An Improved PSO and Random Forest for Intrusion Detection in Software Defined Networks

C. Aswini[1], Dr. M.L. Valarmathi[2], M. Nivedha[3], T. Ponsheka[4]
[1]*Assistant Professor, Government College of Technology, Coimbatore*
[2]*Professor, Dr. Mahalingam College of Engineering and Technology, Pollachi*
[3,4]*Student, Government College of Technology, Coimbatore*

***Abstract*— With the exponential growth of network for huge amount of data transmission, there exist an equal chance of network security issues as well. Software Defined Networks takes care of the network architecture intelligently and also controls them with software application. To make it more effective Intrusion Detection System (IDS) goes in hand with improved features to identify the network anomalies precisely with the help of machine learning concepts. This paper uses the Binary Bat algorithm for the selection of features which is done with the help of swarm division mechanism. Further for the process of flow classification, weighted voting mechanism has been used by altering the sample's weight by the Random forest method. The flow is classified intelligently with selected features to gives better performance result. Evaluation results prove that the modified intelligent algorithms select more important features and achieve superior performance in flow classification. It is also verified that the proposed system shows better accuracy with lower overhead compared with existing solutions. Thus the proposed system helps us in achieving more accurate results when compared to our existing methods.**

***Index Terms*— Binary Bat Algorithm, Swarm Division, Random Forest, Improved Particle Swarm Optimization**

I. INTRODUCTION

Software Defined Networks (SDN) is a paradigm that allows to know the logic behind the network's operation when the data's are being transmitted from source to destination. SDN technology is an approach to network management that enables dynamic programmatically efficient network configuration in order to improve network performance. SDN attempts to centralize every network into one network component. The control plane is considered as the brain of SDN network where the whole intelligence is incorporated by means of Open Flow Protocol. SDN are Directly Programmable, Agile, Centrally Managed and Programmatically Configured. Intrusion Detection System detect the possible intrusions in the network. Specifically it aims to detect computer attacks or misuse and to alert the proper individuals upon detection. IDS tools thus form an integral part of thorough and complete security system. It has essential security functions: they monitor, detect and respond to unauthorized activity by company insiders and outsider intrusion. Although intrusion detection systems monitor networks for potentially malicious activity, they are also disposed to false alarms. Hence, organizations need to fine tune their IDS products when they first install them. It means properly setting up the intrusion detection systems to recognize what normal traffic on the network looks like as compared to malicious activity. There are two detection methods of network intrusion. The first one is the Signature-based Method that detects the attacks on the basis of the specific pattern such as number of bytes or number of 1's or number of 0's in the network traffic. It also detects on the basis of the already known malicious instruction sequence that is used by the malware. The detected patterns in the IDS are known as signatures. The next type is Anomaly-based Method which was introduced to detect the unknown malware attack machine learning to create a trustful activity model and anything coming is compared with that model and it is declared suspicious if it is not found in model. Machine learning based method has a better generalized property in comparison to signature-based IDS as these models can be trained according to the applications and hardware configurations. In order to detect the intrusions in network a new technique is being proposed in this

paper which leverages the accuracy of the attacks. This paper proposes an AI-based two-stage intrusion detection empowered by software defined technology. It flexibly captures network flows with a global view and detects attacks intelligently. This IDS mechanism first leverage Bat Algorithm with Swarm Division and Binary Differential Mutation to select typical features. Then, a modified Random Forest is utilized by adaptively altering the weights of samples using the weighted voting mechanism to classify flows.

II. RELATED WORK

A brief study of earlier methods have been detailed as follows. SDN can be used to perform a security services, cloud integration and distributed application. In [1], the anomaly intrusion detection system combining fuzzy logic and neural networks. In [2], to optimize the membership function for mining fuzzy association rules authors apply the Genetic algorithm. It is not adoptive in real time. [3] SVM showed better performance than other classification technique such as fuzzy logic [4], k-nearest neighbor (KNN) [5], artificial neural network [ANN]. The performance of ids degrades due to the no of features of the audit data becomes larger, in terms of classification accuracy. To address these problem, Genetic algorithm [6] is used to improve the intrusion detection system based on SVM. It is used to supply fast and accurate optimization. However the error rate of SVM was not considered. This issue can be addressed by the combination of SVM and GA was proposed. In this stage of optimal feature selection, a new fitness function is used to decrease the error rate. Using SDN gateway, security approach [7] for IOT devices is proposed .It achieved the mitigation of DDOS attack by monitoring traffic flows. It can only perform well under specific condition. For this reason various Machine Learning technique with flow based classification have been adopted for solving problem [8]. In general, for selecting optimal features, a variety of swarm Intelligence (SI) algorithm, such as ant colony optimization (ACO) [9] and particle swarm optimization (PSO) [10], have been applied. For global optimization, Differential Evolution (DE) [11], [12], is one of the most popular evolutionary algorithm [EA]. After updating of bat, we apply mutation mechanism of differential evolution [13], which increase the diversity of the population and the ability of bats to jump out of local optima. To improve its performance, mutation operators can be modified in DE. Different mutation operators are self-adaptive differential evolution (SaDE)[14] and composite differential evolution(CDE) [15]. In SaDE, both mutation strategies and their associated control parameters values are adopted. In Composite DE, to generate new solution three mutation strategies and control parameters settings are randomly combined. [16] An improved swarm based k-means and is more effective to real world clustering. Compared to the association rules based detection approaches [17],[18],[19],[20],the Random Forest algorithm can process large dataset with many features efficiently. Random Forest [21] is an ensemble classification which is one of the most effective data mining technique and used to preventing over-fitting. The remainder of the paper is organized as binary bat algorithm, swarm division using IPSO-k-means algorithm and Random Forest for flow classification.

III. BINARY BAT ALGORITHM FOR FEATURE SELECTION

Micro bats uses the emission of loudness and pulse of sound and they search randomly for their target. Then identify their prey and after a few time they return back using echolocation. Echolocation works like a sonar. How far they are from a target can be computed by bats. In search space, each bat moves towards continuous valued position. In case the bat's motion is in n-dimensional Boolean lattice, there are many possibility of moving bat across the corners of hypercube. This leads to the problem for bat is to select or drop the given feature. Hence, an enhanced version known as Binary Bat Algorithm (BBA) is used. In BBA, the position of bat is denoted by binary vectors. BBA restrict only bat's position is always binary values using the following sigmoid function.

$$S(v_j^i) = \frac{1}{1+e^{-v_i^j}}$$

$x_i^j = 1\ if\ S(v_j^i) > \sigma$, 0 otherwise

where $\sigma \sim U(0, 1)$ represents the absence of features. Mutation mechanism algorithm lacks the potential, which leads to scarce position of the swarm. To solve

these problem, an enhanced the Swarm division mechanism is used.

IV. IMPROVED SWARM DIVISION

In this mechanism, the whole swarm is divided into subgroups using IPSO K means (Improved Particle Swarm Optimization K means). IPSO K means is based on the improved PSO with K means algorithm. This mechanism has two levels.

1. Present local minima is learnt by the general individuals within each subgroup and their velocity is altered to direct them, which is influenced by both local minima and their historical best positions.
2. Global optima is aimed by the local minima with higher fitness.

Initialize the pulse frequency, f_i, i=1, 2,…….m.

$f_i = f_{min} + (f_{max} - f_{min}).\beta$

Initialize the velocity v and position p of particles randomly. The centroid of cluster is called single particle. Population of particles is defined as

$P = \{X_1, X_2 \ldots\ldots X_m\}$

In search space, Xi is a single possible solution.

$X_i = \{C_{i,1}, C_{i,2} \ldots C_{i,m}\}$

Fitness of all particles X_i in population P was computed by Eqn 6. using the following objective function.

$$F(X_i) = \frac{k}{\left(\sum_{j=1}^{m} \sum_{i_k \in C_{i,j}} (i_k - C_{i,j})^2\right) + d}$$

Assign local best particle l best = p. Using L best particle we have to select the particle with best fitness value. New velocity V new of particle, L best and g best are computed as

$$Vnew_i^{(t+1)} = \lambda * V_i^{(t)} + C_1 * rand(l) * \left(l_{best\,i}^{(t)} - X_i^{(t)}\right) + C_2 * rand(l) * (g_{best}^{(t)} - X_i^{(t)})$$

Next position of particle P' by using P and v-new are generated as

$$X_i^{(t+1)} = X_i^{(t)} + Vnew_i^{(t+1)}$$

Loudness A_i and emission pulse rate r_i are updated for each iteration as

$$A_i^t = \alpha . A_i^{t-1}$$

$$r_i^t = r_i^0 . [1 - e^{-\gamma . t}]$$

Here α and γ are called ad-hoc constant. Initially Ai and ri values are zero. The bat perform the random walk to improve the variability of possible solution and it generate a new solution for each bat.

$$x_{new} = x_{old} + \delta . A_t^*$$

A_t^* Denotes the average loudness of all s at time t.

V. BUILDING THE MODEL WITH RANDOM FOREST

After selecting the features for each category (DOS, Probe, U2R, R2L), the decision tree classifier model is built using random forest. For building the model, the following parameters is considered: a) class-weight b)criterion c)max-depth d)max-features e)max-leaf-nodes f)min-impurity-split g)min-samples-leaf h)min-samples-split i)min-weight-fraction-leaf j)presort k)random-state l)splitter. This model is created for both the training set and testing set. The same parameters are used for both the training set and testing set.

The above mentioned parameters are described in detail:

1. class_weight=None(default)Weights associated with classes in the form {class_label : weight}. If None, all classes are supposed to have weight one.
2. criterion='gini'(default)The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity.
3. max_depth=None(default). The maximum depth of the tree. If None, then nodes are expanded until all nodes are pure.
4. max_features=None(default). The maximum number of features considered for the best split. If None, then max_features=n_features(the number of features when fit is performed).
5. max_leaf_nodes=None(default). Grow a tree with max_leaf_nodes. If None , then grow unlimited number of tress.
6. min_impurity_split=$1^{e-07.}$ Threshold for early stopping in tree growth. A node will split if its impurity is above the threshold, otherwise it is a leaf.
7. min_samples_leaf=1(default). The minimum number of samples required to be at a leaf node. If int, then consider min_samples_leaf as the minimum number.
8. min_samples_split=2(default). The minimum number of samples required to split an internal

node. If int, then consider min_samples_split as the minimum number.

9. min_weight_fraction_leaf=0.0(default). The minimum weighted fraction of the sum total of weights required to be at a leaf node. Samples have equal weight when sample_weight is not provided.
10. presort=False, This parameter is deprecated and will be removed.
11. random_state=0, Controls the randomness of the estimator.
12. splitter='best'(default). The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split.

VI. PREDICTION AND EVALUATION

After building the model, apply the classifier we trained to the test data. In training set, confusion matrix is created using all features for each category. But in testing set, confusion matrix is created only using selected features for each category. In confusion matrix, rows are named as "Actual attacks" and columns are named as "Predicted attacks". The KDD99 Dataset is used to evaluate various intrusion detection approaches. The dataset consists of 4,900,000 single connection vectors each of which contains 42 features and is labelled as either normal or an attack, with exactly one specific attack type. The attacks fall in one of the following four categories: DoS, Probe, R2L, U2R. A
python based SDN emulator named Mininet and Ryu controller is used for setting up the environment.
The following are the confusion matrix for each category for both the training set and testing set:
i.DoS

	Training set		Testing set	
Predicted attacks	0	1	0	1
Actual attacks				
0	9499	212	9602	109
1	2830	4630	2625	4835

ii. Probe

	Training set		Testing set	
Predicted attacks	0	2	0	2
Actual attacks				
0	2377	7374	8709	1002
2	212	2209	944	1477

iii. R2L

	Training set		Testing set	
Predicted attacks	0	3	0	3
Actual attacks				
0	9707	4	9649	62
3	2573	312	2560	325

iv. U2R

	Training set		Testing set	
Predicted attacks	0	4	0	4
Actual attacks				
0	9703	8	9706	5
4	60	7	52	15

The performance of an intrusion detection is evaluated in the light of precision(P), recall(R), F-score(F), Accuracy(A).
Training set

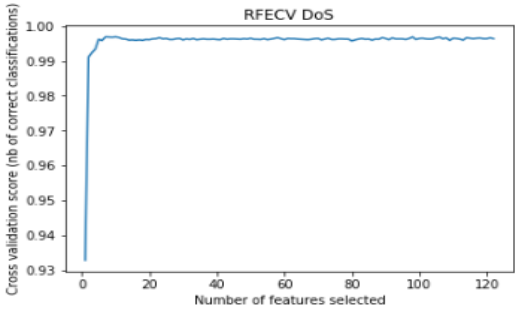
	Accuracy	Precision	Recall	F-measure
DoS	0.99639(+/-0.00341)	0.99505(+/-0.00477)	0.99665(+/-0.00483)	0.99585(+/-0.00392)
Probe	0.99571(+/-0.00328)	0.99391(+/-0.00684)	0.99267(+/-0.00405)	0.99329(+/-0.00512)
R2L	0.97952(+/- 000984)	0.97216(+/-0.01610)	0.96978(+/-0.01337)	0.97093(+/-0.01388)
U2R	0.99663(+/-0.00259)	0.86481(+/-0.08952)	0.91672(+/-0.10661)	0.88628(+/-0.07462)

	Accuracy	Precision	Recall	F-measure
DoS	0.99732(+/-0.00251)	0.99679(+/-0.00464)	0.99705(+/-0.00356)	0.99692(+/-0.00288)
Probe	0.99085(+/-0.00559)	0.98674(+/-0.01180)	0.98467(+/-0.01027)	0.98565(+/-0.00872)
R2L	0.97451(+/-0.00906)	0.96683(+/-0.01316)	0.96069(+/-0.01547)	0.96367(+/-0.01300)
U2R	0.99652(+/-0.00319)	0.87747(+/-0.15709)	0.89183(+/-0.17196)	0.87497(+/-0.11358)

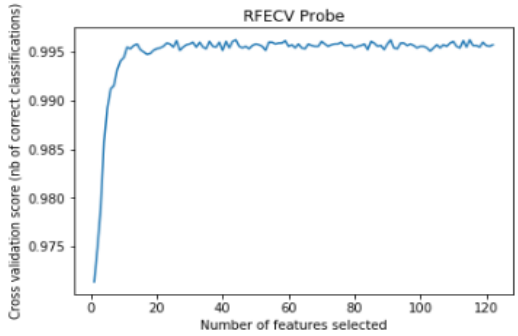
2.Testing set

VII. GRAPHICAL REPRESENTATION OF THE ACCURACY FOR DIFFERENT ATTACK DETECTION

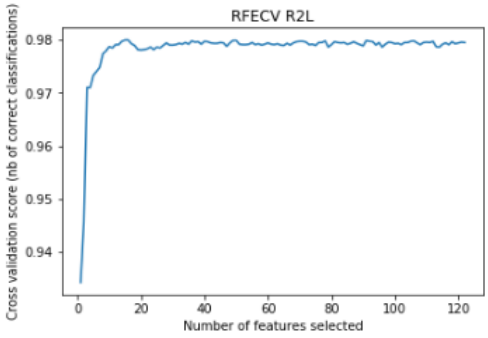
1.DoS



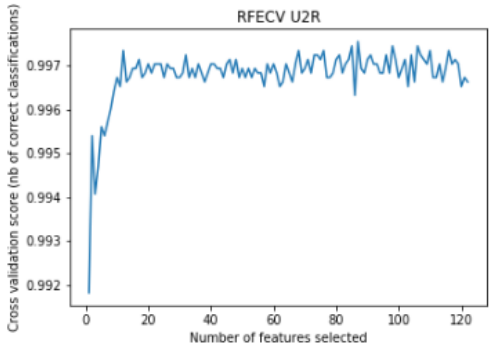
2.Probe



3.R2L



4.U2R



VIII. PICTORIAL REPRESENTATION OF DATASET WITH FEW ROWS AND COLUMNS

duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count
0	tcp	ftp_data	SF	491	0	0	0	0	0	...	25
0	udp	other	SF	146	0	0	0	0	0	...	1
0	tcp	private	S0	0	0	0	0	0	0	...	26
0	tcp	http	SF	232	8153	0	0	0	0	...	255
0	tcp	http	SF	199	420	0	0	0	0	...	255

dst_host_same_srv_rate	dst_host_diff_srv_rate	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate
0.17	0.03	0.17	0.00
0.00	0.60	0.88	0.00
0.10	0.05	0.00	0.00
1.00	0.00	0.03	0.04
1.00	0.00	0.00	0.00

dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	label
0.00	0.00	0.05	0.00	normal
0.00	0.00	0.00	0.00	normal
1.00	1.00	0.00	0.00	neptune
0.03	0.01	0.00	0.01	normal
0.00	0.00	0.00	0.00	normal

IX.CONCLUSION

Thus the proposed work proves a new and effective algorithm to detect the network intrusions intelligently and with high accuracy rate. Initially appropriate features is selected from the given input KDD cup datasets with the binary bat algorithm method. The features that match the most have been grouped together with the help of swarm division mechanism. After the succesful selection of the features the flow has been classified using random forest algorithm.We have introduced the weight of each sample and performed weighted voting mechnism to build the model. Hence when the testing dataset is projected with our proposed method the final results matched the threshold value. Thus, the output generated gave us high accuracy with lower over-heads.

REFERENCES

[1] M. Mohajerani, A. Moeini and M. Kianie, NFIDS: A Neuro-fuzzy Intrusion Detection System, Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems, 2003, pp348-351.

[2] W.D.Wang and S. Bridges, Genetic Algorithm Optimization of Membership Functions for Mining Fuzzy Association Rules, Proceedings of the 7th International Conference on Fuzzy Theory & Technology, Atlantic City, NJ,2000, pp131-134.

[3] T. Mehmood and H. B. M. Rais, “Svm for network anomaly detection using aco feature subset,” in Mathematical Sciences and Computing Research (iSMSC), International Symposium on. IEEE, 2015, pp. 121– 126.

[4] A. Chaudhary, V. Tiwari, and A. Kumar, “A novel intrusion detection system for ad hoc flooding attack using fuzzy logic in mobile ad hoc networks,”in Recent Advances and Innovations in Engineering (ICRAIE), 2014. IEEE,2014, pp. 1–4.

[5] S. Malhotra, V. Bali, and K. Paliwal, “Genetic programming and knearest neighbour classifier

based intrusion detection model," in Cloud Computing, Data Science & Engineering-Confluence, 2017 7th International Conference on. IEEE, 2017, pp. 42–46.

[6] M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in nsl - kdd cup 99 dataset employing svms," in Software, Knowledge, Information Management and Applications (SKIM]

[7] P. Bull, R. Austin, M. Sharma, and R. Watson, "Flow based security for iot devices using an SDN gateway," in IEEE International Conference on Future Internet of Things and Cloud, 2016, pp. 157–163A), 2014 8th International Conference on. IEEE, 2014, pp. 1–6.

[8] A. S. D. Silva, J. A. Wickboldt, L. Z. Granville, and A. Schaeffer-Filho, "Atlantic: A framework for anomaly traffic detection, classification, and mitigation in SDN," 2016, pp. 27– 35.

[9] T. Mehmood and H. B. M. Rais, "SVM for network anomaly detection using ACO feature subset," in International Symposium on Mathematical Sciences and Computing Research, 2016, pp. 121–126.

[10] N. Cleetus and K. A. Dhanya, "Multi-objective functions in particle swarm optimization for intrusion detection," in International Conference on Advances in Computing, Communications and Informatics, 2014, pp. 387–392

[11] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," J. Global Optim., vol.11, no. 4, pp. 341–359, 1997.

[12] K. V. Price, R. M. Storn, and J. A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization. Secaucus, NJ, USA: Springer-Verlag, 2005

[13] J. Wang, J. Liao, Y. Zhou, and Y. Cai, "Differential evolution enhanced with multiobjective sorting-based mutation operators," IEEE Transactions on Cybernetics, vol. 44, no. 12, pp. 2792–2805, 2017.

[14] A. Qin, V. Huang, and P. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," IEEE Trans. Evol.Comput., vol. 13, no. 2, pp. 398–417, Apr. 2009.

[15] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," IEEE Trans. Evol. Comput., vol. 15, no. 1, pp. 55–66, Feb. 2011.

[16] 1. Bai, L., Liang, J., Sui, C., Dang, C.: Fast global k-means clustering based on local geometrical information. Inf. Sci. 245, 168–180 (2013)

[17] D. Barbara, J. Couto, S. Jajodia, L. Popyack, and N. Wu, "ADAM: Detecting intrusions by data mining," in Proc. 2nd Annu. IEEE Workshop Inf. Assur. Secur., New York, Jun. 2001, pp. 11–16

[18] W. Lee and S. Stolfo, "A framework for constructing features and models for intrusion detection systems," ACM Trans. Inf. Syst. Secur., vol. 3, no. 4, pp.227–261, Nov. 2000.

[19] W. Lee and S. Stolfo, "Data mining approaches for intrusion detection," in Proc. 7th USENIX Secur. Symp., San Antonio, TX, Jan. 1998, pp. 79–83.

[20] W. Lee, S. Stolfo, and K. Mok, "Adaptive intrusion detection: A datamining approach," Artif. Intell. Rev., vol. 14, pp. 533–567, Dec. 2000.

[21] K. Singh, S. C. Guntuku, A. Thakur, and C. Hota, "Big data analytics framework for peer-to-peer botnet detection using random forest," Information Sciences, vol. 278, no. 19, pp. 488–497, 2014.