A multi-class classification MCLP model with particle swarm optimization for network intrusion detection

A M VISWA BHARATHY^{1,*} and A MAHABUB BASHA²

¹Anna University, Chennai 600025, India

²Department of Computer Science and Engineering, K.S.R. College of Engineering, Tiruchengode 637215, India

e-mail: viswabharathy86@yahoo.co.in; mahabubbasha1952@yahoo.com

MS received 25 March 2016; revised 22 October 2016; accepted 28 October 2016

Abstract. The critical data we share through computer network gets stolen by unethical means. This unethical way of accessing one's data without proper authentication becomes intrusion. To solve this issue, in this paper we propose a new network intrusion detection method, Multi-Class Classification Multiple Criteria Linear Programming (MCC-MCLP) model. MCLP is a mathematical classification technique that is used widely to solve real-time data mining problems. So far, the literature discusses only about binary classification MCLP. But in this paper we propose a Multi-Class Classification MCLP model. We use PSO for fine-tuning the parameters of MCC-MCLP. KDD CUP 99 data set is used for performance evaluation of the proposed method. Our MCC-MCLP method classifies the data better and helps in fine-tuning the parameters with the help of PSO. The results clearly show that the proposed model performs better in terms of detection rate, false alarm rate and accuracy.

Keywords. Multi-class classification; multiple criteria linear programming; network intrusion detection; particle swarm optimization.

1. Introduction

As stated in the abstract, individuals and organizations rely upon computer networks and internet for their business needs and transactions. It has become a part and parcel of every network [1, 2] to protect it from potential attacks and unethical activities [1, 3]. Every network has its own set of rules and data sets to protect it from intruders and malicious intrusions. Either system-linked or third-party-integrated software system is being implemented in every system or network to protect its privacy and secrecy, for example, antivirus, firewall, internet security services and other data encryption and system or network protection services [3]. However, they have not served their purpose to the full owing to intangible attacks [4].

In particular an intrusion detection system (IDS) is classified into six types [5]. Active and passive, host and network, signature and anomaly are the six variations of IDS. Active deals with prevention before any malicious activity occurs. Passive comes into rescue only when an abnormal event is registered with the system or network. Host protects an individual system in which it is active and network protects the whole network where it is deployed.

It monitors all the incoming and outgoing packets of the network.

The most important and critical types are signature- and anomaly-based IDS [2, 3]. Signature is called as knowledge based, which has an already reported intrusion in its database. Hence, every time a new entry comes it checks its database and verifies it [2]. Thus, even a modification of already existing and reported intrusion cannot be detected by signature IDS. Anomaly-based IDS is an artificial intelligence coupled detection system. It tries to learn the behaviour patterns of the network, infers something and prepares a report of itself. If any abnormal behaviour unrelated to the known existing patterns occurs it is deemed as intrusion [1, 2]. But, always this is not right. Signature has a low false alarm rate and anomaly has a high false alarm rate, as any deviation in normal behaviour is termed as malicious activity [1].

An IDS consists of data gathering, pre-processing, intrusion detection and reportion (report and action integrated) [2], although it is not confined to these systematic steps alone. IDS is still an unexplored area, as no definite system has been developed to date. It is too difficult to build a comprehensive IDS, because many factors such as amount of traffic, a fully unbalanced database and dynamic environment affect its performance [2, 6].

*For correspondence

2. Related work

In 2009 Gang Kou *et al* proposed the multi-class classification model for multiple criteria mathematical programming for network IDS (see [7]). They used two network intrusion data sets: KDD99 and NeWT. The data sets were trained using the following nine classification algorithms: Bayesian network, Naive Bayes, support vector machine (SVM), linear logistic, C4.5, See5, classification and regression tree (CART), MCMP and MCMP with kernel. The results of their experiments were convincing, which showed high classification accuracies, and low false alarm rates were achieved by the MCMP with kernel method.

Multiple criteria linear programming (MCLP) is a classification-based mathematical programming model [8], which is used in detecting credit card and bankruptcy frauds [9, 10]. But the performance degrades when the parameters are not fine-tuned. Although many methods such as grid algorithm and genetic algorithm have been proposed, PSO with SVM proposed by Xibin *et al* [11] shows better performance in terms of short search time, less computation time and global optimal performance.

The support vector machine (SVM) is a widely known classification method proposed by Mohammed and Sulaiman [12] in wireless local area networks. The method proved to be efficient and accurate in detecting intrusions. However, Chen *et al* [13] proposed a combinatorial method of SVM with artificial neural networks (ANNs) to further improve the performance. Gang *et al* [14] came up with another solution, integrating fuzzy clustering (FC) with ANN, to improve the stability of the system and to maintain the frequency of attacks at a very low rate. Chung and Wahid [4] proposed a feature selection method and a simplified swarm optimization for classification by integrating the intelligent dynamic-swarm-based rough set (IDS-RS). However, the idea by Wu and Banzhaf [2] of integrating the application of computational intelligence (CI) with IDS got attention from all researchers. CI methods are classified as artificial immune systems, artificial neural networks, swarm intelligence (SI) and soft computing. Not only did they impress everyone but also showed a greater performance result in terms of high computation speed, low fault tolerance rate and handling of heavy noisy data sets. As proposed in [15] by Kolias *et al*, integration of SI, i.e., particle swarm optimization (PSO) with the other machine learning classification algorithms (MLCA) has witnessed a better performance by decreasing the training time.

In 2013, Wang and Wang [16] proposed a multiple-criteria and multiple-constraint levels linear programming for error correction classification model. This model was supported by a pair of hyper-planes and so it was called dual hyper-plane MC2LP. They came up with two models for correcting two types of errors: Type I and Type II. Both the models were experimented and they corrected errors at a

convincing rate. However, some points that were correctly classified in the original MCLP model were misclassified. One important reason for this was that the new dual-hyperplane MC2LP was not focussed on minimizing the number of misclassification points. Instead, the model optimized the sum of deviations.

In 2014, Shijun and Jiang [17] proposed a Chaos-PSO-trained SVM for automatic sense-through-foliage target recognition system. Their prime focus on the paper was to ease the process of target recognition and detection in military and in other research areas. SVM is a prolific tool to solve the recognition problem with a little sampling, non-linearity and high dimension. Shijun and Jiang used adaptive chaos particle swarm optimization (ACPSO) to determine the best parameters for SVM with the highest possible accuracy and generic ability. Principal component analysis (PCA) was performed to extract the features of targets and a hybrid feature selection technique was used to eliminate the redundant, irrelevant and unreliable features. The computational results of the method showed better performance against the existing techniques.

In 2014, Shahram and Shaker [18] proposed a technique for defending against SYN flooding attacks using the PSO approach. The authors considered the defence issue as an optimization problem and used PSO to solve the problem. They performed packet-level simulation analysis in ns-2 environment. The proposed PSO-SYN defence technique [18] has shown good performance in decreasing the number of blocked TCP connection requests and blocking down the share of attack connections from the buffer space.

In 2015, Ahmad [19] proposed his experiment and results in the paper feature selection using particle swarm optimization in intrusion detection. He used principal component analysis (PCA). It is a statistical method used for data analysis and is a very useful method for feature selection. He flipped an idea that false alarm rates are increasingly on the go as raw data sets are used. Hence he proposed feature selection method (PCA) for transforming raw features into primary features. False alarm rates were reduced drastically and detection rates increased linearly.

In 2015 Ismaila *et al* [20] proposed a combined negative selection algorithm-PSO for e-mail spam detection system. Threshold and distance value were used for selecting unique features. Local outlier factor (LOF) was used as a fitness function to calculate the reachable distance between non-spam space and the particle's LOF within the neighbourhood to get best features during detector generation. The experimental results and analysis showed that NSA-PSO performed better than the NSO model.

In 2015, Hamed and Kannan [21] proposed a hybrid PSO and GA for closed-loop supply chain network design in large-scale networks. They attempted to design and plan a closed-loop supply chain for large-scale networks and achieved success too. They combined the two popular meta-heuristic algorithms GA and PSO and developed a new hybrid algorithm. They used the aspects of PSO to

improve the drawbacks in GA. The new algorithm was validated using CPLEX and MATLAB. Hamed and Kannan [21] experimented their method and documented it as a case study involving an Iranian hospital furniture manufacturer. The results were convincing enough to prove the effectiveness of the hybrid algorithm when compared with the GA and PSO.

Seyed *et al* [22] proposed a model based on MCLP that used PSO for tuning the parameters of MCLP. Although many methods like SVM, grid algorithm, genetic algorithm and C5.0 were used, PSO gave convincing results in terms of accuracy, running time and false alarm rate.

In 2016 Arijit De *et al* proposed a composite particle swarm algorithm for sustainable integrated dynamic ship routing and scheduling optimization [23]. They came up with the idea of mixed integer non-linear programming (MINLP). They dealt with the issues of multiple time horizons, sustainability, and varying demand–supply at various ports. The authors proposed a composite PSO algorithm, which obtained better results than PSO and GA. The results proved that carbon emission, fuel cost and fuel consumption constraints can be comfortably added to the mathematical model for encapsulating the sustainability dimensions.

In 2016 Seyed *et al* came up with a framework based on MCLP–SVM optimized by time-varying chaos PSO for Intrusion Detection System [24]. They considered time-varying inertia weight and acceleration co-efficient for parameter setting and feature selection in MCLP and SVM. A modified form of MCLP, penalized MCLP, is used to support unbalanced data sets. A weighted objective function is used to maximize the detection rate and minimize the false alarm rate. They tested the method on NSL-KDD data set and obtained better results by using all features.

In 2016 Nourelhuda *et al* [25] proposed a two-layer PSO for properly aligning the fragmented protein sequence (FTLPSO) [25]. The problem of Multiple Sequence Alignment was addressed here. A two-layer PSO algorithm was applied to align each fragment, which has the ability to align longer sequences than that by traditional PSO. They cured unconstrained optimization problems and increased the diversity of particles [26]. The method was tested on some Balibase benchmarks of different lengths. The numerical results were compared to CLUSTAL Omega, CLUSTAL W2, TCOFFEE, KALIGN and DIALIGNPFAM. It showed better results in alignment using the proposed FTLPSO.

In 2016, Mansour *et al* [27] proposed a new technique, combinatorial PSO, for solving the Blocking Flowshop Scheduling Problem. They addressed the problem of flowshop scheduling with blocking constraints. Their objective was to minimize the makespan criterion using hybrid combinatorial particle swarm optimization (HCPSP). They proposed the idea of iterated local search algorithm based on probabilistic perturbation. It was sequentially and consistently applied to the PSO for improving the scope of solution. The computational results showed that the approach

improved 76 among 120 best known literature solutions, that too in short time requirements.

References [16, 28] give a detailed review on multiple criteria and multiple constraint-level linear programming method (MC²-LP). The authors show MC² Simplex method and an Interior Point MC² algorithm as solution methods for existing problems of MC²-LP. Although the methods are not discussed and used here, this can be extended in future as Multi-Class Classification [1] along with SOM and PSO [7, 25–32] for more optimum performance.

The remainder of this paper is organized as follows. In section 3 we describe MCC-MCLP; section 4 explains the tuning technique PSO, section 5 describes the experiment performed, section 6 compares all the previous methods MCLP-PSO, MCLP, SVM and C5.0 along with the proposed MCC-MCLP-PSO method given in this paper and gives an overall report supporting the MCC-MCLP-PSO model. Finally we conclude in section 7 and provide future work in section 8.

3. Multi-Class Classification Multiple Criteria Linear Programming

Seyed *et al* [22] discussed Binary Classification of MCLP with PSO. MCLP was given by Shi in 1998. It is a multiple classification problem. It constructs a linear programming model based on the two main objectives and certain constraints to solve a classification problem. As MCLP was not enough to tune the parameters in finding solution, Seyed *et al* proposed MCLP along with PSO for tuning of parameters. They discussed only two classes, such that each sample of testing set belonged to either of the two classes, Good or Bad. But this paper deals with multi-class classification MCLP model with PSO, i.e., each item in the data set is classified into one of these five classifications, namely, Genuine, Probe, DoS, U2R and R2L. The results of their work along with other techniques and the results of the proposed method of this paper will be discussed in section 6 as comparative analysis.

Let X be an n -dimensional attribute vector as given in Eq. (1.1), such that each sample of X belongs to a particular class label attribute.

$$X = (x_1, x_2, \dots, x_n) \quad (1.1)$$

x denotes the samples distributed in the given class X .

The distribution of samples in the space is given in figure 1a. This classifier attempts to find the best hyper-plane that accurately separates X into different classes [33]. An appropriate hyper-plane is given by the relation

$$w \cdot x = b \quad (1.2)$$

where w is defined as the weight for a subset of variables and b , the threshold, is found to separate the multiple classes [22].

Figure 1. Graph depicting the distribution of (a) samples $X = (x_1, x_2, \dots, x_n)$ and (b) weights for each class of sample in the space X .

$$W_i = (w_1, w_2, \dots, w_n)^T, \quad \text{where } i = 1-5 \text{ and} \quad (1.3)$$

$$w = a \vee c \vee d \vee e \vee f \quad (1.4)$$

represents weight for each class.

n represents the number of samples in each subset of class. Figure 1b shows the distribution of weights for each class of sample wrt Eqs. (1.3) and (1.4).

$$W_1 = (a_1, a_2, \dots, a_n)^T \quad (1.5)$$

$$W_2 = (c_1, c_2, \dots, c_n)^T \quad (1.6)$$

$$W_3 = (d_1, d_2, \dots, d_n)^T \quad (1.7)$$

$$W_4 = (e_1, e_2, \dots, e_n)^T \quad (1.8)$$

$$W_5 = (f_1, f_2, \dots, f_n)^T. \quad (1.9)$$

Equations (1.5)–(1.9) denote the weights for separate classes Genuine, Probe, DoS, U2R and R2L, correspondingly.

The class label is given by the equation

$$C = \{G, P, D, U, R\} \quad (1.10)$$

where G = Genuine, P = Probe, D = DoS, U = U2R and R = R2L attacks.

The separable hyper-planes are defined by the equations

$$(w \cdot x_i) \leq b, \quad \forall x_i \in G_i \quad (1.11)$$

$$(w \cdot x_i) \geq b, \quad \forall x_i \in H_i \quad (1.12)$$

where

if $w = c$ then $H = P$

if $w = d$ then $H = D$

if $w = e$ then $H = U$

if $w = f$ then $H = R$.

Figure 2 depicts the bifurcation of the classes based on the hyper-planes denoted by Eqs. (1.11) and (1.12). Equation (1.11) denotes the hyper-plane that separates genuine packets and Eq. (1.12) denotes the separation of four types of attack packets based on their weights.

The two main objectives that we need to formulate are MSD and MMD. MSD is defined as Minimizing the Sum of Distance, which is to decrease the sum of the samples' overlapping degree from the separating hyper-plane to its minimum limit. MMD is defined as Maximizing the Minimum Distance, which is to increase the sum of the distances of each sample from the hyper-plane to its maximum limit.

Here ξ_i is defined as the overlapping degree from the hyper-plane with respect to the training sample x_i , which should be minimized; β_i is defined as distance between the hyper-plane and the training sample x_i , which should be maximized. The MSD and MMD with respect to the hyper-planes are given in Eqs. (1.13) and (1.14), respectively:

$$(w \cdot x_i) \leq b + \xi_i - \beta_i, \quad \forall x_i, y = \text{Genuine} \quad (1.13)$$

$$(w \cdot x_i) \geq b - \xi_i + \beta_i, \quad \forall x_i, y = P \vee D \vee U \vee R. \quad (1.14)$$

A relation between these two variables can be written as

$$Y(Xw - eb) \geq \beta_i - \xi_i \quad (1.15)$$

where e is defined as the vector of ones. Hence the MCLP model can be formulated as follows [34]:

$$\min \sum_{i=1}^n \xi_i \quad (1.16)$$

$$\min \sum_{i=1}^n \beta_i \quad (1.17)$$

$$\text{s.t. } (x_i, w) = b + y_i(\xi_i - \beta_i), \quad i = 1, \dots, n \quad (1.18)$$

$$\xi, \beta \geq 0. \quad (1.19)$$

As described in [35] by Shi *et al* the best available trade-off between $-\sum_{i=1}^n \xi_i$ and $\sum_{i=1}^n \beta_i$ is considered for an optimal solution. To be able to understand this, we assume

Figure 2. Two-dimensional plane showing the separation of classes.

the ideal value of $-\sum_i \xi_i$ to be $\xi^* > 0$ and the ideal value of $\sum_i \beta_i$ to be $\beta^* > 0$. Then if $-\sum_i \xi_i > \xi^*$, the regret measure is written as follows:

$$-d_{\xi^+} = \sum_{i=1}^n \xi_i + \xi^*. \quad (1.20)$$

Otherwise it is defined as 0. If $-\sum_{i=1}^n \xi_i < \xi^*$, the regret measure is written as follows:

$$d_{\xi^-} = \xi^* + \sum_i \xi_i. \quad (1.21)$$

Otherwise it is defined as 0. Thus the relationships of these measures are as follows: [22]:

i.

$$\xi^* + \sum_{i=1}^n \xi_i = d_{\xi^-} - d_{\xi^+} \quad (1.22)$$

ii.

$$\left| \xi^* + \sum_{i=1}^n \xi_i \right| = d_{\xi^-} + d_{\xi^+} \quad (1.23)$$

iii.

$$d_{\xi^-}, d_{\xi^+} \geq 0. \quad (1.24)$$

Similarly, according to [22] we derive

$$\beta^* - \sum_i \beta_i = d_{\beta^-} - d_{\beta^+} \quad (1.25)$$

$$|\beta^* - \sum_i \beta_i| = d_{\beta^-} + d_{\beta^+} \quad (1.26)$$

and $d_{\beta^-}, d_{\beta^+} \geq 0$.

An MCLP model for multi-class separation is designed as follows:

minimize

Figure 3. Geometric meaning of MCC-MCLP model.

$$d_{\xi^-} + d_{\xi^+} + d_{\beta^-} + d_{\beta^+} \quad (1.27)$$

$$s.t. \quad \xi^* + \sum_{i=1}^n \xi_i = d_{\xi^-} - d_{\xi^+} \quad (1.28)$$

$$\beta^* - \sum_{i=1}^n \beta_i = d_{\beta^-} - d_{\beta^+} \quad (1.29)$$

$$(x_i, w) = b + y_i(\xi_i - \beta_i), \quad i = 1, \dots, n. \quad (1.30)$$

Here ξ^* and β^* are the two parameters we have optimized by the PSO method; w and b are unrestricted. The geometric meaning of Eq. (1.27) is shown in figure 3.

4. Particle swarm optimization (PSO)

PSO is a method of distributed solutions for complicated problems by interacting with simple agents and their associated environment [2, 36, 37]. PSO is an inspiration or upgradation of the swarm intelligence (SI) technique [2, 37, 38]. This SI technique was inspired from the group behaviour of animals such as bird flock, fish shoal and ant colony. PSO was given in 1995 by Eberhart and Kennedy. But a simpler and better variant of PSO was proposed by Rui *et al* [39] in 2007. PSO has many merits such as simplicity in implementation, robustness, scalability, flexibility and quick solution finding when compared with any other algorithms of its class [2, 36, 38].

In PSO each sample of the set is called as a particle. The algorithm starts working by creating the initial particles and assigning initial velocities to them. At each particle location, the algorithm evaluates the objective function and assesses the best functional value and best location. Here we need to understand three terminologies: p_{best} , l_{best} and g_{best} .

Each particle in the problem domain space tracks its coordinates associated with the best solution ever the particle has achieved. This value is called p_{best} ; l_{best} is the best location and functional value attained so far by any other particle in the neighbourhood region. Out of all the particle population the best value obtained by a particle is termed as g_{best} .

Figure 4. Position and velocity update for a single particle in a uniformly distributed two-dimensional search space. (a) Time t , (b) time $t + 1$.

Let us consider $f: \mathbb{R}^n \rightarrow \mathbb{R}$ as a cost function, which must be minimal. The function takes argument in the form of a real-number vector and produces output in the form of real numbers. This indicates the objective function value of the given solution. The degree of inclination of f is not known. The goal is to find a solution a for which $f(a) \leq f(b)$ for all b in the search space, which would generally mean that a is the global minimum value. Maximization can be performed by considering the function $h = -f$ instead.

Let P denote the number of particles in the swarm space, where $x_i \in \mathbb{R}^n$ denotes the position of each particle in the search space;

$v_i \in \mathbb{R}^n$ denotes its velocity in the space.

Assume $i = 1, \dots, P$,

DO

SET particle position to $x_i \sim U(b_{lb}, b_{ub})$,

ASSIGN x_i to p_i

IF $(p_i) < f(p_b)$ THEN

SET p_i to p_b

ASSIGN particle's velocity $v_i \sim U(-|b_{ub}-b_{lb}|, |b_{ub}-b_{lb}|)$

REPEAT UNTILL terminating condition is satisfied

For each particle in the space

DO

For each dimension in the space

DO

randomly select numbers as $r_p, r_g \sim U(0,1)$

UPDATE particle's velocity: $v_{i,d} \leftarrow \omega v_{i,d} + \phi_p r_p (p_{i,d} - x_{i,d}) + \phi_g r_g (g_d - x_{i,d})$

Update $x_i \leftarrow x_i + v_i$

IF $f(x_i) < f(p_i)$ THEN

SET x_i to p_i

IF $f(p_i) < f(p_b)$ THEN

SET p_i to p_b

Now g retains the best known solution.

p_i specifies the best known position of particle i and p_b points the best known position of the entire swarm.

b_{lb} and b_{ub} denote, respectively, lower and upper boundaries of the search space. At each iteration the particle updates its velocity and position as follows:

$$v^i[t+1] = w \cdot v^i[t] + c_1 r_1 (p^{i,best}[t] - p^i[t]) + c_2 r_2 (p^{g,best}[t] - p^i[t]) \quad (1.31)$$

$$p^i[t+1] = p^i[t] + v^i[t+1], \quad (1.32)$$

where $i = 1, 2, 3, \dots, N$, N denotes swarm population space.

$v^i[t]$ denotes velocity vector in t th iteration, $p^i[t]$ denotes the current position of the i th particle, $p^{i,best}[t]$ denotes previous best position of i th particle, $p^{g,best}[t]$ denotes previous best position of all particles and w helps in controlling the pressure of local and global search; c_1 and c_2 are positive acceleration coefficients called, respectively, cognitive parameter and social parameter; r_1 and r_2 are random numbers between 0 and 1.

Now we present the geometrical representation of the PSO algorithm. Figure 4 shows how the velocity components drive the particle towards the global best position (g_{best}) at times t and $t + 1$.

Likewise all particles move randomly in the search space and attain their best position. The particle that has the l_{best} , i.e., the local best, attracts all other particles in its subset. At each time the particles form their own subset that has an l_{best} . Figure 5 illustrates l_{best} and g_{best} formation.

Figure 5. Multi-particle position and velocity update in a uniformly distributed two-dimensional search space to attain l_{best} and g_{best} at time t .

5. Experiments and results

Our proposed model is tested with 10% data randomly selected from the global standard data set KDD CUP 99. The KDD99 data set was derived during the year 1999 from DARPA 98 network data set with the sole purpose of evaluating IDSs [2, 4]. It is the global standard benchmark used to measure the performance of the IDS [1, 40]. Ref. [22] describes only genuine attacks. However, we have discriminated the attacks using MCC-MCLP, optimized the parameters and features using PSO and tested individually. The TCP connection consists of 41 features in the training set with a label that determines whether it is normal or an attack [40]. The attack comes under any of the following categories such as Probe, DoS, U2R and R2L, which are already described in the MCLP model. Altogether 5263 normal record and 6572 attacks have been chosen by the random selection method. The proposed method's experimental procedure and its evaluation methodology are shown as follows.

Step 1: Prepare the training data set $X = (x_1, x_2, \dots, x_n)$, where X is the n -dimensional attribute vector; $y_i \in \{0, 1\}$, where 0 denotes normal and 1 denotes attack.

Step 2: Initialize the parameters $w, c_1, c_2, r_1, r_2, \zeta^*, n$ and β^* . We define the values of the parameters as $c_1 = 1, c_2 = 1.5, w_{max} = 0.85, w_{min} = 0.30, n = 20$ and $-0.00001, -0.1$ are assigned, respectively, as lower and upper bounds of ζ^* [8]; 10 and 1000 are assigned as lower and upper bounds of β^* [35].

Step 3: Accuracy is defined as $\frac{(TP+TN)}{(TP+FP+FN+TN)}$. It is the ratio of correct predicted records to entire records. It serves as the fitness function. $FAR = \frac{FP}{FP+TN}$. It is the false alarm rate, which specifies normal records notified as attacks.

Step 4: In each iteration, update the position and velocity of each particle based on p_{best} and g_{best} .

Step 5: End PSO by obtaining best accuracy and MCLP parameters ζ^* and β^* .

Step 6: Apply the best optimized MCLP model on the training data set.

Table 1. Confusion matrix.

Actual class	Predicted class	
	P	N
P	True positive (TP)	False negative (FN)
N	False positive (FP)	True negative (TN)

Now that we have understood the experiment of MCC-MCLP-PSO, its results are tabulated and compared to the other four models in section 6. The comparative analysis in section 6 gives a detailed inference of all the methods and ranks them according to the features.

6. Comparative analysis

The proposed model considers four metrics into account: accuracy, false alarm rate, execution time and detection rate, to compare the results of MCC-MCLP-PSO model to those of MCLP-PSO, MCLP, SVM and C5.0. Table 1 shows the confusion matrix and table 2 compares these models.

Table 2 clearly brings out a comparison of all five models with parameters detection rate, accuracy, false alarm rate and execution time. It is very clear from the table that PSO fine-tunes the parameter and performs better when applied to train and classify the features of MCLP method. The detection rate has increased to 99.34 from 99.13%, which shows an increase of 0.21%. The accuracy has risen from 98.62 to 99.14%, thereby improving the performance by 0.52%, which is highly progressive in nature. False alarm rate has gone down from 1.947 to 1.765%, which is 0.182% lesser. This increase in accuracy and detection rate and reduction in false alarm rate can be attributed to the multi-class classification method in MCLP. As the features were well classified and trained using PSO it performs better when compared with the MCLP-PSO

Table 2. Comparison of MCC MCLP based on PSO model with the MCLP-PSO, MCLP, support vector machine (SVM) and C5.0.

Classifier	Training data set				Testing data set			
	DR	Accuracy	FAR	ET (ms)	DR	Accuracy	FAR	ET (ms)
MCC MCLP-PSO	0.9988	0.9872	0.01823	3.21	0.9934	0.9914	0.01765	3.39
MCLP-PSO	0.9987	0.9721	0.01724	3.56	0.9913	0.9862	0.01947	3.37
MCLP	0.9823	0.9805	0.03718	3.27	0.9746	0.9738	0.03633	3.44
SVM	0.9964	0.9937	0.02256	3.63	0.9897	0.9819	0.02751	3.58
C5.0	0.9777	0.9776	0.06329	3.31	0.9838	0.9827	0.04268	3.29

FAR false alarm rate, ET execution time in milli-seconds, DR detection rate.

Table 3. Ranking of classifiers based on DR.

Classifier	DR	Rank
MCC MCLP-PSO	0.9934	1
MCLP-PSO	0.9913	2
SVM	0.9897	3
C5.0	0.9838	4
MCLP	0.9746	5

Table 4. Ranking of classifiers based on accuracy.

Classifier	Accuracy	Rank
MCC MCLP-PSO	0.9914	1
MCLP-PSO	0.9862	2
C5.0	0.9827	3
SVM	0.9819	4
MCLP	0.9738	5

Table 5. Ranking of classifiers based on FAR.

Classifier	FAR	Rank
MCC MCLP-PSO	0.01765	1
MCLP-PSO	0.01947	2
SVM	0.02751	3
MCLP	0.03633	4
C5.0	0.04268	5

method. Though PSO optimizes the parameters well, when the features are properly classified and trained it performs even better. This is clearly inferred from this paper. We present a series of tables ranking each method based on the parameters detection rate, accuracy, false alarm rate and execution time.

Tables 3, 4, 5 and 6 clearly present the scenario of how well PSO performs when applied on well-defined and classified parameters and features. MCC-MCLP-PSO gets clear edge over others in all three categories—DR, accuracy and FAR. At the same time we have to agree that it ranks third in terms of running time. It is 0.02 ms slower than MCLP-PSO and 0.10 ms slower than C5.0. However, this is a negligible fact when compared with its convincing

Table 6. Ranking of classifiers based on ET.

Classifier	ET	Rank
C5.0	3.29	1
MCLP-PSO	3.37	2
MCC MCLP-PSO	3.39	3
MCLP	3.44	4
SVM	3.58	5

Figure 6. Comparison chart of testing data set of multi-class classification MCLP based on PSO model with the MCLP-PSO, MCLP, support vector machine (SVM) and C5.0.

performance of detection capability, accurate efficiency and false alarm rate.

The comparison chart in figure 6 clearly shows the performance of the five models that have been considered for study and analysis. The MCC-MCLP-PSO model performs better than the other four models in terms of accuracy, false alarm rate and detection rate. The higher execution time can be attributed to fine-tuning of multi-classes, namely probe, DoS, U2R and R2L attacks.

Table 7 compares and analyses the ability of MCC MCLP-PSO model with respect to different types of attacks and genuine data sets. Every classification type is given additional 15% attack or normal data to analyse its performance. The results are convincing. We were able to understand the following things from this experiment.

When 85% genuine data, i.e., almost only genuine data, are given to the model, it shows high accuracy and detection

Table 7. Comparison of multi-class classification MCLP based on PSO model with respect to different types of attacks and genuine data.

Classification	Training data set				Testing data set			
	Accuracy	FAR	ET (ms)	DR	Accuracy	FAR	ET (ms)	DR
<i>MCC MCLP-PSO</i>								
Genuine + 15% attack	0.9954	0.00817	1.654	0.9976	0.9972	0.00886	1.571	0.9963
Probe + 15% normal	0.9959	0.01112	2.01	0.9964	0.9961	0.01131	2.13	0.9949
DoS + 15% normal	0.9931	0.01217	2.09	0.9967	0.9889	0.01206	2.19	0.9951
U2R + 15% normal	0.9872	0.02368	2.17	0.9887	0.9892	0.02347	2.08	0.9893
R2L + 15% normal	0.9891	0.02197	2.14	0.9912	0.9923	0.02171	2.21	0.9901

FAR false alarm rate, ET execution time in milli-seconds, DR detection rate.

Figure 7. Comparison chart of testing data set of multi-class classification MCLP based on PSO model with respect to different types of attacks.

rate ranging from 99.63 to 99.72%. It also recorded low false alarm rate of 0.886% and low execution time of 1.571 ms. With probe also it shows high accuracy and detection rate of almost above 99.50%. For other sets of data with Dos, U2R and R2L the accuracy dropped down below 99.25% with Dos and U2R ranking worse below 99% and detection rate swivels between 99 and 99.50%. But FAR has taken a whopping increase of above 2% with U2R and R2L. The ET of all sets of data except genuine has crossed 2 ms with worse running time of 2.21 ms by R2L. We have presented a graphical analysis of the results in table 7 in figure 7 to understand the MCC-MCLP-PSO model better with respect to different types of attack and genuine data sets.

7. Conclusion

Globally many scientists and researchers are struggling every day to find out an optimum solution for Intrusion Detection System that supports high accuracy, high detection rate, low execution time and low false alarm rate. In this paper we have proposed a Multi-Class-Classification-based MCLP model, which uses PSO to fine-tune the parameters. The performances of the proposed

methodology were examined by the data set obtained from the KDD CUP99 DARPA 98 network. It is very clear from the analysis that MCC-MCLP-PSO method performs better than other methods. The proposed model performs better in terms of accuracy, detection rate and false alarm rate but with a slight compromise with running time. However, the latter is a negligible fact. The parameters are chosen by either user validation or cross-validation.

8. Future work

As a future work multiple criteria and multiple constraint-level linear programming method (MC^2 -LP) can be extended as multi-class classification and combined with self-organizing particle swarm optimization (SOPSO) to mitigate the intrusions [41, 42]. The results of the work can be discussed and compared with the metrics such as false alarm rate, detection rate, accuracy and execution time.

References

- [1] Gang K, Peng Y, Chen Z and Shi Y 2009 Multiple criteria mathematical programming for multi-class classification and application in network intrusion detection. *Inf. Sci.* 179(4): 371–381
- [2] Wu S X and Banzhaf W 2010 The use of computational intelligence in intrusion detection systems: a review. *Appl. Soft Comput.* 10(1): 1–35
- [3] Tsai C F, Hsu Y F, Lin C Y and Lin W Y 2009 Intrusion detection by machine learning: a review. *Expert Syst. Appl.* 36(10): 11994–12000
- [4] Chung Y Y and Wahid N 2012 A hybrid network intrusion detection system using simplified swarm optimization (SSO). *Appl. Soft Comput.* 12(9): 3014–3022
- [5] <http://www.omniseu.com/security/infrastructure-and-email-security/types-of-intrusion-detection-systems.php>
- [6] Ahmed P, Mona T, Kaveh B and Joaquim C Jr 2013 An intrusion detection and prevention system in cloud computing: a systematic review. *J. Netw. Comput. Appl.* 36(1): 25–41

- [7] Anurag S and Christian W O 2009 Performance comparison of particle swarm optimization with traditional clustering algorithms used in self-organizing map. *Int. J. Comput. Intell.* 5(1): 32–41
- [8] Shi Y 2010 Multiple criteria optimization-based data mining methods and applications: a systematic survey. *Knowl. Inf. Syst.* 24(3): 369–391
- [9] Li A, Yu Y and Liu Z 2013 Sensitivity analysis of b in MCLP model for credit cardholders classification. *Proc. Comput. Sci.* 17: 1023–1031
- [10] Zhiwang Z, Gao G and Shi Y 2014 Credit risk evaluation using multi-criteria optimization classifier with kernel, fuzzification and penalty factors. *Eur. J. Oper. Res.* 237(1): 335–348
- [11] Xibin W, Wen J, Zhang Y and Wang Y 2014 Real estate price forecasting based on SVM optimized by PSO. *Optik—Int. J. Light Electron Opt.* 125(3): 1439–1443
- [12] Mohammed M N and Sulaiman N 2012 Intrusion detection system based on SVM for WLAN. *Proc. Technol.* 1: 313–317
- [13] Chen W H, Hsu H S and Shen H P 2005 Application of SVM and ANN for intrusion detection. *Comput. Oper. Res.* 32(10): 2617–2634
- [14] Gang W, Hao J, Ma J and Huang L 2010 A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. *Expert Syst. Appl.* 37(9): 6225–6232
- [15] Kolias C, Kambourakis G and Maragoudakis M 2011 Swarm intelligence in intrusion detection: a survey. *Comput. Secur.* 30(8): 625–642
- [16] Wang B and Wang Y 2013 A multiple-criteria and multiple-constraint levels linear programming based error correction classification model. *Proc. Comput. Sci.* 17: 1073–1082
- [17] Shijun Z and Jiang T 2014 A novel particle swarm optimization trained support vector machine for automatic sense-through-foliage target recognition system. *Knowl.-Based Syst.* 65: 50–59
- [18] Shahram J and Shaker V 2014 Defense against SYN flooding attacks: a particle swarm optimization approach. *Comput. Electr. Eng.* 40(6): 2013–2025
- [19] Ahmad I 2015 Feature selection using particle swarm optimization in intrusion detection. *Int. J. Distrib. Sens. Netw.* 11(10): 806954
- [20] Ismaila I, Ali S, Ngoc Thanh N, Sigeru O, Ondrej K, Kamil K and Marek P 2015 A combined negative selection algorithm—particle swarm optimization for an email spam detection system. *Eng. Appl. Artif. Intell.* 39: 33–44
- [21] Hamed S and Kannan G 2015 A hybrid particle swarm optimization and genetic algorithm for closed-loop supply chain network design in large-scale networks. *Appl. Math. Model.* 39(14): 3990–4012
- [22] Seyed M H B, Behnam A, Mahboubbeh M and Shi Y 2015 A new intrusion detection approach using PSO based multiple criteria linear programming. *Proc. Comput. Sci.* 55: 231–237
- [23] De A, Vamsee K R M, Angappa G, Nachiappan S and Manoj K T 2016 Composite particle algorithm for sustainable integrated dynamic ship routing and scheduling optimization. *Comput. Ind. Eng.* 96: 201–215
- [24] Seyed M H B, Huadong W, Tian Y and Shi Y 2016 An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. *Neurocomputing* 199: 90–102
- [25] Nourelhuda M, Moustafa E, Tarek Hosny T and Mofreh S 2016 Fragmented protein sequence alignment using two-layer particle swarm optimization (FTLPSO). *J. King Saud Univ. – Sci.* <http://dx.doi.org/10.1016/j.jksus.2016.04.007> (in press)
- [26] Bonyadi M 2014 Particle swarm optimization: theoretical analysis, modifications and applications constrained optimization problems. PhD Thesis, School of Computer Science, Faculty of Engineering, Computer and Mathematical Sciences, The University of Adelaide. <https://digital.library.adelaide.edu.au/dspace/bitstream/2440/92599/3/02whole.pdf>
- [27] Mansour E, Bassem J and Siarry P 2016 Combinatorial particle swarm optimization for solving blocking flowshop scheduling problem. *J. Comput. Des. Eng.* <http://dx.doi.org/10.1016/j.jcde.2016.05.001>
- [28] Dandan C, Zhong Y, Liao Y and Li L 2013 Review of multiple criteria and multiple constraint-level linear programming. *Proc. Comput. Sci.* 17: 158–165
- [29] Liberios V, Baláz A and Chovanec M 2006 Intrusion detection system using self organizing map. *Acta Electrotech. Inf.* 6(1): 1–6
- [30] Lotfi Shahreza M, Moazzami D, Moshiri B and Delavar M R 2011 Anomaly detection using a self-organizing map and particle swarm optimization. *Sci. Iran. D* 18(6): 1460–1468
- [31] Xiang X, Ernst R D, Russell E, Zina B M and Robert J O 2003 Gene clustering using self-organizing maps and particle swarm optimization. In: *Proceedings of the International Parallel and Distributed Processing Symposium, IPDPS'03*, p. 154
- [32] Swagatam D, Ajith A and Amit K 2008 Automatic kernel clustering with a multi-elitist particle swarm optimization algorithm. *Pattern Recogn. Lett.* 29(5): 688–699
- [33] Fang W, Zhang P, Shang Y and Shi Y 2013 The application of MCLP in advertisement clicking events prediction. *Proc. Comput. Sci.* 18: 1720–1729
- [34] Li A, Shi Y and He J 2008 MCLP-based methods for improving “Bad” catching rate in credit cardholder behavior analysis. *Appl. Soft Comput.* 8(3): 1259–1265
- [35] Shi Y, Tian Y, Kou G, Peng Y and Li J 2011 *Optimization based data mining: theory and applications*. Springer, London
- [36] Mandal J K, Mukhopadhyay S and Pal T 2016 *Handbook of research on natural computing for optimization problems*. Information science reference, IGI Global, Pennsylvania
- [37] Olariu S and Zomaya A Y 2005 *Handbook of bioinspired algorithms and applications*. CRC Press, London
- [38] Riccardo P, Kennedy J and Blackwell T 2007 Particle swarm optimization: an overview. *Swarm Intell.* 1(1): 33–57
- [39] Mendes R, Kennedy J and Neves J 2004 The fully informed particle swarm: simpler, maybe better. *IEEE Trans. Evol. Comput.* 8(3): 204–210
- [40] Abubakar A I, Haruna C, Sanah A M and Libabatu B I 2015 A review of the advances in cyber security benchmark datasets for evaluating data-driven based intrusion detection systems. *Proc. Comput. Sci.* 62: 221–227
- [41] Kohonen T 2001 *Self-organizing maps*. 3rd ed. Springer, Berlin
- [42] O'Neill M and Brabazon A 2008 Self-organising Swarm (SoSwarm). *Soft Comput.* 12(11): 1073–1080