Improved Intrusion Detection Algorithm based on TLBO and GA Algorithms

Mohammad Aljanabi[1,2] and MohdArfian Ismail[2]
[1]College of Education, Aliraqia University, Iraq
[2]Faculty of Computing, Universiti Malaysia Pahang, Malaysia

Abstract: *Optimization algorithms are widely used for the identification of intrusion. This is attributable to the increasing number of audit data features and the decreasing performance of human-based smart Intrusion Detection Systems (IDS) regarding classification accuracy and training time. In this paper, an improved method for intrusion detection for binary classification was presented and discussed in detail. The proposed method combined the New Teaching-Learning-Based Optimization Algorithm (NTLBO), Support Vector Machine (SVM), Extreme Learning Machine (ELM), and Logistic Regression (LR) (feature selection and weighting) NTLBO algorithm with supervised machine learning techniques for Feature Subset Selection (FSS). The process of selecting the least number of features without any effect on the result accuracy in FSS was considered a multi-objective optimization problem. The NTLBO was proposed in this paper as an FSS mechanism; its algorithm-specific, parameter-less concept (which requires no parameter tuning during an optimization) was explored. The experiments were performed on the prominent intrusion machine-learning datasets (KDDCUP'99 and CICIDS 2017), where significant enhancements were observed with the suggested NTLBO algorithm as compared to the classical Teaching-Learning-Based Optimization algorithm (TLBO), NTLBO presented better results than TLBO and many existing works. The results showed that NTLBO reached 100% accuracy for KDDCUP'99 dataset and 97% for CICIDS dataset.*

Keywords: *TLBO, feature subset selection, NTLBO, IDS, FSS.*

Received July 24, 2019; accepted May 9, 2020
https://doi.org/10.34028/iajit/18/2/5

1. Introduction

Recent advancements in, and popularization of, network and information technologies have increased the significance of network information security. [3] When compared to conventional network defense mechanisms, human-based smart Intrusion Detection Systems (IDSs) are able to take the initiative to either warn or intercept network intrusion. Nevertheless, most studies on information security have focused on ways to improve the effectiveness of smart network IDSs [4]. The use of smart IDSs is currently seen as an effective network security solution that can offer protection against attacks. Meanwhile, since the detection rate of existing IDSs is low when faced with new attacks and there is a high overhead when working with audit data, machine learning methods and optimization algorithms are often used for intrusion detection [26].

When the accuracy of detection is increased, the execution time will sometimes increase by a substantial amount. On the other hand, the execution time may significantly reduce at a cost of less accuracy. Therefore, the Feature Subset Selection (FSS) problem can be considered a multi-objective optimization problem, with more than one solution to the problem presenting themselves, from which the best one may be chosen [11]. For some, accuracy is very important. The solution that offers accuracy is therefore is chosen. Meanwhile, for others, the best solution is the one that reduces execution time, even if accuracy is also compromised to some extent.

As a novel metaheuristic, the Teaching-Learning-Based Optimization algorithm (TLBO), has been recently applied to various intractable optimization problems with considerable success. TLBO demonstrates its superiority to many other algorithms such as Genetic Algorithm, Particle Swarm, and Ant Colony. Moreover, TLBO requires fewer parameters for tuning during the execution process as compared to other algorithms. The combination of new multi-objective TLBO framework with supervised Machine Learning (ML) techniques is proposed in this paper for FSS in Binary Classification Problems (FSS-BCP) for intrusion detection. The process of selecting the least number of features without any effect on the result accuracy in FSS is considered a multi-objective optimization problem. The first objective is the number of features, while the second one concerns with the accuracy of the detection. The performance of TLBO has been reported as remarkable when compared to other metaheuristics algorithms. The New Teaching-Learning-Based Optimization (NTLBO) and a set of supervised ML techniques were deployed in this study for the selection of optimal feature subset. The contributions of this study are as follows: the first contribution is the utilization of the TLBO algorithm for feature selection in IDS for the first time; the second contribution is the new TLBO algorithm proposed in this study.

The remaining part of this paper is presented in the following manner: Section 2 presents a review of the works related to this study, while the FSS problem is introduced in section 3. In section 4, the proposed NTLBO is presented, while section 5 introduces the machine learning techniques applied with NTLBO. Section 6provides the results of the NTLBO algorithm in comparison to TLBO. Section 7 concludes the study.

2. Related Works

Intrusion detection is a trending security infrastructure topic in this era of big data. A combination of different ML methods together with optimization algorithms has been made and applied in IDS in order to differentiate normal network access to attacks. Some of the existing combinations include fuzzy logic, Cuttlefish Optimization Algorithm, K-Nearest Neighbor (KNN), Artificial Neural Network (ANN), Particle Swarm Algorithm (PCA), Support Vector Machine (SVM), and Artificial Immune System (AIM) approaches [2, 23, 34, 35]. Most of the approaches that combine ML with optimization algorithms have shown better performance as compared to conventional classification methods [31]. Numerous researchers have also proposed ML and optimization-based IDS [5]. Louvieris *et al.* [22] proposed a novel combination of techniques, namely K-means Clustering, Naïve Bayes, Kruskal-Wallis, and C4.5, allowing cyber-attacks as anomalies to be pinpointed with a high degree of accuracy within the cluttered and conflicted cybernetwork environment. Furthermore, the inclusion of the Naïve Bayes features election and the Kruskal-Wallis test in this approach facilitates the classification of both statistically significant and relevant feature sets, including a statistical benchmark for the validity of the approach. On the other hand, the detection of Structured Query Language (SQL) injection in this work remains low. Črepinšek *et al.* [8] presented a method for network intrusion detection based on Self-Organizing Maps (SOM) and Principal Component Analysis (PCA). Noise within the dataset and low-variance features is filtered by means of PCA and Fisher Discriminant Ratio (FDR). This procedure uses the most discriminative projections that are not solely based on the variance explained by the eigenvectors prototypes generated by the self-organizing process, which are modeled by d Gaussians, where d is the number of SOM units. This allows the proposed system to be trained only once, so the main limitation of this work is that the detection rate is still low [6]. Bamakan *et al.* [5] proposed a time-varying chaos-particle swarm optimization method to provide a new machine-learning intrusion-detection methodology based on two conventional classifiers: Multiple Criteria Linear Programming (MCLP) and SVM. The proposed method has been applied to set the parameters of these classifiers as well as provide the most appropriate subset of features simultaneously. The main drawback of this work is that the time needed for training is still considerable and needs to be decreased. Even though such combinations can improve the performance of IDS in terms of learning speed and detection rate as compared to conventional traditional algorithms, there is still a need for further improvement. With the increase in the number of audit data features, the performance of most IDSs has been affected in terms of classification accuracy and training time. This paper proposes the use of the TLBO method to address this issue through a fast and accurate optimization process that can improve the capability of IDS in finding the optimal detection model based on ML.A TLBO algorithm has been proposed by Rao and Patel [28], in which the optimization process for mechanical design problems does not need any user-defined parameter. This novel technique was tested on different benchmark functions where the results demonstrated the better performance of TLBO as compared to Particle Evolutionary Swarm Optimization, Artificial Bee Colony (ABC), and cultural Differential Evolution (DE). Das and Padhy [11] studied the possibility of applying a novel TLBO algorithm to the selection of optimal free parameters for an SVM regression model of financial time-series data using multi-commodity futures index data retrieved frommulti-cut crossover (MCX). From the experimental results, the proposed hybrid SVM-TLBO model appeared to have succeeded in finding the optimal parameters and yielded better predictions as compared to the conventional SVM.

Das *et al.* [10] proposed an extension of the hybrid SVM-TLBO model by introducing a dimension-reduction technique where the number of input variables can be reduced using PCA, Kernel Principal Component Analysis (KPCA), and Independent Component Analysis (ICA) (three common techniques for dimension reduction). The previous study also examined the feasibility of the proposed model using multi-commodity futures index data retrieved from Multi-Cut Crossover (MCX). Rao *et al.* [26] confirmed the superiority of the model as compared to some population-inspired optimization frameworks. Rao *et al.* [27] investigated the effect of sample size and number of generations on the algorithmic performance, and concluded that it is possible to apply this algorithm to several optimization cases with ease. Crepinšek *et al.* [8] solved the exact problems presented in [20, 29] using TLBO. Nayak *et al.* [25] developed a multi-objective TLBO in which a matrix of solutions was created for each objective. The teacher selection process in TLBO is mainly based on the best solution presented in the solution space, and learners are taught just to maximize that objective. All the available solutions in the solution space are sorted so as to generate a collection of optimal solutions. Shukla *et al.* [30] presented a multi-objective

TLBO based on different teaching techniques. A crossover operator was used between solutions in the teaching and learning phases, rather than a scalar function. Kiziloz *et al*. [18] suggested three multi-objective TLBO algorithms for FSS-BCP. Among the presented methods, a multi-objective TLBO with Scalar Transformation (MTLBO-ST) was found to be the fastest algorithm, although it provides a limited number of non-dominated solutions. Regarding the multi-objective TLBO with Non-dominated Selection (MTLBO-NS), it explores the solution space, produces a set of non-dominated solutions, and requires much execution time. Multi-objective TLBO with Minimum Distance (MTLBO-MD) generates similar solutions to that of MTLBO-NS;yet, in a significantly lesser amount of time. The proposed multi-objective TLBO algorithms have been evaluated in terms of performance using Logistic Regression (LR), SVM, and Extreme Learning Machine (ELM). According to Sultana and Jabbar [31] feature subset selection in the Wrapper method is made as a black box, i.e., there is no knowledge about the underlying algorithm. Feature subsets are selected based on inductive algorithms. This chosen feature subset estimates the accuracy of the training model. Depending on the accuracy measured from the previous step, the method will decide whether to add or remove a feature from the selected subset. Due to this, Wrapper methods are computationally more complex. Another method is Filter method. In this method, the model starts with all features and selects the best feature subset based on statistical measures, such as Pearson's correlation [32], Linear Discriminant Analysis (LDA), ANOVA, Chisquare [33], Wilcoxon Mann Whitney test [34], and Mutual Information (MI). All these statistical methods depend on the response and feature variables present in the dataset. Pearson's Correlation (PC) and Mutual Information methods are the commonly used statistical methods [9]. To date, the feature selection methods as discussed earlier use feature subset at the preprocessing level. The following algorithm that will be discussed is Embedded method. This type of method works in a way that the best features are selected during the learning process. The blending of feature selection during the learning process has advantages of improving computational cost, classification accuracy, and also avoiding training the model each time a new feature is added. The Embedded method selects the feature subset, and the interactions of the learning algorithm are different from other feature selection methods. Filter-based learning algorithms are not used for feature selection, whereas the Wrapper method uses the learning algorithm to test the quality of selected feature subsets. The Embedded method overcomes the computational complexity. In this method, appropriate feature selection and model learning are performed at the same time, and the features are selected during the training stage of the model. Due to this, the computational cost of this method is decidedly less as compared to the Wrapper method.

3. Feature Subset Selection Problem

FSS refers to the selection of feature subsets from a larger set of features. FSS prevents complex calculations by minimizing the number of features in a dataset, thereby improving the speed of classifiers. Several definitions of FSS exist in the literature [16]. Some definitions deal with the reduction of size of the selected subset, while others focus on the improvement of prediction accuracy. FSS is essentially a process of constructing an effective subset that represents the information contained in a dataset by eliminating redundant and irrelevant features. FSS mainly aims at finding the least number of features without having any significant influence on classification accuracy. Given that, optimal feature subset extraction is a complicated process and no exact polynomial time algorithm exists for solving it. FSS is, therefore, considered an NP-hard problem [12]. There are four steps in a typical FSS [16]: the first step involves a search for the selection of candidate features that will constitute the subsets. In the second step, these subsets are evaluated and compared with each other. The third step involves a determination of whether the termination condition has been met; otherwise, the first and second steps will be repeated. The final step checks if the optimal feature subset has been established based on prior knowledge.

Problem definition: This study involves two major parts: best feature subset selection, and performance evaluation. Since there are two major objectives, FSS can be considered as a multi-objective problem. A formal definition of finding optimal solutions through the satisfaction of both objectives is given in Equation (1).

$$\begin{aligned}&\text{Min } (f1)\\&\text{Max } (f2)\\&\text{Subject to}\\&f1 = |k|\\&f2 = \text{accuracy } (k) \text{ where } k \subseteq K\end{aligned} \tag{1}$$

Where *k* represents the subset of the original dataset (K) that optimizes *f1* and *f2* (the objectives). The second part involves the evaluation of the selected feature subsets based on accuracy (an established performance evaluation metric), as provided in Equation (2). Accuracy calculation requires the division of the instances that are classified correctly by all instances.

$$\text{Accuracy} = (TP+TN) / (TP+FP+FN+TN) \tag{2}$$

Where TP = true positive, TN=true negative, FP= false positive, and FN=false negative. In the proposed algorithms, the new TLBO (NTLBO) algorithm was executed at the FSS phase. The TLBO algorithm was initialized by randomly generating an initial population, namely the teacher and a set of students, which represent the set of solutions. In order to represent the features in the NTLBO algorithm, NTLBO is combined with Genetic Algorithm (GA), and the features represented are as a chromosome, which is one of the GA properties. For updating this chromosome, crossover and mutation were applied in the current study, as well as GA operators. Each solution in the population (classroom) is considered as an individual or a chromosome (refer to Figure 1 for the schematic representation of a chromosome). A feature gene of a chromosome with a value of 1 is considered selected, while a value of 0 denotes otherwise. The TLBO algorithm runs through iterations where the teacher is the best individual in the population and the rest of the individuals become the students. Having selected the teacher, NTLBO works in three phases: Teacher, Best Classmates (Learner Phase 1), and Learner Phase 2. In the Teacher Phase, the teacher shares knowledge with each student in a bid to enhance their level of knowledge. In the Best Classmate Phase, two best students are chosen to interact with the rest of the students. For the Learner Phase 2, the students interact randomly with each other in a bid to enhance their levels of knowledge. New chromosomes are generated in the proposed NTLBO using special crossover operators called half-uniform crossover and bit-flip mutation operators (refer to Figures 2 and 3). Two parent chromosomes, which may be a teacher, a student, or two students, are needed for the crossover operator. The crossover operator relies on the information of the two parent chromosomes; if both parents feature the same gene, the gene is kept; however, whenever there are different feature genes in both parents, a parent's gene is randomly chosen [13]. After this operation, one new chromosome is generated. The bit-flip mutation operates on a single chromosome in a bid to change a single gene based on a probabilistic ratio. If the value of the gene is zero, the value will be updated as one, or vice versa.

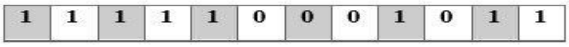

Figure 1. Schematic representation of a chromosome,1=selected features,0=unselected features.

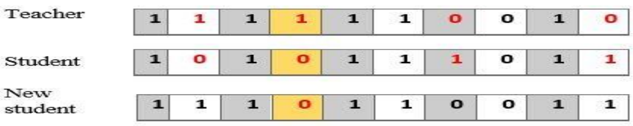

Figure 2. Crossover operator.

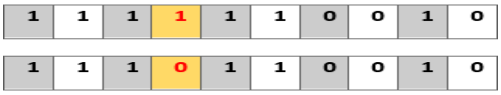

Figure 3. Mutation operator.

The detailed steps of NTLBO are as follows:

- *Step* 1: Initialize the population randomly with each population having a different set of features from 1 to a maximum number of features (41 in Knowledge Discovery and Data Mining Tools Competition dataset (KDD) and 78 in Canadian Institute for Cybersecurity Intrusion Detection system dataset (CICIDS). This step is captured in Line 1 of Algorithm 1.
- *Step* 2: Choose the best individual as a teacher. The chosen teacher interacts with all other individuals separately, and a crossover is applied with each one and then, a mutation is applied to all of the resulting individuals. The operators used are half-uniform crossover and bit-flip mutation operators (represented in Lines 2 to 8 in Algorithm 1).
- *Step* 3: Check the population (chromosome) that results from the crossover and mutation; if the new chromosome is better than the previous, then, the new one is kept; otherwise, the old one is retained. All the above mentioned steps are collectively called the Teacher Phase because all individuals learn from the best one (the teacher). This step is represented in Lines 9 to 10 in Algorithm 1.
- *Step* 4: After that, Learner Phase 1 or learning from best classmates is initiated. This phase begins with the fifth step, which is the selection of the best two individuals as students and the application of a crossover between them, followed by a mutation. If the new one is better than the previous two students, then, the newer choice is kept; otherwise, the older best choice is kept. This process is repeated with all other individuals (students). At this point, Learner Phase 1 is terminated (viewed in Lines 11 to 20 in Algorithm 1).
- *Step* 5: This step consists of Learner Phase 2, which involves choosing two random individuals (students) between whom a crossover is applied, followed by a mutation on the new individual. If the new individual is better than the previous two students, then, the new one is kept; otherwise, the best old one is retained. This step is repeated with all other students. At this point, the three main stages of ITLBO are completed and a check should be carried out on whether the termination criteria have been satisfied or not. If the termination criteria are satisfied, proceed to the next step; otherwise, the three main stages are repeated (Teacher Phase, Learner Phase 1, and Learner Phase 2). This step is represented in Lines 21 to 30 in Algorithm 1.
- *Step* 6: The final step is the application of non-dominated sorting to the result. Non-dominated sorting means no result (individual) is better than all other individuals. This step can be viewed in Line 31 in Algorithm 1.

4. New TLBO Algorithm (NTLBO)

The establishment of the best solution, or the decision on whether a new individual has improved, is a complicated task in a multi-objective optimization process. This is due to the chances of an enhancement in one objective causing a significant reduction in the other. The original multi-objective TLBO with minimum distance was first presented by Kiziloz *et al.* [18]. In the proposed NTLBO algorithm, non-dominated sorting and selection are used. The dominance of an individual over another individual is determined strictly on the basis of whether a minimum of one of its objectives is superior to that of the other, while keeping all the other objectives the same. A non-dominated scenario arises when an individual is not dominated by any other individual. All the non-dominated individuals make up the front line of the solution set. Those that are closest to the ideal point in the front line are chosen as the teacher. All the teachers teach all students discretely at the Teacher, Best Classmate, and Learner Phases. The details of the NTLBO algorithm are presented in Algorithm 1 and Figure 4.

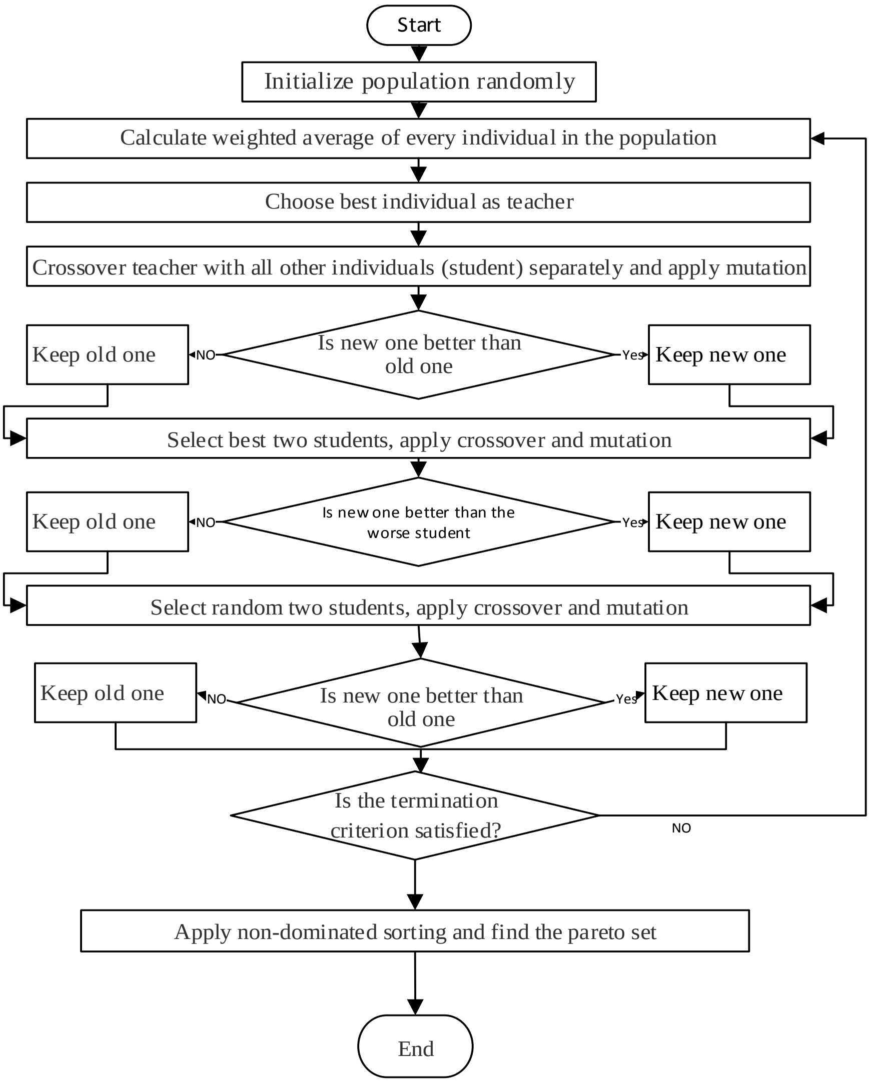
Start
Initialize population randomly
Calculate weighted average of every individual in the population
Choose best individual as teacher
Crossover teacher with all other individuals (student) separately and apply mutation
Keep old one
NO
Is new one better than old one
Yes
Keep new one
Select best two students, apply crossover and mutation
Keep old one
NO
Is new one better than the worse student
Yes
Keep new one
Select random two students, apply crossover and mutation
Keep old one
NO
Is new one better than old one
Yes
Keep new one
Is the termination criterion satisfied?
NO
Apply non-dominated sorting and find the pareto set
End

Figure 4. NTLBO Algorithm.

Algorithm 1: presents the details of the NTLBO algorithm.

```
Generate_population(population);
Calculate_weighted_average_of_individuals
(population);
for (k:=1 to number_of_generations) do
    Xteacher:=Best_individual (population);
    /* Learning from Teacher */teacher phase =e
    for (i:=1 to number_of_individuals) do
        Xnew: = Crossover (Xteacher, Xi);
        Xnew := Mutation(Xnew);
        if (Xnew is better than Xi) then
            Xi: = Xnew][
    /* Learning from Best Classmates */learner phase 1
    for (i: =1 to number_of_individuals) do
        m: = Select_best_individual_from (population);
        n: = Select_best_individual_from (population);
        n ≠m ≠ teacher*/
        Xnew: = Crossover(Xm, Xn);
        Xnew: = Mutation(Xnew);
        if (Xnew is better than Xm) then
            Xm: = Xnew;
        if (Xnew is better than Xn) then
            Xn: = Xnew;
    /* Learning from Classmates */learner phase 2
    for (i: =1 to number_of_individuals) do
        m: =Select_random_individual_from (population);
        n: =Select_random_individual_from (population);
        n ≠m ≠ teacher*/
        Xnew: = Crossover (Xm, Xn);
        Xnew:= Mutation (Xnew);
```

```
if (Xnew is better than Xm) then
    Xm: = Xnew;
if (Xnew is better than Xn) then
    Xn: = Xnew;
Show_the_pareto_optimal_set (population);
```

- *Step* 1: the first step in this algorithm is to initialize the population randomly. A total of 20 populations are generated; each population having a different set of features from 1 to a maximum number of features (41 in KDD and 78 in CICIDS).
- *Step* 2: the second step is to calculate the weighted average of every individual population, weighted average is the accuracy of each set of features
- *Step* 3: the third step is to choose the best individual as the teacher. The chosen teacher interacts with all other individuals separately, and a crossover is applied with each one before a mutation is employed to all resulting individuals. The crossovers used are half-uniform crossover and bit-flip mutation operator.
- *Step* 4: the fourth step involves checking the population (chromosome) resulting from the crossover and mutation; if the new chromosome is better than the previous, then the new one is kept. Otherwise, the old one is retained. The best means have the best accuracy, All the above mentioned steps are included in the Teacher Phase, because all individuals learn from the best one (the teacher). After that, Learner Phase 1, or learning from best classmates, commences. This phase begins with the fifth step,
- *Step* 5: this step begin by selecting the best two individuals as students and applying a crossover between them, followed by a mutation. If the new one is better than the previous two students, then the newer choice is kept; otherwise, the older choice is kept. This process is then repeated with all other individuals (students).
- At this point, Learner Phase 1-learning from the best students-is finished and the next stage is Learner Phase 2.
- *Step* 6: this step starts with choosing two random individuals (students) between whom a crossover is then applied, followed by a mutation on the new individual. If that new individual is better than the old two students (in terms of accuracy), then the new one is kept. Failing that, the best old one is kept. This step is repeated with all other students. At this point, the main three stages of NTLBO are completed, and a check should be carried out on whether the termination criteria have been satisfied or not. If the former, the next step is proceeded; in the case of the latter, the three main stages (Teacher Phase, Learner Phase 1, and Learner Phase 2) are repeated. The final step is the application of non-dominated sorting to the result. Non-dominated sorting means no result (individual) is better than all other individuals. Each phase requires a comparison of Xnew with Xold using one of the machine learning algorithms applied in this study (SVM, ELM, and LR). The comparison involves classifying the compared subsets of features (Xnew and Xold) to obtain high accuracy, which signifies is the best one.

5. Applied Machine Learning

The present study evaluated the solutions achieved using ITLBO by deploying three ML techniques (LR, SVM, and ELM). LR is a common, fast, and easily implemented classifier; SVM is well-known for its effectiveness in binary classification; whereas ELM is a newly introduced but promising classifier. LR: Classification with LR is performed by estimating an event's occurrence probability based on the similarity of given data points. It finds the probability of the event occurrence by employing a sigmoid function. If the occurrence probability of an event is > 0.5, then the LR predicts the event as "occurred" or "not occurred", as the case may be. SVM: Classification tasks using SVM are performed through the construction of a separating line between the given data points [15]. The data points closest to this line are designated as Support Vectors (SVs). This line is iteratively constructed through the maximization of the margin between the SV and the line of the classes. This idea originates from the assumption that an increase in the margin can reduce the generalization error. ELM: ELM is built as a Feed forward Neural Network (FFNN) with a hidden layer, an input layer, and an output layer. The training data are fed into the model through the input layer, where they are then weighted and forwarded to the hidden layer via a function. A similar transformation is executed between the hidden and output layers. The FFNN requires iterative tuning of its parameter; however, no parameter tuning occurs in the ELM. Therefore, the learning time of ELM is lower as compared to those of conventional FFNNs.

6. Experimental Setup

The experimental scenario, problem instances, and the outcome of the experiments are all presented in this section. In this study, the experiments were performed on two intrusion datasets (KDDCUP'99 and CICIDS), which were reduced, because of the focus on binary classification to accommodate only two classes (normal and intrusion). In order to make the validation fairer, K-fold validation was used, where the value of K is set to 10 [17].

KDDCUP'99: This dataset was first used to build a network intrusion detector at the 3rd International Knowledge Discovery and Data Mining Tools Competition [7]. The DARPA intrusion detection evaluation program was set up in 1998 by the MIT Lincoln Laboratory as a simulated environment for gathering raw TCP/IP dump data for a Local Area

Network (LAN) [23]. It was set up with the aim of comparing various intrusion detection methods based on their performance. A version of the DARPA'98 dataset was used in the KDDCUP'99 dataset [12]. The DARPA'98 dataset consists of compressed raw TCP dump data of seven weeks of network traffic. It is approximately4 gigabytes in size and can be processed into about 5,000,000 connection records, each of about 100 bytes [14]. In the dataset, the two weeks' test data contains approximately 2,000,000 connection records. The KDD training dataset is comprised of about 4,900,000 single connection vectors of 41 features each, which are labeled either as normal or an attack of a specific type [1]. The attack types in the dataset were categorized into four major categories:

1. Probing attack: This is an effort by an attacker to gain network information simply to circumvent the network's security controls.

The CICIDS2017 dataset consists of benign and most current common attacks, which mimic real-world data (PCAPs). It also contains the results of a network traffic analysis, obtained by using a CICFlowMeter. The flows are labeled based on the timestamp, source and destination ports, source and destination IPs, protocols, and attack.

2. Denial-of-Service (DoS) attack: In this type of attack, the intruder intentionally denies legitimate network access by making the system too busy to process legitimate requests.
3. User-to-Root (U2R) attack: The attacker gains access to the network by accessing the system as a legitimate user, before exploiting the lapses in some systems to gain root access.
4. Remote-to-User (R2L) attack: This is a form of attack where an invader exploits vulnerabilities in machines by sending packets to them over a network in a bid to gain local access as a legal user.

Although several types of R2U attacks exist, the most common types are those executed via social engineering. These attacks (DoS, U2R, R2L, and probing) are classified into 22 different attack types in the KDDCUP'99 dataset, as shown in Table 1. These do not only refer to the specific case of KDDCUP'99 dataset; additionally, several known classifications and taxonomies of computer system attacks were also analyzed in this study [15].

Table 1. KDD dataset.

Attack classes	22 types of attacks
DoS	smurt, neptune, pod, teardrop, back, land,
R2L	phf, ftp-write, imap, multihop, warezclient, warezmaster, spy, guess password
U2R	perl, loadmodule, buffer-overflow, rootkit
Probing	portsweep, ipsweep, satan, nmap

The CICIDS2017 dataset [7, 21] satisfies the 11 indispensable features of a valid IDS dataset, namely anonymity, available protocols, feature set, attack diversity, complete capture, complete interaction, complete network configuration, complete traffic, metadata, heterogeneity, and labeling [1, 7, 16]. There are 3,057,503 rows in the CICIDS2017, devised on eight files with each row containing 79 features. In the CICIDS2017, each row is labeled as benign or as one of the 14 attack types. A summary of the distribution of different attack types and the benign rows is presented in Table 2.

Table 2. CICIDS dataset.

Attack class	14 types of attacks
DOS	DDoS, slowloris, Heratbleed, Hulk, GoldenEye, Slowhttptest
PortScan	Portscan
Bot	Bot
Brute-Force	FTP-Patator, SSH-Patator
Web Attack	Web attack XSS, web attack SQL injection, web attack brute force
Infiltration	Infiltration

In this study, the experiments were performed on a computer running an Intel Core i7-4810 processor with a CPU clock rate of 2.80 GHz and an 8GB main memory. The classification aspect of the algorithms was done using Matlab 2017a. The two important parameters that must be decided prior to running NTLBO were population size and number of generations. A higher value of these parameters ensures a higher result of accuracy, even though the computation time will be increased. An investigation of a new individual is time-inefficient.

7. Experiment Results and Discussion

The parameters used in this study are shown in Table 3.

Table 3. Parameters used in this study.

Parameter	**Value**
Population size for NTLBO	40
Number of generations for NTLBO	60
Crossover type	Half-uniform
Mutation type	Bit-flip

The tables below presents the accuracy results for both datasets. The accuracy result of the KDDCUP'99 dataset is presented in Table 4.

Table 4. Accuracy result of kddcup'99 dataset.

Classifier	**TLBO**		**NTLBO**	
	No. of features	Accuracy	No. of features	Accuracy
LR	3	0.995	1	0.99
			2	0.995
Total time	12.2512		11.4023	
SVM	3	0.995	2	0.97
	6	1.00	3	0.995
			4	1.00
Total time	2382.3301		1305.0355	
ELM	3	0.97	1	0.985
	4	0.99	2	0.99
	5	0.995	3	1.00
	8	1.00		
Total time	4.0717		4.4261	

From Table 4, both TLBO and NTLBO offered the same execution time for each ML technique. For each ML, the number of features, accuracy, and execution time were calculated. The numbers in red suggest the best results for both TLBO and NTLBO. NTLBO consistently presented better accuracies as compared to TLBO using the three ML techniques. It also presented better time accuracy using LR and SVM ML techniques. However, TLBO provided a better execution time with ELM as compared to NTLBO. The results of the CICIDS2017 dataset are presented in Table 5.

Table 5. Accuracy Result of CICIDS2017 Dataset.

Classifier	TLBO		NTLBO	
	No. of features	Accuracy	No. of features	Accuracy
LR	14	0.94	7	0.945
	15	0.965	9	0.95
	27	0.975	13	0.955
			15	0.965
			22	0.97
Total time	33.06		29.089	
SVM	24	0.84	14	0.905
	26	0.92	18	0.92
			21	0.93
Total time	4161.3924		5484.097	
ELM	13	0.86	6	0.88
	15	0.885	7	0.92
	16	0.905		
	19	0.91		
	20	0.92		
Total time	3.4071		6.5312	

With the CICIDS2017 dataset, NTLBO consistently showed better accuracy than TLBO using the three ML techniques. With the LR technique, NTLBO presented a better execution time as compared to TLBO. In contrast, with the SVM and ELM techniques, TLBO was better than NTLBO. The reason that the result of NTLBO was always better than TLBO in terms of accuracy is Learner Phase 1 (learning from best classmates). In TLBO, there was no learning from best classmates other than choosing random students and learning from them; whereas in NTLBO, learning from best classmates means that in the final phase (Learner Phase 2), learning from the best students provides the optimal solution.

The Detection Rate (DR) is the percentage of the samples correctly classified by the classifier to their correct class.

$$\text{Detection rate} = \frac{\text{TP}}{TP+FP} \tag{3}$$

Another statistical test is the error rate, which is the proportion of patterns that have been incorrectly classified by the model. ER is calculated based on Equation (4).

$$\text{Error Rate} = \frac{\text{FP+FN}}{TP+FP+FN+TN} \tag{4}$$

Table 6 illustrates the results of Equations (3) and (4).

Table 6. Results of Equations (3) and (4).

	KDDCUP'99	CICIDS2017
Detection Rate	0.9995	0.9903
Error Rate	0.0045	0.027

Another statistical test (T-test) was applied to demonstrate the superiority of NTLBO over TLBO. The P-values and T-values are shown in Table 7, whereby the small values showed that NTLBO was highly significant.

Table 7. T-Test.

	KDDCUP'99	CICIDS2017
P-Value	0.0156	0.0068
T-Value	3.174	4.044

Comparison results with existing works showed that the proposed model performed better than many existing works in terms of accuracy as shown in Table 8.

Table 8. Comparison with existing works.

Ref.	Dataset	Accuracy
[24]	CICIDS	97.90 %
[2]	CICIDS	97.08 %
[19]	KDD	99.75 %
[26]	KDD	99.89 %
Proposed method	CICIDS	97.5 %
Proposed method	KDD	100 %

8. Conclusions

This paper proposes a new multi-objective teaching learning-based algorithm (NTLBO) for feature subset selection problems in intrusion detection. The performance of the new algorithm was demonstrated to be superior to that of TLBO in FSS problems on two large intrusion datasets. The proposed NTLBO consistently presented better accuracy in the execution time than TLBO in several instances. On the statistical tests (confusion matrix) applied to the NTLBO detection rate and error rate extracted from the confusion matrix, NTLBO showed a higher detection rate for both the KDDCUP'99 and ICIDS2017 datasets. It showed a low error rate for the two datasets. As a recommendation, the proposed NTLBO should be applied to multi-class classification problems, and more ML techniques could be used for evaluating its performance.

Acknowledgment

This study was supported by Fundamental Research Grant Scheme (FRGS) with Vot No. FRGS/1/2018/ICT02/UMP/02/2: RDU190113 from Ministry of Higher Education (MOHE) and managed by Universiti Malaysia Pahang.

References

[1] Aljarah I. and Ludwig S., "Mapreduce Intrusion Detection System Based on A Particle Swarm Optimization Clustering Algorithm," *in Proceedings of IEEE Congress on Evolutionary Computation Conference*, Cancun, pp. 955-962, 2013.

[2] Aljawarneh S., Aldwairi M., and Yassein M.,

"Anomaly-Based Intrusion Detection System Through Feature Selection Analysis and Building Hybrid Efficient Model," *Journal of Computational Science*, vol. 25, pp. 152-160, 2018.

[3] Altay B., Dokeroglu T., and Cosar A., "Context-Sensitive and Keyword Density-Based Supervised Machine Learning Techniques for Malicious Webpage Detection," *Soft Computing*, vol. 23, no. 4, pp. 4177-4191, 2018.

[4] Alsajri M., Ismail M., and Abdul-Baqi S., "A Review on the Recent Application of Jaya Optimization Algorithm," *in Proceedings of 1st Annual International Conference on Information and Sciences*, Fallujah, pp. 129-132, 2018.

[5] Bamakan S., Wang H., Yingjie T., and Shi Y., "An Effective Intrusion Detection Framework Based on MCLP/SVM Optimized by Time-Varying Chaos Particle Swarm Optimization," *Neurocomputing*, vol. 199, pp. 90-102, 2016.

[6] Cai J., Luo J.,Wang S., and Yang S., "Feature Selection in Machine Learning: A New Perspective," *Neurocomputing*, vol. 300, pp. 70-79, 2018.

[7] Chaudhary A., Tiwari V., and Kumar A., "A Novel Intrusion Detection System for Ad Hoc Flooding Attack Using Fuzzy Logic in Mobile Ad Hoc Networks," *in Proceedings of International Conference on Recent Advances and Innovations in Engineering*, Jaipur, pp. 1-4, 2014.

[8] Črepinšek M., Liu S., and Mernik L., "A Note on Teaching-Learning-Based Optimization Algorithm," *Information Sciences*, vol. 212, pp. 79-93, 2012.

[9] Dash M. and Liu H., "Feature Selection for Classification," *Intelligent Data Analysis*, vol. 1, no. 3, pp. 131-156, 1997.

[10] Das S., Achary N., and Padhy S., "Novel Hybrid SVM-TLBO Forecasting Model Incorporating Dimensionality Reduction Techniques," *Applied Intelligence*, vol. 45, no. 4, pp. 1148-1165, 2016.

[11] Das S. and Padhy S., "A Novel Hybrid Model Using Teaching-Learning-Based Optimization And A Support Vector Machine for Commodity Futures Index Forecasting," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 1, pp. 97-111, 2018.

[12] De la Hoz E., De la Hoz E., Ortiz A., Ortega J., and Prieto B., "PCA filtering and Probabilistic SOM for Network Intrusion Detection," *Neurocomputing*, vol. 164, pp. 71-81, 2015.

[13] Ding D., Han Q., Xiang Y., Ge X., and Zhang X., "A Survey on Security Control and Attack Detection for Industrial Cyber-Physical Systems," *Neurocomputing*, vol. 275, pp. 1674-1683, 2018.

[14] Dokeroglu T., "Hybrid Teaching-Learning-Based Optimization Algorithms for The Quadratic Assignment Problem," *Computers and Industrial Engineering*, vol. 85, pp. 86-101, 2015.

[15] Dumais S., Platt J., Heckerman D., and Sahami M., "Inductive Learning Algorithms and Representations for Text Categorization," *in Proceedings of the 7th International Conference on Information and Knowledge Management*, Bethesda, pp. 148-155, 1998.

[16] Eesa A., Orman Z., and Brifcani A., "A Novel Feature-Selection Approach Based on the Cuttlefish Optimization Algorithm for Intrusion Detection Systems," *Expert Systems with Applications*, vol. 42, no. 5, pp. 2670-2679, 2015.

[17] Guo C., Ping Y., Liu N., and Luo S., "A Two-Level Hybrid Approach for Intrusion Detection," *Neurocomputing*, vol. 214, pp. 391-400, 2016.

[18] Kiziloz H., Deniz A., Dokeroglu T., and Cosar A., "Novel Multiobjective TLBO Algorithms for The Feature Subset Selection Problem," *Neurocomputing*, vol. 306, pp. 94-107, 2018.

[19] Khaleel M., Ismail M., Yunan U., and Kasim S., "Review on Intrusion Detection System Based on the Goal of the Detection System," *International Journal of Integrated Engineering Special Iss*, vol. 10, no. 6, pp. 197-202, 2018.

[20] Lin W., Ke S., and Tsai C., "CANN: An Intrusion Detection System Based on Combining Cluster Centers and Nearest Neighbors," *Knowledge-Based Systems*, vol. 78, pp. 13-21, 2015.

[21] Li Y., Wang J., Tian Z., and Young C., "Building Lightweight Intrusion Detection System Using Wrapper-Based Feature Selection Mechanisms," *Computers and Security*, vol. 28, no. 6, pp. 466-475, 2009.

[22] Louvieris P., Clewley N., and Liu X., "Effects-Based Feature Identification for Network Intrusion Detection," *Neurocomputing*, vol. 121, pp. 265-273, 2013.

[23] Mahdavifar S. and Ghorbani A., "Application of Deep Learning to Cybersecurity: A Survey," *Neurocomputing*, vol. 347, 2019.

[24] Mohammed M., Hasan R., Ahmed M., Tapus N., and Shanan M., "A Focal Load Balancer Based Algorithm For Task Assignment In Cloud Environment," *in Proceedings of The 10th International Conference on Electronics, Computers and Artificial Intelligence*, Iasi, pp. 1-4, 2018.

[25] Nayak M., Nayak C., and Rout P., "Application Of Multi-Objective Teaching Learning Based Optimization Algorithm to Optimal Power Flow Problem," *Procedia Technology*, vol. 6, pp. 255-264, 2012.

[26] Rao R., Savsani V., and Vakharia D., "Teaching-Learning-Based Optimization: A Novel Method for Constrained Mechanical Design Optimization

Problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303-315, 2011.

[27] Rao R., Savsani V., and Balic J., "Teaching-Learning-Based Optimization Algorithm for Unconstrained and Constrained Real-Parameter Optimization Problems," *Engineering Optimization*, vol. 44, no. 12, pp. 1447-1462, 2012.

[28] Rao R. and Patel V., "An Improved Teaching-Learning-Based Optimization Algorithm for Solving Unconstrained Optimization Problems," *Scientia Iranica*, vol. 20, no. 3, pp. 710-720, 2013.

[29] Sen R., Chattopadhyay M., and Sen N., "An Efficient Approach to Develop an Intrusion Detection System Based on Multi Layer Backpropagation Neural Network Algorithm: IDS Using BPNN Algorithm," *in Proceedings of the ACM SIGMIS Conference on Computers and People Research*, California, pp. 105-108, 2015.

[30] Shukla A., Kumar S., and Singh H., "ANN Based Execution Time Prediction Model and Assessment of Input Parameters through ISM," *The International Arab Journal of Information Technology*, vol. 17, no. 5, pp. 683-691, 2020.

[31] Sultana A. and Jabbar M., "Intelligent Network Intrusion Detection System Using Data Mining Techniques," *in Proceedings of 2nd International Conference on Applied and Theoretical Computing and Communication Technology*, Bangalore, pp. 329-333, 2016.

[32] Tao P., Sun Z., and Sun Z., "An Improved Intrusion Detection Algorithm Based on GA and SVM," *IEEE Access*, vol. 6, pp. 13624-13631, 2018.

[33] Tian Y., Mirzabagheri M., Bamakan S., Wang H., and Qu Q., "Ramp Loss One-Class Support Vector Machine; A Robust and Effective Approach to Anomaly Detection Problems," *Neurocomputing*, vol. 310, pp. 223-235, 2018.

[34] Yang Y. and Pedersen J., "A Comparative Study on Feature Selection in Text Categorization," *in Proceedings of the 4th International Conference on Machine Learning*, San Francisco, pp. 412-420, 1997.

[35] Yu L. and Liu H., "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," *in Proceedings of the 20th International Conference On Machine Learning*, Washington DC, pp. 856-863, 2003.

Mohammad Aljanabi is currently Lecturer at Alsalam University College and Al Iraqia University. He is working for a PhD in computer science at Universiti Malaysia Pahang; he received his BSc from Almustansryah University (Iraq) and his MSc from BAMU University (India).

MohdArfian Ismail is Senior Lecturer at the Faculty of Computer Systems and Software Engineering in Universiti Malaysia Pahang. He received his BSc, MSc, and PhD degrees in computer science from Universiti Teknologi Malaysia in 2008, 2011, and 2016, respectively. His current research interests are in the areas of ML and optimization methods.