with decomposition (MOEA/D) and MOEA/D with the injection of a proposed local search operator are adopted to solve the Multi-objective optimization (MOO) followed by Naïve Bayes (NB) classifier for classification purpose and judging the ability of the proposed models to distinguish between attack network traffic and normal network traffic. The performance of the proposed models is evaluated against two baseline models feature vitality based reduction method (FVBRM) and NB. The experiments on network security laboratory-knowledge discovery and data mining (NSL-KDD) benchmark dataset ensure the ability of the proposed MOO based models to select an optimal subset of features that has a higher discriminatory power for discriminating attack from normal over the baselines models. Furthermore, the proposed local search operator ensures its ability to harness the performance of MOO model through achieving an obvious feature reduction on average from 16.83 features to 8.54 features (i.e., approximately 50%) in addition to the increase in NB classifier accuracy from 98.829 to 98.859 and detection rate from 98.906 to 99.043.

Keywords: Feature selection, Intrusion Detection, Naïve Bayes, Multi-objective evolutionary algorithm.

اختيار الميزة المعتمد على الخوارزمية التطورية متعددة الاهداف لكشف التطفل

ضحى عماد محمود*، سراب مجيد حميد

قسم علوم الحاسبات ، كلية العلوم، جامعة بغداد، بغداد، العراق.

الخلاصة

في الوقت الحاضر، مع تطور الاتصالات عبر الانترنيت والتي تقدم العديد من التسهيلات للمستخدم يؤدي ذلك بدوره الى تزايد الوصول غير المصرح به. ونتيجة لذلك، اصبح نظام كشف التطفل ضروري لتوفير مستوى عالي من الأمن لكمية كبيرة من المعلومات المنقولة في الشبكة لحمايتها من التهديدات. واحدة من التحديات الرئيسية لكشف التطفل هي الأبعاد العالية من فضاء الميزة وكيفية تحديد الميزات ذات الصلة لتمييز حركة المرور الطبيعية على الشبكة من الهجوم. في هذا البحث، اعتمدت الخوارزمية التطورية متعددة

*Email: doha89emad@gmail.com

الاهداف مع التحلل (MOEA/D) و (MOEA/D) مع حقن مشغل البحث المحلي المقترح لحل مشكلة امثلية تعدد الاهداف يليه المصنف نيف بايز (NB) لغرض التصنيف والحكم على قدرة النماذج المقترحة للتمييز بين حركة المرور الطبيعية على الشبكة من الهجوم. اداء النماذج المقترحة تم تقييمه بالمقارنة مع نموذجين من النماذج الاساسية وهي (FVBRM) و NB. تضمن التجارب على البيانات القياسية (NSL−KDD) قدرة النماذج المقترحة المعتمدة على امثلية تعدد الاهداف على اختيار امثل مجموعة فرعية من الميزات التي لديها اعلى طاقة تمييزية لتمييز الهجوم من الطبيعي بالمقارنة مع النماذج الاساسية. وعلاوة على ذلك، ان مشغل البحث المحلي المقترح يضمن قدرته على الاستفادة من اداء نموذج امثلية تعدد الاهداف الذي حقق تقليل واضح للميزات بمعدل من 16.83 الى 8.54 ميزة (اي مايقارب 50%) بالأضافة الى زيادة دقة مصنف نيف بايز (NB) من 98.829 الى 98.859 ومعدل الكشف من 98.906 الى 99.043.

Introduction

Nowadays, computer network has been widely used in many aspects of human life and network security is becoming more essential to all companies and institutions using the Internet. Intrusion detection is an active network security protection method which can collect and analyze network traffic data, aiming to find out whether there are malicious internal behaviors violating security policy and signs of being attacked by external intruders [1].

Up to the moment, researchers focus on building IDS that is capable of detecting attacks in different environments. However, several problems should be considered when designing IDS one of them is the high dimensionality of the feature space and how the relevant features to distinguish the normal network packet from attack network packet are selected. This challenge will be stated in the presented paper while considering the following:

- *How the irrelevant feature can be avoided*.
- *How the performance of the proposed intrusion detection model can be improved*.

The problem of considering relevant features for intrusion detection is modeled as a multi-objective optimization (MOO) problem and Evolutionary Algorithm with Decomposition (MOEA/D) is adopted for this purpose. In addition, a local search is proposed and inserted into multi-objective evolutionary algorithm to harness its strength. Moreover, the Naïve Bayes NB classifier is applied to judge the ability of the proposed model to classify normal and attack network traffics.

The organization of the paper is as follows. Section 2 describes the related work. Section 3 presents a brief description of the basic concepts for multi-objective optimization, and Multi-objective evolutionary algorithm based on decomposition (MOEA/D). Section 4 introduces the proposed Multi-Objective Evolutionary Algorithm (MOEA) models for intrusion detection. Section 5 evaluates the proposed models. Finally, section 6 concludes the work and hints some research future line.

Related Work

In literature, the intrusion detection has been addressed as a classification problem with different approaches. Some of these approaches are as follows:

Chi-Ho Tsang et al. in (2006) [2] presented a model for intrusion detection that uses genetic algorithm (GA) and multi-objective optimization to extract IF–THEN rules from network traffic packet. Testing of the model was performed on KDD-Cup99 benchmark dataset. In addition, the classification algorithms namely NB, C4.5, support vector machine (SVM) and K-nearest neighbor (KNN) and the feature selection techniques including best first (BF), forward sequential selection (FSS), backward sequential selection (BSS) and GA were used for comparison purpose. Results clarified that the presented model provides better detection rate and low false alarm rate with minimum number of features.

Gomathy and Lakshmipathi in (2011) [3] proposed a model for intrusion detection that uses GA as a wrapper feature selection and back propagation neural network (BPN) for classification purpose. Testing of the model was performed on KDD-Cup99 benchmark dataset. Results showed that the proposed model could improve the accuracy of intrusion detection significantly after feature selection.

Mukherjee and Sharma in (2012) [4] proposed a model for intrusion detection based on new feature reduction method named feature vitality based reduction method (FVBRM) and NB for classification purpose. The proposed model is evaluated and compared against three standard feature selection methods including: correlation based feature selection, information gain and gain ratio. The

experimental results on network security laboratory-knowledge discovery and data mining (NSL-KDD) showed significant improvements to the accuracy result of intrusion detection.

Ahmad et al. in (2013) [5] presented a model for intrusion detection that uses GA to search the principal component analysis (PCA) space to choose a subset of principal components followed by SVM for classification purpose. Testing of the model was performed on KDD-Cup99 benchmark dataset and a comparison was made with traditional method that selects specific percentage of the top principal components. Results demonstrated that the proposed model has the ability to improve the detection rate and reduce the number of features.

Malik et.al in (2013) [6] purposed a multi-objective particle swarm optimization (PSO) for feature selection and random forests (RF) for classification to model intrusion detection. Testing of the model was performed on KDD-Cup99 benchmark dataset and a comparison was made against the classification techniques including BayesNet, Bagging, Jrip, and NBTree with four feature selection techniques namely information gain (IG), correlation-based feature selection (CFS), consistency based feature selection, and symmetrical uncertainty. Results demonstrated that the proposed model outperforms in most of the cases.

Hota and Shrivas in (2014) [7] introduced four different feature selection techniques: IG, CFS, Relief and symmetrical uncertainty followed by C4.5 decision tree technique to model intrusion detection. The proposed model was compared against classification and regression tree (CART), iterative dichotomizer 3(ID3), RepTree and decision table methods using NSL-KDD dataset. Results showed that better results could be obtained from a combination of the IG and SVM than other techniques.

Eesa et.al in (2014) [8] presented a cuttlefish algorithm (CFA) based feature selection and decision tree (DT) to classify a network traffic into normal and attack. The proposed model was evaluated using KDD-Cup99. The results illustrated that the proposed model provides better detection rate and lower FAR in comparison to the obtained results with all features.

Aslahi-Shahri et al. in (2015) [9] introduced a GA for minimizing the number of features and support vector machine (SVM) for classifying the network packet as normal or attack. KDD-Cup99 benchmark dataset was used as an evaluation dataset and as an evaluation metrics, detection rate and false alarm rate were used. Experimental results showed that the intrusion detection model based on GA and SVM has the ability to provide a better detection rate and lower false positive rate when compared with other techniques.

Kumar and Kumar in (2015) [10] proposed a multi-objective genetic algorithm (MOGA) namely archive based micro genetic algorithm 2 (AMGA2) and a multi-layer perceptron (MLP) as a classifier to model intrusion detection. To evaluate and validate the proposed model, two datasets namely KDD-Cup99 and ISCX 2012 were used. The results showed that the proposed model could produce high detection rate when compared with bagged MLP and boosted MLP techniques.

Thaseen and Kumar in (2016) [11] presented an intrusion detection model based on linear discriminant analysis (LDA) and chi square to recognize the optimal subset of features followed by a modified NB for classification purpose. The experimental results applied to NSL-KDD dataset indicated that the proposed model produced better detection rate and lower FAR when compared to the discriminative multi nominal Naïve Bayes and traditional Naïve Bayes.

Kanakarajan and Muniasamy in (2016) [12] presented three feature selection techniques including information gain, symmetrical uncertainty and CFS followed by a greedy randomized adaptive search procedure with annealed randomness Forest (GAR-Forest) for both binary and multi-class classification. NSL-KDD benchmark dataset was used as an evaluation dataset and detection rate, accuracy and false alarm rate were used as an evaluation metrics. Experimental results showed that the GAR-forest performs better when compared with random forest, C4.5, NB and multilayer perceptron for binary and multi-class classification problem.

Basic Concepts for Multi-objective Optimization

Many real-world problems involve optimizing simultaneously several conflicting objectives. For Multi-Objective Optimization Problems (MOPs), instead of a single optimum which defines the optimal solution in a single objective optimization problems, there is a set of alternative trade-offs that represent the set of optimal solutions for the problem regarding all the objectives in it that are not dominated by any solution in the search space [13]. The best solution be determined by the need of the

designer or decision maker [14] [15]. Multi-objective optimization problem can be formulated as in equation 1 [14]:

$$minimize\ or\ maximize\ F(x) = [f_1(x), f_2(x), \dots \dots f_m(x)]^T$$
$$subjected\ to\ x \in \Omega \quad (1)$$

Where:

x: is an n-dimensional decision variable vector.

Ω: contains all possible x that can be used to satisfy an evaluation of $F(x)$.

m: is the number of objective functions to be optimized simultaneously.

Optimality Concept in MOPs

Definition 1 (Pareto dominance [16]**)**

Let $x, y \in \Omega$, x dominates y (denoted by $x \prec y$) if and only if, $f_i(x) \geq f_i(y)$ and $f_i(x) > f_i(y)$ in at least one f_i, $\forall i = 1, \dots, m$.

Definition 2 (Pareto optimal [16]**,** [17]**)**

x^* is called Pareto optimal solution, if there is no other solution x such that $x \prec x^*$. In other words, the definition means that x^* is Pareto optimal if there exists no feasible vector x that would make an increase some measure without causing at the same time a decrease in at least one other measure.

Definition 3 (Pareto optimal set [16]**)**

The Pareto optimal set (PS) is defined by:

$PS = \{x \in \Omega | \text{x is a Pareto optimal solution}\}$

The Pareto optimal set comprises from the globally optimal solutions (i.e., all the global noninferior solutions). In MOPs , it is possible to have local Pareto optimal sets such as the case of single-objective optimization, where local optima can exist in the search space [18].

Definition 4 (Pareto optimal front [16]

The Pareto front (PF) is defined by:

$PF = \{F(x) = (f_1(x), \dots \dots, f_m(x)) | x \in PS\}$

Multi-objective evolutionary algorithm based on decomposition (MOEA/D)

Multi-objective evolutionary algorithms (MOEAs) have been proven to be well suited for complex MOPs with two or three objectives. Through simulating the basic principles of the evolutionary process on a set of individuals, these algorithms can cope with multi-objective problems [19].

Multi-objective evolutionary algorithm based on decomposition (MOEA/D) presented by Zhang and Li is one of the popular algorithms for multi-objective optimization problems [20]. The following points illustrate the basic idea of MOEA/D [20], [21], [22], [23]:

1. In MOEA/D, the MOP is decomposed into a number of Single-Objective Optimization Problems (SOPs) by using a decomposition approach.
2. Each non-dominated solution of MOP is equivalent to SOP optimal solution that is determined by a certain weight vector.
3. In MOEA/D, optimization of each subproblem takes in consideration the information from its neighboring subproblems.

Several approaches are existed to obtain a number of scalar optimization from the approximation problem of the PF. These are Tchebycheff approach and weighted sum approach. In this paper, Tchebycheff approach is adopted due to its popularity.

Tchebycheff approach is used to decompose a MOP into number of scalar optimization problems as in Equation (2) [20] [22].

$$minimize\ g^{te}(x|\lambda, z^*) = \max_{1 \leq i \leq m}\{\lambda_i |f_i(x) - z_i^*|\} \quad (2)$$
$$subjected\ to\ x \in \Omega$$

Where

x : is the variables to be optimized.

λ : is a weighted vector, i.e.,$\forall i = 1, \dots, m$: $\lambda_i \geq 0$ and $\sum_{i=1}^{m} \lambda_i = 1$.

z^* : is a reference point, $z^* = (z_1^*, \dots, z_m^*)^T$.

For each Pareto optimal point x^* there exists a weight vector λ such that x^* is the optimal solution of (2) and each optimal solution of (2) is a Pareto optimal solution of (1). Hence, changing the weight vector leads to obtain different Pareto optimal solutions [20].

Multi-objective Evolutionary Algorithm for Intrusion Detection

The main challenge in developing a model for intrusion detection is the process of detecting distinguished features that may recognize normal network traffic from attack network traffic. The proposed intrusion detection model takes in consideration the problem of high dimensionality of the feature space through introducing two models based on multi-objective optimization using MOEA/D namely ($MOEA$ and Multi-Objective Evolutionary Algorithm with Local Search $MOEA-LS$). The $MOEA-LS$ model based on injection a local search into MOEA/D is proposed to make an improvement on the performance of $MOEA$ model. For the classification of the network traffic, Naïve Bayes classifier is adopted.

Feature Selection Stage

The aim of feature selection is to pick up a set of distinguishing features that can describe the NSL-KDD dataset in a similar or a more effective way than the whole set of features. NSL-KDD dataset can be formally described as:

$\mathbb{D} = \{D_1, D_2, \dots, D_{nt}\}$,

Where

nt is the total number of network traffic in D.

Each network traffic $D_k \in D$ can be expressed as follows:

$\forall k \in \{1, \dots, nt\}$

$D_k = \{d_{k1}, d_{k2}, \dots, d_{k42}\}$

d_{k42}: determines the status of network traffic as either normal or attack as formulated in what follow:

$$d_{k42} = \begin{cases} 1 \; if \; D_k is \; an \; attack \\ 0 \; otherwise \end{cases}$$

The process of selecting a relevant feature set, $\mathcal{F}$, out of the whole set of 41 features, or in other words, if $\mathrm{F} = \{\mathrm{F}_1, \mathrm{F}_2, \dots, \mathrm{F}_{41}\}$ is the whole feature set, then, $\mathcal{F} \subseteq \mathrm{F}$, is carried out through proposing two models: model based on MOEA/D, and model based on MOEA/D with local search operator.

MOEA/D algorithm is adopted to solve the optimization problem of intrusion detection. Representation of an individual is considered as a vector of fixed-length, where each gene denotes the existence or absence of the corresponding feature (i.e. gene value is either 0 or 1).

At each generation of MOEA/D, First, genetic operators including selection, crossover and mutation are applied: in selection operator, two parents I^{p1} and I^{p2} are selected randomly from the neighbors of individual I . Next, uniform crossover is applied to these parents with probability p_c to produce new individual, I'. Next, flip mutation is applied to each allele in I' with probability p_m to yield new individual, I''.

Second, reference points $z^* = (z_1^*, z_2^*)$ are updated such that $\forall j = \{1,2\}$, z_j^* is set to the largest value of multi-objective achieved so far. Third, the neighbors of I'' are updated according to Tchebycheff approach as mentioned in Equation 2. Finally, EP is updated by eliminating all solutions dominated by I'' and addition of I'' to EP is performed if it is non-dominated by any other solution in the archive.

Multi-Objective Model Based Feature Selection

Intrusion detection problem involves simultaneous optimization of two contradictory objective functions. Therefore, two MOO models are introduced to get the best trade-off between the contradictory objectives. In MOO, a set of Pareto optimal solutions is obtained instead of one optimal solution as in SOO.

The first model ($MOEA$) aims to find a good trade-off between true negative rate and detection rate. It involves two objective functions: the first objective $F_{TNR}(I)$ concerns true negative rate that to be maximized, while the second objective concerns detection rate $F_{DR}(I)$ and should also be maximized. $MOEA_2$ is formulated as in Equation (3):

$$Maximize \; MOEA = [F_{TNR}(I), \; F_{DR}(I)]^T \quad (3)$$

$$F_{TNR}(I) = \frac{TN}{TN + FP} \times 100$$

$$F_{DR}(I) = \frac{TP}{TP + FN} \times 100$$

Where

True positive (TP) denotes intrusion that is correctly classified,
True negative (TN) denotes normal traffic that is correctly classified,
False positive (FP) denotes normal traffic misclassified as an intrusion, and
False negative (FN) denotes intrusion misclassified as normal traffic.
$F_{TNR}(I)$ means the percentage of normal traffic that is correctly classified by NB, over the NSL-KDD training dataset, $\mathbb{D}$. While $F_{DR}(I)$ means the percentage of intrusion that is correctly classified by NB, over the NSL-KDD training dataset $\mathbb{D}$.

In the second model ($MOEA-LS$), a local search operator is proposed to improve the performance of MOEA/D. It is injected in MOEA/D to refine the set of non-dominated solutions founded in EP at each generation. The basic idea for the proposed local search operator is to remove feature that its existence has little impact on the performance classification process.

For each non-dominated solution $P^{optimal}$ in EP, the proposed local search operator evaluates the vitality of feature included in $P^{optimal}$ according to accuracy and detection rate (DR). If feature corresponds to j^{th} gene exists (i.e., gene value=1) in $P^{optimal}$, then the j^{th} gene value is set to zero to discover if the performance tend to be improved through its absence or existence in the solution. If the presence feature is with low impact on the solution, then it should be removed from $P^{optimal}$. Otherwise, it should be preserved. This process is continued for all features (i.e., gene value=1) that exist in $P^{optimal}$. Algorithm (1) highlights the detailed steps of local search operator. The detailed steps of the proposed MOEA/D with local search is presented in Algorithm 2.

Algorithm 1: Local Search Operator

Input:

- $\boldsymbol{EP}$: External Population.
- $\boldsymbol{n=41}$: Individual Length
- $\mathbb{D}$**:** NSL-KDD Dataset

Output:

- $\boldsymbol{EP^*}$**:** Enhanced External Population

1: $\boldsymbol{For}\ i = 1\ to\ |EP|$

2: **Invoke NB classifier on $\mathbb{D}$with nondominated solution $\boldsymbol{P_i^{optimal}}$ included in $\boldsymbol{EP}$ to return two evaluation metrics:$\boldsymbol{Acc}$ and $\boldsymbol{DR}$.**
$[Acc, DR] = NB\ (\mathbb{D}, P_i^{optimal})$

3: **Remove feature presence from nondominated solution, $\boldsymbol{P_i^{optimal}}$**
$\forall j \in \{1, \dots, n\}, \wedge\ {P^{optimal}}_{ij} = 1$
Turn off j^{th} gene, ${P^{optimal}}_{ij}$=0
$\boldsymbol{P^{optimal'}} = P_i^{optimal}$

4: **Evaluate the impact of j^{th} feature by invoking NB classifier on $\mathbb{D}$ with new $\boldsymbol{P^{optimal'}}$**
$[Acc', DR'] = NB\ (\mathbb{D}, P^{optimal'})$

5: **Check the detection rate and the accuracy of $\boldsymbol{P^{optimal'}}$ if they are out of a threshold limit**
$\boldsymbol{If}\ Acc' >= Acc\ \&\&\ DR >= DR'$
$Acc = Acc'$
$DR = DR'$
$\boldsymbol{Else}$
${P^{optimal}}_{ij}$=1

4: **Store $\boldsymbol{P_i^{optimal}}$ in $\boldsymbol{EP^*}$**
$EP_i^* = P_i^{optimal}$

Algorithm 2: MOEA/D with Local Search

Input:

- Multi-objective intrusion detection problem

$$Maximize\ MOEA = [F_{TNR}(I), \quad F_{DR}(I)]^T$$

- $\boldsymbol{N = 100}$**:** Total no. of subproblems
- $\boldsymbol{n}$**=41:** Individual length.
- $\boldsymbol{Max_g = 100}$**:** Maximum no. of generations
- $\boldsymbol{T = 5}$: neighborhood size
- $\mathbb{D}$**:** NSL-KDD training dataset
- $\boldsymbol{p_c}$**:** Probability of crossover

$\boldsymbol{p_m}$: Probability of mutation

Output: $\boldsymbol{EP}$**:** External Population containing non-dominated solutions

1: **Initialization**

$g = 0$

$EP = \emptyset$

$z_1^* = 0;\ z_2^* = 0$

For $i = 1\ to\ N$

$\lambda_{i1} = \frac{i}{N};\ \lambda_{i2} = \frac{N-i}{N}$

End

Generate initial population randomly, $I = \{I_1, I_2, \dots, I_N\}$

Evaluate I using $MOEA$ as in Equation (3).

Compute Euclidean distance between weight vectors $\lambda_1, \dots, \lambda_N$ and the T closest vectors $\lambda_i^1, \dots, \lambda_i^T$ are worked out for each weight vector λ_i. $\forall i = 1, \dots, N$, set $B(i) = \{i_1, .., i_T\}$.

2: **Updating**

For $i = 1\ to\ N$

3: Select randomly two indices $k, l\ \in B(i)$ to generate new solution I'' from I^k and I^l

4: Apply uniform crossover with p_c

If $rand\ \leq p_c$

For $j = 1\ to\ n$

If $rand\ \leq 0.5$

$i'_j = i_j^k$

Else

$i'_j = i_j^l$

End

Else

$i'_j = \quad i_j$

5: Apply flip mutation with p_m

For $j = 1\ to\ n$

If $rand\ \leq p_m$

$i''_j = 1 - i'_j$

Else

$i''_j = i'_j$

End

6: Evaluate new solution I'' using $MOEA(I'')$

7: Update z^*, $\forall j \in \{1,2\}$

$if\ z_1^* < F_{TNR}\ \ (I'')$, then set $z_1^* = F_{TNR}\ \ (I'')$

$if\ z_2^* < F_{DR}\ \ (I'')$, then set $z_2^* = F_{DR}\ \ (I'')$

8: **Update neighboring solutions**: $\forall j\ \in B(i)$, $if\ g(I''|\lambda_j, z^*) \leq g(I_j|\lambda_j, z^*)$, then *set* $I_j = I''$ and $MOEA = MOEA^{\ j}(I'')$

9: **Update EP**: *Remove* from EP all vectors dominated by $MOEA^j\ (I'')$.

Insert $MOEA(I'')$ to EP if no vector in EP dominate $MOEA(I'')$

10: Apply Local search as mentioned in Algorithm (1)

10: **Termination Condition**

Output **EP** if $g > Max_g$,otherwise $g = g + 1$ and go to **step 2**

Classification Stage

The role of classification stage in intrusion detection is to categorize network traffic as either normal or attack. Features resulted from feature selection stage are used as the input to this stage. Naïve Bayes classifier is adopted to judge the ability of the proposed model to classify normal and attack network traffics. NB is used according to its ability to generate the probability of features by scanning a training data only once, which makes the task of classification to be straightforward.

NB classifier involves two phases: learning phase and classification phase. The goal of learning phase is to estimate the prior probability of normal class and attack class and the probability of predictor given class (i.e., likelihood). Whereas the goal of the classification phase is to categorize the incoming network traffic into either normal or attack.

In learning phase, as clarified in algorithm 3, given NSL-KDD training dataset $\mathbb{D} = \{D_1, D_2, \dots, D_{nt}\}$ and their corresponding labels $C = \{0,1\}$, the prior probability $P(c_j)$, $c_j \in C, \forall j \in \{1,2\}$, is calculated as the frequency of network traffic belongs to c_j divided by the total number of network traffic in NSL-KDD training dataset as in Equation 4.

$$P(c_j) = \frac{\sum_{i=1}^{nt} x_i}{nt},\ c_j \in C, \forall j \in \{1,2\} \quad (4)$$

Where

$$x_i = \begin{cases} 1\ if\ c_i = c_j \\ 0\ otherwise \end{cases}$$

Whereas estimating the features distribution of the given class (i.e., likelihood) is the relative frequency of each feature value for a given class. For each feature value v_{kl}, the probability of feature value v_{kl} given class c_j is calculated as in Equation 5

$c_j \in C, \forall j \in \{1,2\}, \forall k \in \{1, \dots, 41\}, \forall l \in \{1, \dots, k_n\}$

$$P(v_{kl}|c_j) = \frac{\sum_{i=1}^{nt} y_i}{\sum_{i=1}^{nt} x_i} \quad (5)$$

Where

$$y_i = \begin{cases} 1\ if\ c_i = c_j\ and\ d_{ik} = v_{kl} \\ 0\ otherwise \end{cases}$$

$$x_i = \begin{cases} 1\ if\ c_i = c_j \\ 0\ otherwise \end{cases}$$

k_n:is the number of values in feature k.

Algorithm 3: Naïve Bayes Classifier: Learning Phase

Input:

- $\mathbb{D}$**:** NSL-KDD training dataset
- $\boldsymbol{nt}$: Number of network traffic in $\mathbb{D}$
- $\boldsymbol{C}$ **=2**: Number of classes
- $\boldsymbol{n = 41}$: Total number of features

Output:

- $\boldsymbol{P}$**:** Prior probability
- $\boldsymbol{Likelihood}$: Probability of features

1: Calculate the prior probability for each class

$\boldsymbol{For}\ j = 1\ to\ C$

$P(c_j) = \frac{\sum_{i=1}^{nt} c_i = c_j}{nt}$

2: Calculate the likelihood of feature values in $\mathbb{D}$

$\boldsymbol{For}\ k = 1\ to\ n$

$\boldsymbol{For}\ l = 1\ to\ k_n$

$P(v_{kl}|c_j) = \frac{\sum_{i=1}^{nt} c_i = c_j\ and\ d_{ik} = v_{kl}}{\sum_{i=1}^{nt} c_i = c_j}$ // **Likelihood**

$\boldsymbol{End}$

$\boldsymbol{End}$

$\boldsymbol{End}$

In classification phase, the prior probability and likelihood of each feature value resulted from the learning phase are used as the input to the classification phase. Then, for each of the network traffic in NSL-KDD testing dataset $\mathbb{T} = \{T_1, \dots, T_{ns}\}$, the posterior probability of each class $c_j \in C, \forall j \in \{1,2\}$ is computed as illustrated in Equation 6 [24].

$$P(c|x) = \frac{P(v|c)P(c)}{P(v)} \quad (6)$$

Where
$P(c|v)$ is the posterior probability of class.
$P(c)$ is the prior probability of class.
$P(v|c)$ is the probability of predictor given class.
$P(v)$ is the prior probability of predictor**.**
Moreover, the problem of unseen feature value in the training phase that leads the estimation equals to zero should be considered in the classification phase. Laplace smoothing method is utilized to tackle this problem as illustrated in Equation (7).

$$P(v_{kl}|c_j) = \frac{(\sum_{i=1}^{nt} y_i)+1}{(\sum_{i=1}^{nt} x_i)+k_n+1} \quad (7)$$

Where

$$y_i = \begin{cases} 1 \; if \; c_i = c_j \; and \; d_{ik} = v_{kl} \\ 0 \; otherwise \end{cases}$$

$$x_i = \begin{cases} 1 \; if \; c_i = c_j \\ 0 \; otherwise \end{cases}$$

Finally, after computing the two posterior probabilities, a label is assigned for the network traffic T_i. The network traffic T_i is categorized as a normal when the posterior probability of c_1 greater than the posterior probability of c_2. Otherwise, T_i will be categorized as an attack. In other words, T_i belongs to class with highest posterior probability. Algorithm (4) demonstrates the steps of the classification phase.

Algorithm 4: Naïve Bayes Classifier: Classification Phase

Input:

- $\mathbb{T}$ **:** NSL-KDD testing dataset
- $\boldsymbol{ns}$: Number of network traffics in $\mathbb{T}$
- $\boldsymbol{C}$ **=2**: Number of classes
- $\boldsymbol{n = 41}$: Total number of features
- $\boldsymbol{P}$: Prior probability of class
- $\boldsymbol{Likelihood}$: Probability of feature values for given class.

Output: Classified $\mathbb{T}$

1: **Calculate the likelihood for each network traffic** T_i **in** $\mathbb{T}$

$\boldsymbol{For}\ i = 1\ to\ ns$
$\boldsymbol{For}\ j = 1\ to\ C$
$L = 1$
$\boldsymbol{For}\ k = 1\ to\ n$
$\boldsymbol{For}\ l = 1\ to\ k_n$
$L = L\ \times P\left(v_{kl}|c_j\right)$
$\boldsymbol{End}$
$\boldsymbol{End}$

2: **Calculate the posterior probability for each class**

$Posterior(j) = P(c_j) \times L$
$\boldsymbol{End}$

3: **Assign label normal or attack to the network traffic** $\boldsymbol{T_i}$

$$T_i = \begin{cases} noraml \; if Posterior(1) > Posterior(2) \\ attack, otherwise \end{cases}$$

$\boldsymbol{End}$

Performance Evaluation

Dataset Description

The proposed model uses NSL-KDD [25] benchmark dataset as an evaluation data. It involves three different datasets: the complete dataset, 20% of the complete dataset for training and KDD full testing dataset. This paper uses 20% training dataset that contains 25192 normal and attack instances. Each instance consists of a set of 41 features and a label that marks each record as either normal or specific attack type. The features have all forms of continuous, discrete, and symbolic variables. Testing the proposed intrusion detection model is performed through applying 10-fold cross-validation approach. The total number of normal and attack instances for each fold in training and testing dataset are illustrated in Figures- 1and 2 respectively.

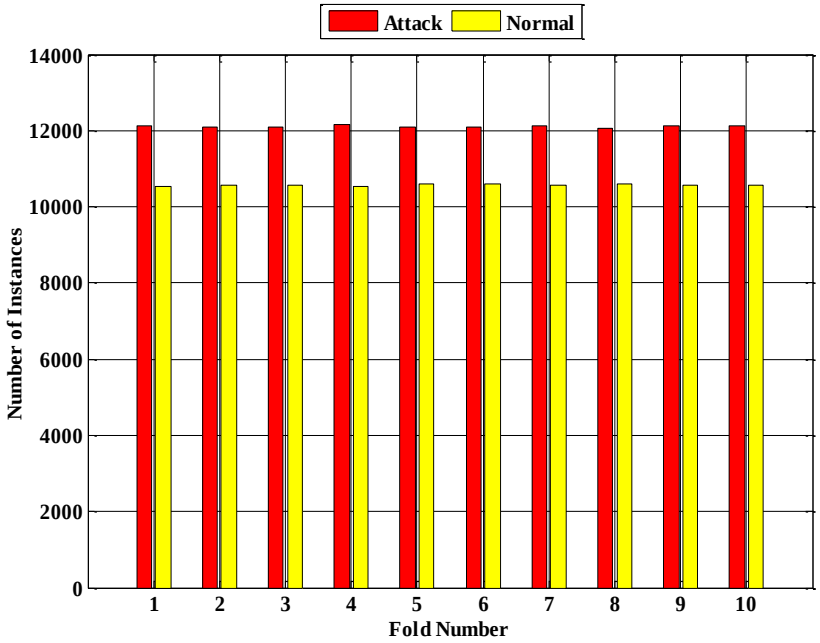

Figure 1- Total number of normal and attack instances for each fold in the training dataset.

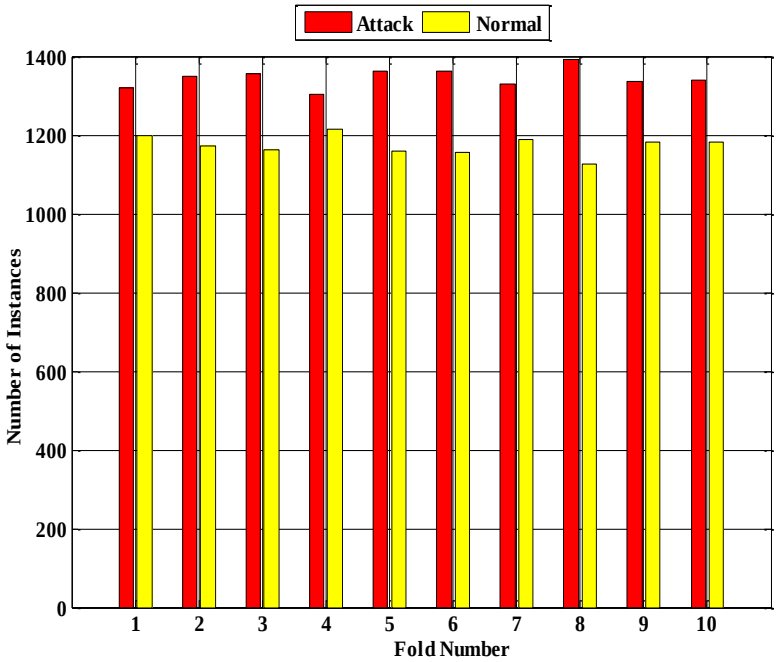

Figure 2- Total number of normal and attack instances for each fold in the testing dataset

Evaluation Metrics

To evaluate the performance of the proposed intrusion detection model the following criteria are used [26]:

1. **Accuracy (Acc)** measures the ratio of correctly classified connection to the total number of connections.
2. $Acc = \frac{TP+TN}{TP+TN+FP+FN}$ (8)
3. **Detection Rate (DR)** measures the ratio of correctly detected attacks to the total number of attacks.

$DR = \frac{TP}{TP+FN}$ (9)

False Alarm Rate (FAR) measures the ratio of normal connections that are misclassified as attack to the total number of normal connections.

$$FAR = \frac{FP}{TN+FP} \quad (10)$$

Results and Discussions

Several experiments for each fold in NSL-KDD testing dataset were conducted in order to assess the effectiveness of the proposed multi-objective optimization based models $MOEA$, and $MOEA-LS$ for intrusion detection. The optimization models are solved using multi-objective evolutionary algorithm with decomposition. The obtained results are represented by an average of different 10 runs with the same $MOEA$ setting parameters as in the following:

1. Population size $N = 100$.
2. Neighborhood size $T = 5$.
3. Maximum number of generations$Max_g = 100$.
4. Crossover probability $p_c = 0.9$.
5. Mutation probability $p_m = 0.1$.

The experiments are conducted under 10 runs using 10 folds with the same MOEA parameters setting and the NB classifier uses the set of Pareto optimal solutions is obtained by MOEA. Table- 1 and Table- 2 report the average accuracy (Acc'), average detection rate (DR'), average false alarm rate (FAR'), and average number of selected features ($F_{Selected}$) for the $MOEA$ and $MOEA$ with local search ($MOEA-LS$) operator models over 10 runs for each fold.

Table 1- Acc',DR' , FAR' and $F_{Selected}$ of $MOEA$ for ten folds

Fold #	Acc'%	DR'%	FAR	$F_{Selected}$
1	98.646	98.817	0.01509	17.5
2	98.937	99.086	0.01193	16.3
3	98.925	98.839	0.01002	18
4	98.761	98.337	0.00844	19.5
5	98.829	98.843	0.01183	15.4
6	98.797	98.547	0.00990	17
7	98.543	99.134	0.01986	16.5
8	98.996	99.246	0.01207	16.4
9	99.043	99.281	0.01167	14.4
10	98.817	98.933	0.01286	17.3

T**able 2-** Acc',DR' FAR' and $F_{Selected}$ of $MOEA-LS$ for ten folds

Fold #	$\boldsymbol{Acc'}$%	$\boldsymbol{DR'}$%	$\boldsymbol{FAR'}$	$\boldsymbol{F_{Selected}}$
1	98.845	99.042	0.01334	9.6
2	98.849	99146	0.01408	8
3	98.992	99.097	0.01098	9.2
4	98.992	99.399	0.01388	8.1
5	98.845	98.955	0.01249	6
6	98.797	98.503	0.00954	10.7
7	98.491	99.126	0.02077	9.3
8	99.047	99.201	0.01078	8
9	99.043	99.146	0.01047	6.5
10	98.690	98.815	0.01420	10

Table-3 report the performance comparison in terms of overall average accuracy $\overline{Acc}$, average detection rate $(\overline{DR})$, average false alarm rate $(\overline{FAR})$, and average number of selected features $(\overline{F_{Selected}})$ among the proposed models against the baseline model: $FVBRM$ [4] and NB with all features (i.e.,NB-41). In addition, the values of standard deviation of overall average accuracy, detection rate and false alarm rate are revealed in bold.

Table 3- Comparison of proposed models against FVBRM and $NB-41$

Model	$\boldsymbol{\overline{Acc}}$%	$\boldsymbol{\overline{DR}}$%	$\boldsymbol{\overline{FAR}}$	$\boldsymbol{\overline{F_{Selected}}}$
$NB-41$	97.122 **0.28833**	96.191 **0.44139**	0.02067 **0.002818**	41
FVBRM [4]	98.114 **0.29424**	97.725 **0.36630**	0.01546 **0.00600**	15.2
$MOEA$	98.829 **0.04011**	98.906 **0.09602**	0.01237 **0.00129**	16.83
$MOEA-LS$	98.859 **0.03001**	99.043 **0.04809**	0.01305 **0.00071**	8.54

The results observed that the proposed MOO based models, $MOEA$ and $MOEA-LS$ outperform the baseline models in terms of accuracy, detection rate and false alarm rate. The result show that the capability of the $MOEA-LS$ model to reduce the number of features to 8.54 and increase the accuracy and DR to 98.859 % and 99.043 % respectively. Furthermore, the results illustrate that there is a less deviation (as presented in bold) in the proposed models than the baseline models. This means that the proposed models trained using any nine folds acquire adequate information to realize balancing among number of features, detection rate and accuracy if the tenth fold is used for testing.

Conclusions

In this paper, an intrusion detection is molded as a multi-objective optimization problem (MOO) to select an optimal subset of feature and NB for classification purpose is proposed. The results reveal that the proposed MOO based models $MOEA$ and $MOEA-LS$ ensure their ability to select an optimal subset of features that has a higher discriminatory power for discriminating attack from normal against the baselines models. The proposed local search operator has a positive effect of on the performance of $MOEA$ model. This achieves an obvious averaged feature reduction from 16.83 features to 8.54 features (i.e., approximately 50%) in addition to the increase of NB classifier accuracy from 98.829 to 98.859 and detection rate from 98.906 to 99.043. In the future work, the proposed intrusion detection models designed to distinguish the normal network packet from attack network packet. These models can be developed to classify the attack network packet based on attack categories.

References

1. Wang, Y. Xu, S. and Huang, Q. **2015**. A Novel Evaluation Approach to Finding Lightweight Machine Learning Algorithms for Intrusion Detection in Computer Network. *International Journal of Network Security & Its Applications* (*IJNSA*), **7**: 1-13.
2. Tsang, C. H., Kwong, S. and Wang, H. **2007**. Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. *Pattern Recognition*, **40:** 2373 – 2391.
3. Gomathy, A. and Lakshmipathi, B. **2011**. *Network Intrusion Detection Using Genetic Algorithm and Neural Network*. In Advances in Computing and Information Technology, Springer-Verlag Berlin Heidelberg, pp. 399–408.
4. Mukherjee, S. and Sharma, N. **2012**. Intrusion Detection using Naive Bayes Classifier with Feature Reduction. *Procedia Technology*, **4:** 119 – 128.
5. Ahmad, I. and Hussain, M. **2014**. Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components. *Neural Computing and Applications*, **24**: 1671-1682.
6. Malik, A. J. and Khan, F. A. **2013**. A Hybrid Technique using Multi-objective Particle Swarm Optimization and Random Forests for PROBE Attacks Detection in a Network., in 2013 IEEE International Conference on Systems, Man, and Cybernetics,
7. Hota, H. S. and Shrivas, A. K. **2014**. *Decision Tree Techniques Applied on NSL-KDD Data and Its Comparison with Various Feature Selection Techniques*. In Advanced Computing, Networking and Informatics-Volume 1, Springer International Publishing, pp: 205-211.
8. Eesa, A. S., Orman, Z. and Brifcani, A. M. **2015**. A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. *Expert Systems with Applications*, **42**: 2670-2679.
9. Aslahi-Shahri,B. M., Rahmani, R., Chizari, M., Maralani, A., Eslami, M., Golkar, M. J. and Ebrahimi, A. **2015**. A hybrid method consisting of GA and SVM for intrusion detection system. *Neural Computing and Applications* : 1-8.
10. Kumar, G. and Kumar, K. **2015**. *A Multi-objective Genetic Algorithm Based Approach for Effective Intrusion Detection Using Neural Networks*. In Intelligent Methods for Cyber Warfare, Springer International Publishing, pp. 173-200.
11. Thaseen, I. S. and Kumar, C. A. **2016**. Intrusion Detection Model Using Chi Square Feature Selection and Modified Naïve Bayes Classifier. In Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC – 16'),,
12. Kanakarajan, N. K. and Muniasamy, K. **2015**. Improving the Accuracy of Intrusion Detection Using GAR-Forest with Feature Selection. In Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015, 2016.
13. Jaimes, A. L. and Coello, C. A. C. **2008**. An Introduction to Multi-Objective Evolutionary Algorithms and some of Their Potential Uses in Biology. *Applications of Computational Intelligence in Biology*, : 79-102,
14. Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S. and Coello, C. A. C. **2014**. A Survey of Multi-Objective Evolutionary Algorithms for Data Mining: Part-I. *IEEE Transactions on Evolutionary Computation*, **18**: 4-19.
15. Branke, J., Deb, K., Miettinen, K. and Słowi´nski, R. **2008**. *Multiobjective optimization: Interactive and evolutionary approaches*. Lecture Notes in Computer Science, Springer-Verlag Berlin, **5252,** pp: 27-57.
16. Martínez, S. Z. **2013**. Use of Gradient-Free Mathematical Programming Techniques to Improve the Performance of Multi-Objective Evolutionary Algorithms. Ph.D. Thesis, Center for Research and Advanced Studies of the National Polytechnic Institute of Mexico, June
17. Coello, C. A. C,. Lamont, G. B. and Veldhuizen, D. A. V. **2007**. *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2nd, Ed., Springer Science & Business Media,
18. Montaño, A. A. **2012**. Design of Multi-Objective Evolutionary Algorithms for Aeronautical Problems. Ph.D. Thesis, Research and Study Center Advanced from The National Polytechnical Institute.

19. Lücken, C. V., Barán, B. and Brizuela, C. **2014**. A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications*, **58**: 707-756.
20. Zhang, Q. and Li, H. **2007**. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on evolutionary computation*, **11**: 712-731.
21. Chen, C. M., Chen, Y. P. and Zhang, Q. **2009**. Enhancing MOEA/D with Guided Mutation and Priority Update for Multi-objective Optimization. In 2009 IEEE Congress on Evolutionary Computation.
22. Guo, X. F., Wang, X. and Wei, Z. **2015**. MOEA/D with Adaptive Weight Vector Design. In 2015 11th International Conference on Computational Intelligence and Security (CIS).
23. Zhou, A,. Qu, B. Y. Li, H., Zhao, S. Z., Suganthan, P. N. and Zhang, Q. **2011**. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, **1:** 32–49.
24. Desale, K. S., Kumathekar, C. N. and Chavan, A. P. **2015**. Efficient Intrusion Detection System using Stream Data Mining Classification Technique. In Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on.
25. The NSL-KDD Dataset [Online]. http://iscx.ca/NSL-KDD/.
26. Wu, S. X. and Banzhaf, W. **2010**. The Use of Computational Intelligence in Intrusion Detection Systems: A Review. *Applied Soft Computing*, **10**: 1-35.