ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2020.0287

IMPROVED INTRUSION DETECTION CLASSIFIER USING CUCKOO SEARCH OPTIMIZATION WITH SUPPORT VECTOR MACHINE

D. Viknesh Kumar[1] and Velmani Ramasamy[2]

[1]Department of Computer Science and Engineering, Sri Guru Institute of Technology, India

[2]Department of Electronics and Communication Engineering, Siddhartha Institute of Technology, India

Abstract

This paper proposes Cuckoo Search Optimization (CSO) with Support Vector Machine (SVM) for the intrusion detection system (IDS). This work covers modules including preprocessing, feature selection and classification. The pre-processing is carried out using min-maximum standardization to remove missing values and filter the redundancy characteristics from the specified NSL KDD cup data set. Preprocessing helps primarily to increase the accuracy of the description. Instead CSO is used to pick the most suitable and optimum functions. With CSO, the search efficiency is improved and then the analysis is carried out more effectively to classify the intrusions using the SVM algorithm. This classification algorithm is used to increase the accuracy of attack detection. The test results show that the CSO with SVM algorithm is more efficient than existing methods.

Keywords:

Intrusion Detection, Feature Selection, SVM, CSO

1. INTRODUCTION

Intrusion Detection has always been an important part of machine learning research. The identification of improper use and the detection of phenomenon were two common methods. User activity is comparable with the documented trend of disruptive attacks in the harassment detection, which essentially is an adaptive pattern process. Look for trends that vary from the normal characteristics when identifying abnormalities. The detection system has the fundamental challenge in identifying typical characteristics, given its ability to detect unknown attacks [1].

Host-based IDS (HIDS) program is characterized as the mechanism by which incidents are tracked, captured and evaluated on a host to recognize intrusions that breach the IT safety policy, and contribute to a lack of privilege, honesty or accessibility. Therefore, HIDS must be in a circumstance to identify and mark them as an anomalous operation, which typically prohibits intrusion from proceeding on the host [2].

An algorithm of reciprocal knowledge that chooses the best classification method analytically. The feature selection algorithm based on mutual information may handle linear and nonlinear, dependent data characteristics. In the case of network intrusion detection, its efficacy is measured. An Intrusion Detection System is developed using the features chosen by our proposed feature selection algorithm [3], known by the Least Square Support Vector Machine-based IDS (LSSVM-IDS).

Three intrusion detection data sets-KDD Cup 99, NSL-KDD and Kyoto 2006+ evaluate the efficiency of LSVM-IDS. The test results indicate that LSSVM-IDS has more important characteristics for increasing accuracy and the computing costs with the selection function algorithm.

For known signatures of discovered vulnerabilities, IDS can reliably identify intrusion attacks. Nevertheless, safety experts introduced an intrusion that restricted applicability for misuse detection in order to establish intelligent IDS to identify rules or signatures. The anomaly detection method, on the other side, tackles statistical analysis and pattern recognition issues. It is stated that it will allow detection of new attacks for the classification model, which can extract the pattern and knowledge of intrusion while training, without prior knowledge [4]. However, this method suffers a high FPR on normal network traffic classification.

The most powerful approach for minimizing the manual creation of interference and foreseeing novel attacks while information is obtained from data is gained through studying the classification principles of network data [5]. The problem of detection accuracy and unbalanced detection rates, however, is false positive and the input attributes are redundant. The identification of intruding into real-time high-speed network is another issue with existing IDS networks due to the fact that huge amounts of data have to be managed in a very short time by the IDS program.

Feature-selection approaches manage large amounts of data with unrelated and redundant properties, resulting in sluggish training and testing, extreme computing resources and little identification precision. The collection of roles is thus an important problem in the identification of intrusion. The approach explores the use of developmental algorithms for IDS selection functions. It compared the performance of three feature selection algorithms: Genetic Algorithms (GAs), Particle Swarm Optimization (PSO) and Differential Evolution (DE) using KDD Cup 1999 dataset [6].

2. RELATED WORK

Hu et al. [11] provided high-detection, low false-alarm and fast machine-learning intrusion-based intrusion detection algorithms. Decision stumps are used as ineffective classifiers during the process. For definite and incessant features, the rules for decision are provided. The associations of these two different types are definitely managed without involuntary change between the continual and categorical characteristics by integrating the weak classifiers for continuous features and weak classifiers for categorical features into a powerful classifier. The algorithm is assisted by adaptable initial weights and a conservative solution.

Ambusaidi et al. [9] concentrate on the collection of features and use a method that chooses features dependent on Mutual Information (MI) for the classification of features. The best number of nominee apps is selected from the top of the rankings in wrapper fashion, searching for the best subset with the greatest precision. The method combines two main phases: filter ranking and the improved forward floating selections (IFFS) based on wrappers using LS-SVM and accuracy in classification. The filter approach seeks to reduce wrapper quest computation costs by removing unnecessary and obsolete functions from the initial feature set. In order to search for an acceptable subset that increases the classification precision, the IFFS framework wrapper is used. The objective is to achieve high wrapper accuracy as well as the effectiveness of filter approaches. Lastly, the final subset is then forwarded to the LSSVM Classifier to construct IDS to test the utility of our proposed function selection method. Experimental results for validation obtained with various KDD Cup 99 data sets are presented.

In conjunction with several selection methods including principal component analysis, sequential bloating and correlation-based selection, Aziz et al. [7] employed an approach for generating anomalous detectors, using genetic algorithms. Genetic algorithm was used to generate a series of detectors with a deterministic crowding niche technology. The results show that sequential floating techniques with the genetic algorithm have the best results, relative to the results of others, particularly sequential floating selection with train precision 92.86% and 85.38%.

Moradi et al. [8] has an approach to intrusion detection in the neural network. For intrusion detection based on an off-line research method a Multi-Layer Perceptron (MLP) is used. Although the majority of previous studies have focused on classifying documents into one of the two types–regular and assault–the goal is to solve a problem of a multi-class nature in which the attack form is also identified by the neural network. In order to find the optimal neural network with regard to the number of hidden layers, different neural network structures are analyzed. In the training stage, a validation method for an early stop is also used to increase the neural network's generalization capacity. The results show that the built system is able to identify records with an exactness of 91% in the neural network with two cached neuron layers and 87% with one cached layer.

3. PROPOSED METHODOLOGY

For a stronger NSL KDD cup dataset classification outcome, the CSO with SVM algorithm is implemented in the proposed system. The overall architecture is given in Fig.1.

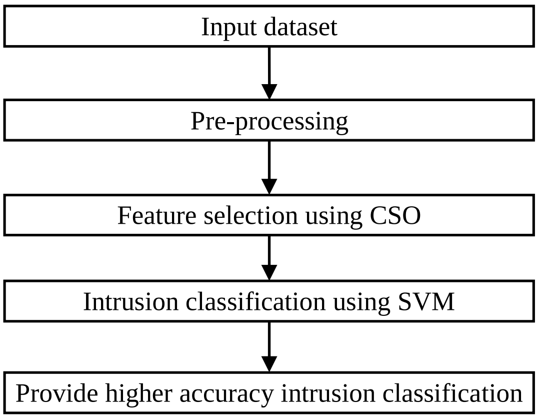
Input dataset
Pre-processing
Feature selection using CSO
Intrusion classification using SVM
Provide higher accuracy intrusion classification

Fig.1. Overall block diagram of proposed intrusion detection system

3.1 DATASET COLLECTION

The studies using NSL KDD data set because the analysis of different methods for intrusion detection is considered an important data set. NSL KDD has a reasonable number of training and testing results.

3.2 PRE-PROCESSING

Min-max standardization is used for preprocessing the data set in the proposed system. Data standardization is a tool of preprocessing data used in the mining of the data stream [10]. A data set attribute is normalized to include a small range, such as 0 to 1 by scaling its values. Normalization is especially useful for SVM classification algorithms. A number of data standardization methods include min-max normalization, z-score normalization, and decimal standardization.

3.3 CSO FEATURE SELECTION

Cuckoo Search is a meta-heuristic algorithm of optimization which is inspired by cuckoo species reproduction process. In nests of other host birds of various species, cucumbers lay their eggs. The host bird could then discover the unusual eggs in his nest and would either kill the eggs or leave the nest and build a new egg. This technique is improved instead of basic random walks by the so-called Levy flights. Compared with other methods, the advantage of CS is that, as GA, it is used for optimal enhancement in feature performance.

The CS is a novel heuristic evolutionary algorithm based on population output to overcome optimization issues. CS has the advantages of quick recognition and less conditions of power. This algorithm illustrates the mandatory breeding activity of cuckoo in conjunction with some birds and fruit fly Levy flight behavior. It has been used to address a wide range of real-world issues such as systemic optimization.

The CSO first uses the Levy test flights in the search area, and then uses the GWO change mode location to speed up the particles for the optimum solution convergence. At the same time, CS's random removal technique will effectively escape the local optima and maximize search efficiency for the optimal solution.

```
Input: $N$ - Number of Wolves, $N_{iter}$ - Number of iterations
Output: Optimal features
Step 1:  Initialize population of N wolves positions at random
Step 2:  Until stopping criteria met
Step 3:  Do
Step 4:  Get a cuckoo randomly by Levy flight
Step 5:  Evaluate individual fitness using fitness function
Step 6:  Find the best wolf position $\alpha$, $\beta$, $\delta$
Step 7:  Fractions of worse nests are abandoned and new ones are built;
Step 8:  Keep the best solutions;
Step 9:  Calculate the parameter $a$ with current iteration and maximum number of iterations
Step 10: For each $Wolf_i$ do
Step 11: Update the $Wolf_i$ position given the $\alpha$, $\beta$ and $\delta$ solutions, a and the current position $Wolf_i$
```

Step 12: Evaluate the new features

Step 13: End

3.4 MULTI-OBJECTIVE GREY WOLF OPTIMIZATION FEATURE SELECTION

GWO is a strategy for reducing wrapper attributes that is driven by filters that use wrapper-based classification accuracy and filter-based output. This GWO works in two phases therefore. GWO searches for a combination of attributes which maximizes the information of each other.

The fitness function shows the goodness of the combination of attributes and how well the attributes selected correctly categorize the class labels and are dependent on them. The convergence of the GWO depends on its search capacity and the fitness function of the GWO. For the number of iterations predefined, this process is stopped. Diversity is regulated through the discovery and extraction component and its limit varies from 2 to 1.

At the end of the first point, the population is a series of solutions which optimize mutual knowledge. The resulting population is used as the first solution for the optimization of the second level. In optimizing classification precision, the efficiency of GWO is as follows:

$$\text{Fitness} = CCR(D) \tag{1}$$

The precision of the SVM-Naïve Bayes classifier is increased in the second phase of classification. The measurement of human qualities takes more time than the first move. The search agents in the attribute space can thus be motivated with the attributes from the first phase. The combination works by using Naïve Bayes in the testing dataset *D* to evaluate the previous and conditional probabilities.

The probability of each class occurring in training data *D* is determined by counting the number of times. Number of occurrences of each attribute is calculated. Similarly, it can be determined how often an attribute value is contained in training dataset *D* for each attribute.

The algorithm then categorizes all examples with targeted class and conditions. The algorithm estimates the probability of these data in each class in order to classify data. The likelihood of the data in a class is the product of a conditional probability of a pre-class attribute value. For each class the rear likelihood is calculated and the data type is chosen with optimum rear likelihood. The class values change the maximum likelihood of a post-training outcome after description of training cases. All examples in the training dataset are followed by this.

Upon completion of the training with Naïve Bayes, the information is given to the target group SVM. In DoS, R2L, U2R and PROBE classification, the SVM classifies the attack type. In order to intensively find solution to better classification performance, this level of optimisation uses exploitation. GWO parameter for diversification and intensification is placed within the range 1 to 0 to maximize solution intensification. This choice of parameters allows fewer deviations from initial and second-stage solutions, and makes it possible to find guided solutions to the classifier.

3.5 CLASSIFICATION

In this section, SVM algorithm is proposed to improve the classification accuracy. It provides more accuracy results using the trained features.

Step 1: (Initialization) set initial value $f^{(0)}$ as the solution of SVM with labeled data and precision tolerance level $\varepsilon>0$

Step 2: (Iteration) at iteration k+1, solve, yields solution $f^{(k+1)}$.

Step 3: Obtain the selected features from CSO algorithm

Step 4: Run training and testing phase

Step 5: (Stopping rule). Terminate when $|s(f^{(k+1)} - s(f^k)| \leq \varepsilon$. Then the estimate f is the best solution among f^k; k=0,1,….

Step 6: Retrieve the accurate classification results

4. EXPERIMENTAL RESULT

The data collection of CUP 1999 data for examination of the algorithms for knowledge discovery and data extraction is given in this section. This dataset is the most trustworthy and reliable public benchmark to evaluate system algorithms for intrusion detection. In the dataset, the system connects 41 structures and 9 specific features and 32 continuous features. Attacks are descending into the following types in the dataset: Denial-of-Service, U2R: Illegal access to local super-user privileges, R2L: Illegal admission from a remote machine and Probe: Observation and added probing.

The detection accuracy using 41 features are shown in Table.1.

Table.1. Classification detection accuracy for CSO with SVM

Attacks	No. of records	Detection Accuracy (%)
DoS	1581	91
Probe	1902	93
U2R	1745	78
R2L	1745	83

The intrusion classification obtained using CSO with SVM is given in Fig.3.

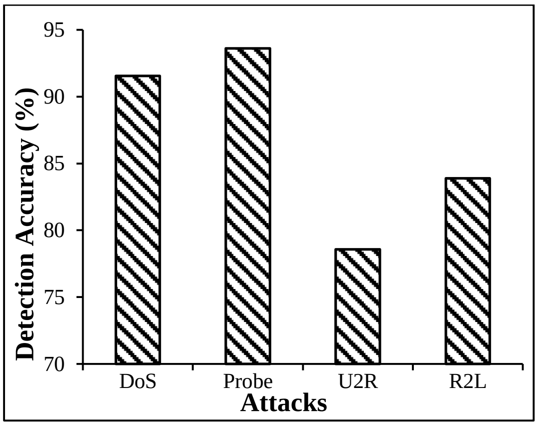

Fig.3. Classification detection accuracy of rule based classification

The results of detection accuracy for a rule based classification is given in Fig.3 and the results for 1581 records classification for DoS attack is given in Table.2.

Table.2. Time Analysis for Dos Attack in CSO-SVM

Features	**Time**	
	Selected features (18)	**Total features (41)**
1	2.57	2.62
2	2.41	2.53
3	2.38	2.42
Average	2.45	2.52

The time it takes to identify the DoS attack using CSO-SVM is given in Fig.4. The selected features are compared with the description of the 41 KDD cup dataset properties. The time taken to compute the DoS attacks with CSO-SVM is shown in Fig.4.

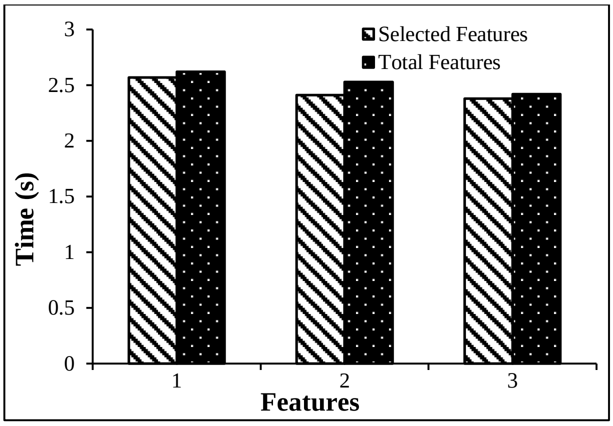

Fig.4. Computation time for DoS attack classification

Table.3. Time Analysis for Probe Attack in CSO-SVM

Features	**Time**	
	Selected features (10)	**Total features (41)**
1	4.56	7.42
2	4.38	7.08
3	4.12	6.99
Average	4.35	7.16

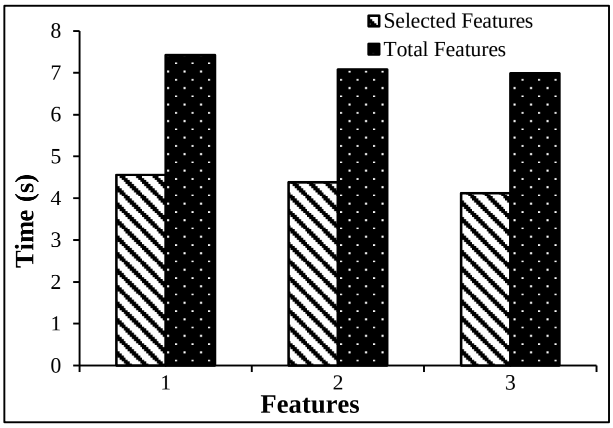

Fig.5. Computation time for probe attack classification

The computation time for record classification using rule based classification for a probe attack is given in Table.3.

The time it takes to identify the Probe attack using CSO-SVM is given in Fig.5. The selected features are compared with the description of the 41 KDD cup dataset properties. The time taken to compute the Probe attacks with CSO-SVM is shown in Fig.5.

Table.4. Time Analysis for U2R attack in CSO-SVM

Features	**Time**	
	Selected features (10)	**Total features (41)**
1	3.69	6.72
2	3.58	6.48
3	3.5	6.37
Average	3.59	6.37

The computation time for record classification using rule based classification for a U2R attack is given in Table.4.

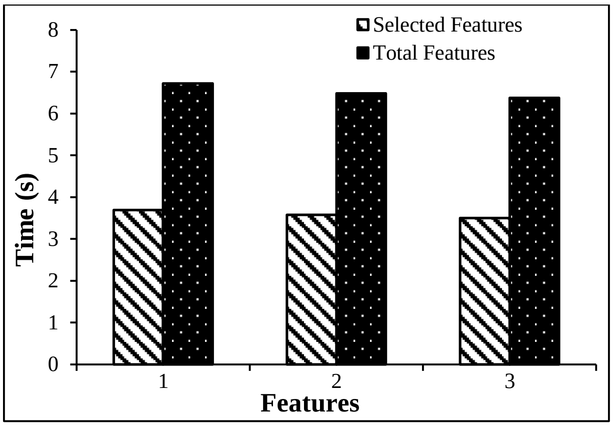

Fig.6. Computation time for U2R attack classification

The time it takes to identify the U2R attack using CSO-SVM is given in Fig.6. The selected features are compared with the description of the 41 KDD cup dataset properties. The time taken to compute the U2R attacks with CSO-SVM is shown in Fig.6.

Table.5. Time Analysis for R2L Attack in CSO-SVM

Features	**Time**	
	Selected features using OFS (10)	**Total features (41)**
1	3.74	7.09
2	3.68	6.82
3	3.62	6.61
Average	3.68	6.84

The computation time for intrusion classification using rule based classification for a R2L attack is given in Table.5.

The time it takes to identify the R2L attack using CSO-SVM is given in Fig.6. The selected features are compared with the description of the 41 KDD cup dataset properties. The time taken to compute the R2L attacks with CSO-SVM is shown in Fig.7.

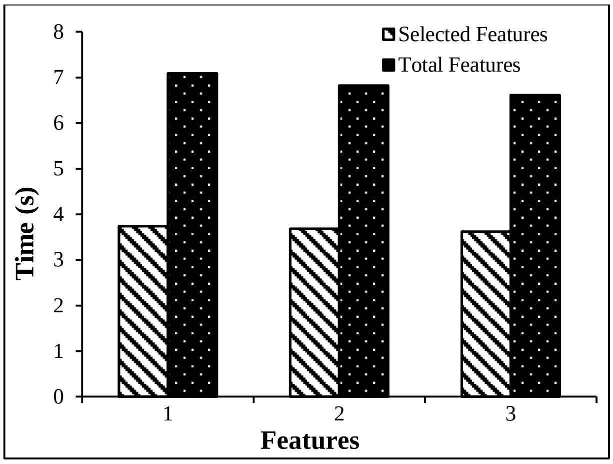

Fig.7. Computation time for R2L attack classification

4.1 PRECISION

Precision is the ration of true positives, and the sum of true positives and false positives against the intrusion and real features. It is defined as follows

$$Precision = TP/(TP+FP) \quad (2)$$

Table.6. Precision

Samples	SVM+PSO	GACO	GWO	IRVM+MFA	CSO-SVM
0	79.56	81.6	84.66	86.7	89.76
50	80.58	82.62	85.68	87.72	90.78
100	81.6	83.64	86.7	88.74	91.8
150	82.62	84.66	87.72	89.76	92.82
200	83.64	85.68	88.74	90.78	93.84
250	84.66	86.7	89.76	91.8	94.86
300	85.68	87.72	90.78	92.82	95.88

From Table.6, the precision comparison for the intrusion dataset is noted. The proposed CSO with SVM algorithm has higher precision than existing methods that accounts for better classification results.

4.2 RECALL

Recall value is computed to identify the data retrieval at TP prediction and FN prediction and it is defined as

$$Recall=\mathrm{TP}/(\mathrm{TP}+\mathrm{FN}) \quad (3)$$

Table.7. Recall

Samples	SVM+PSO	GACO	GWO	IRVM+MFA	CSO-SVM
0	66.3	68.34	70.38	72.42	74.46
50	67.32	69.36	71.4	73.44	75.48
100	68.34	70.38	72.42	74.46	76.5
150	69.36	71.4	73.44	75.48	77.52
200	70.38	72.42	74.46	76.5	78.54
250	71.4	73.44	75.48	77.52	79.56
300	72.42	74.46	76.5	78.54	80.58

From Table.7, recall comparison for the intrusion dataset is noted. The proposed CSO with SVM algorithm has higher recall than existing methods that accounts for better classification results.

4.3 SENSITIVITY

Sensitivity is the percentage of TP, which are intrusion and real feature classes. It is observed that the classified percentage of real and false intrusions and it is defined as:

$$Sensitivity = TP/(TP+FN) \quad (4)$$

where

TP – True positive,

FN – False negative,

TN – True negative, and

FP – False positive.

Table.8. Sensitivity

Samples	SVM+PSO	GACO	GWO	IRVM+MFA	CSO-SVM
0	76.9488	79.7844	83.2932	86.3532	88.74
50	77.2854	80.5698	84.4458	86.4858	89.76
100	77.8668	81.3654	85.0272	87.0672	90.78
150	79.2234	82.2834	86.6184	88.6584	91.8
200	79.5396	83.6196	86.8632	88.9032	92.82
250	80.121	84.201	88.3932	89.4132	93.84
300	80.2434	85.3434	89.5866	90.6066	94.86

From Table.8, sensitivity comparison for the intrusion dataset is noted. The proposed CSO with SVM algorithm has higher sensitivity than existing methods that accounts for better classification results.

4.4 SPECIFICITY

Specificity is the percentage of actual negatives related to negative class i.e. intrusion feature is classified as real feature and real feature are classified as intrusion feature.

$$\text{Specificity} =TN/(TN+FP) \quad (5)$$

Table.9. Specificity

Samples	SVM+PSO	GACO	GWO	IRVM+MFA	CSO-SVM
0	81.09	82.62	86.7	87.924	89.76
50	81.804	83.64	87.72	89.658	90.78
100	83.436	84.66	88.74	90.576	91.8
150	84.558	85.68	89.76	91.698	92.82
200	85.374	86.7	90.78	91.902	93.84
250	86.292	87.72	91.8	92.106	94.86
300	86.904	88.74	92.82	93.126	95.88

From Table.9, specificity comparison for the intrusion dataset is noted. The proposed CSO with SVM algorithm has higher specificity than existing methods that accounts for better classification results.

4.5 ACCURACY

Accuracy is the overall model correctness and it is the sum of actual classification parameters (*TP*+*TN*) separated by the total classification parameters (*TP*+*TN*+*FP*+*FN*)

$$Accuracy = (TP+TN)/(TP+TN+FP+FN) \quad (6)$$

Table.10. Accuracy

Samples	SVM+PSO	GACO	GWO	IRVM+MFA	CSO-SVM
0	86.4055	93.3644	94.2128	96.2328	93.3644
50	87.1933	94.5966	95.9399	96.9499	94.5966
100	88.6982	95.3137	96.3338	97.3438	95.3137
150	89.385	96.5762	97.0711	98.0811	96.5762
200	90.2233	97.1923	98.1215	99.1315	97.1923
250	91.3444	98.0912	98.6467	99.6567	98.0912
300	92.1322	98.2629	99.0911	100.1011	98.2629

From Table.10, accuracy comparison for the intrusion dataset is noted. The proposed CSO with SVM algorithm has higher accuracy than existing methods that accounts for better classification results.

5. CONCLUSION

In this paper, the accuracy of intrusion detection is indicated by CSO with SVM algorithm. Previous research does not ensure accurate results of attack detection and thus significantly reduces system performance. The hybrid optimization algorithm and efficient classification algorithm is proposed in the research to overcome the aforesaid problems.

The study includes modules such as preprocessing, feature selection and classification. Preprocessing is carried out using the min-max normalization approach to increase the accuracy of the attack detection. The feature selection process is then done through the use of a CSO algorithm to select optimal features from the dataset of the NSL KDD cup.

Better generation of better fitness values, best features are modified. The intrusion and normal features are classified more effectively from the dataset by using the SVM classification method. The research and evaluation model is used to create more detailed functionality from the chosen characteristics. Performance metrics such as precision, reminder, specificity, sensitivity and precision are evaluated. Higher performance metrics are available in the proposed CSO algorithm. The outcome of the analysis determines that the suggested CSO does not have current solutions but rather a superior performance.

In future, further threats will be found using sophisticated computation and classification algorithms.

REFERENCES

[1] Y. Zhao, "Network Intrusion Detection System Model based on Data Mining", *Proceedings of IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pp. 1123-1129, 2016.

[2] O. Koucham, T. Rachidi and A. Nasser, "Host Intrusion Detection using System Call Argument-based Clustering Combined with Bayesian Classification", *Proceedings of IEEE International Conference on Intelligent Systems*, pp. 1-8, 2015.

[3] S. Babu Devasenapati and K.I. Ramachandran, "Random Forest based Misfire Detection using Kononenko Discretiser", *ICTACT Journal of Soft Computing*, Vol. 2, No. 2, pp. 270-275, 2012.

[4] D. Amutha Guka, "Anomaly Detection in Networking using Hybrid Artificial Immune Algorithm", *ICTACT Journal of Soft Computing*, Vol. 2, No. 2, pp. 298-304, 2012

[5] S. Noel and S. Jajodia, "Advanced Vulnerability Analysis and Intrusion Detection through Predictive Attack Graphs", *Proceedings of IEEE International Conference on Armed Forces Communications and Electronics Association Solutions Series*, pp. 1-10, 2009.

[6] Safaa Zaman, Mohammed El Abed and Fakhri Karray. "Features Selection Approaches for Intrusion Detection Systems Based on Evolution Algorithms", *Proceedings of 7th International Conference on Ubiquitous Information Management and Communication*, pp. 1-7, 2013.

[7] Amira Sayed A. Aziz, Aboul Ella Hassanien and Ahmad Thaer Azar, "Genetic Algorithm with Different Feature Selection Techniques for Anomaly Detectors Generation", *Proceedings of IEEE International Conference on Computer Science and Information Systems*, pp. 8-11, 2013.

[8] M. Moradi and M. Zulkernine. "A Neural Network based System for Intrusion Detection and Classification of Attacks", *Proceedings of IEEE International Conference on Advances in Intelligent Systems-Theory and Applications*, pp. 1-12, 2004.

[9] M.A. Ambusaidi, X. He, Z. Tan, P. Nada, L.F. Nu and U.T. Nagar, "A Novel Feature Selection Approach for Intrusion Detection Data Classification", *Proceedings of IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 1-8, 2014.

[10] Y. Kumar and S.K. Bhandare, "Min Max Normalization based Data Perturbation Method for Privacy Protection", *International Journal of Computer and Communication Technology*, Vo. 2, No. 8, pp. 45-50, 2011.

[11] W. Hu and S. Maybank, "Adaboost-based Algorithm for Network Intrusion Detection", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 38, No. 2, pp. 577-583. 2008.