International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-9, July 2020

Tuning Random Forest Parameters using Simulated Annealing for Intrusion Detection

Ambikavathi C, S. K. Srivatsa

Abstract*: Intrusion detection system (IDS) security model is successfully utilized in static, distributed and dynamic network environments. Generally IDS needs a classification method for the decision of normal and abnormal events. This classification task is based on a set of features and massive amount of samples. However, all the features do not contribute the same level of prediction during classification. Hence feature selection (FS) has to be done before classification to select best features. Random forest (RF) does the dual role of FS and classification. Experiments have been done to prove that RF is the best classifier among other machine learning (ML) algorithms such as SVM classifier and C5.0 decision tree algorithm. However, the default parameter values of RF are not well suited for distributed environments such as cloud. It leads to poor accuracy and less efficiency in intrusion detection since enormous events have to be analyzed. So the parameters of RF have to be optimized by an efficient method. The important parameters of RF are number of trees, maximum depth of a tree, sample size, number of features considered to split a node (M_{try}), node size and maximum leaf nodes. Among these parameters the hyper parameters are selected based on three decision factors, randomness; split rule; tree complexity. The issues to be considered during parameter tuning are to avoid over-fitting and under-fitting. Therefore Simulated Annealing (SA) is utilized for tuning these hyper parameters of RF which leads to improve detection accuracy and efficiency of IDS. The idea of using SA for parameter optimizing process is to avoid those issues since it never struck in local optimum. The proposed system significantly boosts the results of IDS. The efficiency of the proposed SA-RF is validated using CICIDS2017 dataset.

Keywords: Cloud computing, Intrusion Detection System, Random Forest, Simulated Annealing.

I. INTRODUCTION

Cloud computing is a blooming research domain that has provided various benefits to all the fields such as business, education and communication technology. Consumers can access resources without direct active management of those resources. The usage of cloud computing is increasing exponentially because it is beneficent to both cloud service provider (CSP) and consumers.

Revised Manuscript Received on July 30, 2020.
***** Correspondence Author
Ambikavathi*, Research Scholar, Faculty of S&H(Computer Applications), Sathyabama University, Chennai, India. Email: abi.kumar79@gmail.com[1]
S. K. Srivatsa, Professor (Rtd.), MIT, Anna University, Chennai, India. Email: profsks@rediffmail.com[2]

However, the efficiency and security are the two important aspects of private and public cloud. IDS has proved as a major element for providing network security. This security model has been employed to observe and analyse the incidents which mainly crosses or abuse the computer security guidelines in computers and networks. An important issue in this security system is the poor accuracy in detection rate because IDS has to process massive amount of network data. So IDS face the problem of processing data using right classifier [1]. Intrusion detection is generally done using data mining concepts such as data preprocessing, selection, transformation, pattern matching etc. The intention is to identify the user behavior using samples based on the information in features [2]. In addition to this, unbalanced datasets are tough to tackle [3]. Massive and unbalanced data usage leads the data mining algorithms such as SVM (Support Vector Machine), RF leads to degrade the intrusion detection results. In addition, the availability of more features, relevancy among features is also a major issue. It reduces the classification accuracy with high computational complexity. RF has proved itself as the best classifier among its competitors. RF also performs FS, is an added benefit. However, cloud based IDS adds more and more events for detection which leads to heavy weighted trees for RF. The default parameter values of RF are well and good for static environments. However, they provide variations in classification accuracy as well as in feature importance for distributed and dynamic environment such as cloud. There are many ways to tune the RF parameters. tuneRF, tuneranger, mlrhyperopt, caret are some of them. 'tuneRF' only tunes M_{try} parameter whereas 'tuneranger' tunes three parameters which are node size, sample size and M_{try}. 'mlrhyperopt' tunes two parameters: node size and M_{try}. 'Caret' package tunes number of trees and M_{try} parameters. The hyper parameters tuned in our work are M_{try}, node size, sample size and number of trees. All the mentioned parameters are interrelated. Optimizing single parameter never yields good results. Since they are correlated strongly, they have to be improved as a set of values simultaneously.

The approaches for tuning hyper parameters of any classifier are grid search and random search. Grid search method comprehensively searches the search space which gives best results but takes more execution time. Grid search is not applicable when there are more numbers of features with more possible values. Random search method searches randomly for parameters within less time but not yields best results. Since random values are chosen, it may miss better values of parameters.

RF has also been applied with both grid and random search methods. In this paper, the issue is resolved by optimizing the parameters of RF using SA. SA is a slow and steady stochastic optimization algorithm used to find the best parameter values in a search space.

The outstanding portion of the paper is formulated as follows: Section II briefs related research works and background of the proposed work. Section III presents the proposed parameter tuning model procedure. Section IV discusses the results of tuned model with default parameter values. Finally, concluded the work with paper highlights and future direction.

II. LITERATURE REVIEW

Various works have been done by researchers on IDS and applied ML algorithms such as SVM, C5.0 and RF for FS and classification to change over traditional IDS appropriate to cloud environment. As well as ML algorithms' default parameters are optimized using suitable techniques.

A collaborative N-IDS using SVM classifier in [4] is proposed for detecting anomalies. It integrates anomaly and signature based methods. SNORT is used for signature based detection. They deployed NIDS sensors at the front end and VM monitors at the back end. Finally the alerts are in the central log database. Authors of [5] have proposed online anomaly IDS for malware detection in cloud. They used one class SVM classifier at the hypervisor level. System and network level features of a cloud node are gathered for detection. SVM and C5.0 algorithm are combined together for efficient intrusion detection in [6]. RF has been effectively used for FS and classification for intrusion detection [7-9]. A separate list of hyper parameters of RF that influences FS and classification performance has been given in [10].The hyper parameters influencing classification performance are number of trees, M_{try}, node size and splitting rule. The hyper parameters influence variable importance are M_{try}, sample size, replacement, node size, number of trees and splitting rule. Also strategies for tuning those parameters have been explained. Sequential model based optimization (SMBO) is used to optimize M_{try}, node size and sample size parameters parallel. Finally the comparison between SMBO and default hyper parameters shows that there is an improvement in prediction performance. The effect of splitting rule on RF is discussed in [11]. The author proposed weighted splitting rules to possess end cut preference (ECP) property. He proved that randomized adaptive splitting, instead of pure random split, yields computationally efficient results. Performance has been measured by Brier score and 10 fold cross validation. The important dependent parameters of RF in [12] are number of trees, subsample size, tree depth and M_{try}. Also concluded that optimizing both tree depth and subsample size, is not needed since optimizing any one of these two parameters gives same performance gain as optimizing both. SA is mostly used by optimization problems by many researchers. FS is considered as an optimization problem to select optimal features and used SA in [13] and [14]. A hybrid solution combining spotted hyena optimization (SHO) with SA is proposed in [13] for FS problem. They have evaluated two different hybrid models. In the first model SHOSA1, SA is embedded with SHO at each iteration level. In the second model SHOSA2, SA is applied at the end of SHO iterations. They found that SHOSA1 performs better than SHOSA2 and other native algorithms. In [14] another hybrid is done by combining whale optimization algorithm (WOA) with SA for feature selection in order to gain best classification results. SA is added with WOA to improve the exploitation ability of WOA. SA is used to find the best solution in the neighborhood after each step of WOA. The initial temperature and the cooling temperature have been set to 2*no. of attributes and T*0.93 respectively.

III. PROPOSED WORK

The aim of this presented work is to make IDS as a strong hazard against hackers and intruders. The best classification method for intrusion detection is found initially. SVM classifier, C5.0 decision tree algorithm and RF algorithm are the three classifiers used for identifying the best one. After data preprocessing, training and testing sets are created by segregating total samples in 70:30 ratios. Each classifier parameters are set and invoked to build the model. Predictions are made using the model. RF gives the best classification results compared to other two. However, the default parameter values of RF have to be tuned to improve detection accuracy.

A. Hyper parameters of RF

The hyper parameters focused in this paper are based on the factors: Randomness; Splitting rule and Tree complexity.

1) Randomness

Randomization takes place in RF at two places of training phase. Random samples of the dataset are used to train each tree and random subset of features is used at each node split. M_{try}, node size and sample size are the parameters used to control the randomness nature of RF [10]. Randomness in RF is regularized implicitly by the parameter M_{try} [15].

2) Splitting Rule

Splitting rule by default depends on M_{try} parameter and Gini impurity measure, also called as Gini index, given in (1), is the degree of probability of features being wrongly classified. 'C' is the number of normal and attack events and p(i) is the probability of event 'e' in the dataset. This measure possesses ECP property [11].

$$GI = \sum_{i=1}^{C} p(e) * (1 - p(e)) \qquad (1)$$

M_{try} based on splitting rule, is the number of features considered to split a node in a tree. The default value of M_{try} depends on number of features calculated as in (2) for classification, where 'f' represents number of features in the dataset. Depending on feature relevancy M_{try} value is set. Feature relevancy and M_{try} are inversely proportional to each other.

$$\text{Mtry} = \sqrt{\text{f}} \qquad (2)$$

3) Tree complexity

Tree depth, node size, sample size, M_{try} and number of trees are the parameters impacts tree complexity factor. More number of trees yields more classification accuracy and produces stability in feature importance.

However, it should be balanced with the computation time and error rate. Out-of-bag (OOB) error rate is used to tune both M_{try} and number of trees parameters. Optimal number of decision trees in the forest can be obtained by optimizing node size and sample size parameters.

Hence by considering the above factors, M_{try} is the most important parameter since it influences all the three factors. Next node size and sample size impact both factors randomness and tree complexity. Finally number of trees in the forest is also chosen. The hyper parameters to be tuned are represented as a set HP_t with four variables as in (3).

$$HP_t = \{M_{try}, node\ size, sample\ size, no.of\ trees\} \quad (3)$$

Tuning each parameter separately never produces better classification results since all the parameters are dependent. Each parameter in HP_t should be balanced with other three parameters. So we propose parameter tuning method using SA for RF.

B. Tuning using SA

Simulated Annealing is a randomized local search algorithm. It is used to find optimal parameters in a discrete search space with large number of combination of hyper parameters. It is used to define combinatorial optimization problems by complex objective functions. Here SA is assumed as a search method, a probabilistic way to identify the important features for attaining better predictors. The required initial solution of SA is randomly selected. A main factor that influences the results of the SA is to define the suitable cost function to evaluate individual solutions [16].The cost function $c(p)$ is defined as in (4), where 'N' is number of samples in the training set and 'M' is number of features.

$$c(p) = \frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{M} e^{ij}} \quad (4)$$

The pseudo code of the parameter tuning based on SA is given as algorithm steps.

Algorithm: Parameter tuning using SA

Input:
Combination of parameter values $HP_t = \{P_{s1}, P_{s2}\ldots P_{sn}\}$
Output: Optimized values: P_b

1. $P_b \leftarrow \{\phi\}$;
2. $T_i \leftarrow 10000$;
3. $C \leftarrow 0.9999$;
4. Randomly choose a set of initial hyper parameter values as P_i;
5. $P_b \leftarrow \{P_{si}\}$;
6. Find the cost of initial solution, Cost(P_b);
7. **While** ($T_i > 0.1$) do
8. $P_n = \{P_b\}+1$;
9. **If** (Cost(P_n) <= Cost(P_b))
10. $P_b \leftarrow P_n$;
11. **Else**
12. r = rand(0,1)
13. **If** ($r < e^{-\frac{cost(P_n)-cost(P_b)}{T_i}}$)
14. $P_b \leftarrow \{P_n\}$
15. End if
16. $T_i \leftarrow C * T_i$
17. **End While**

The tuning process using SA operates as follows: Initial temperature T_i is assigned as 10000 since the four parameters takes ten different values. The cooling or reducing criteria level is assigned as 0.9999 in order to reduce T_i by 1. First, a set of hyper parameter values is randomly chosen as an initial solution and it is considered as an optimal one. Next, the cost of this optimal solution is determined by the cost function defined in (4). While the temperature T_i does not meet the stopping criteria, next set of the parameter values of optimal solution is chosen and its cost is determined. When the cost of the currently chosen solution is lesser than or equal to that of the present optimal solution, then it is replaced with the recently chosen new solution. When the cost of the nearby solution is higher than the present optimal solution, a random value 'r' is selected in the range of 0 to 1. In this case, the optimal solution set will be replaced by new set only when the random value $r < e^{-\frac{Cost(p_n)-Cost(p_b)}{T_i}}$. Then T_i is reduced by cooling criteria and the algorithm steps from line no.8 to 15 will be repeated until the criteria $T_i > 0.1$ is true. Finally P_b contains the best set of hyper parameter values. These parameter values are used by RF in FS and classification.

IV. RESULTS AND DISCUSSION

A. Dataset

The implementation is done in R language using CICIDS2017 dataset [17]. CICIDS2017 dataset is presented by Canadian Institute of Cyber security offered by University of New Brunswick (UNB) described with 83 features. This dataset is built by capturing network data for five days with recorded abstract behavior of 25 users. The data set comprises of ten features which are complete network configuration, complete traffic, labeled data set, complete interaction, complete capture, protocols, attack diversity, heterogeneity, feature set and metadata. It comprises of 3.1 million flow records recorded using CICFlowMeter. It contains pcap files and a .csv file with attack and normal samples. This .csv file covers 15 classes of attack samples such as Botnet, Brute force, DDoS, DoS, FTP, Infiltration, Port scan, SQL injection, SSH and XSS. Table I, shows the preprocessed dataset extracted from the dataset for validating our results. This extracted dataset covers six types of attacks and normal samples. The attacks covered are DDoS, DoS, FTP-patator, Infiltration, port scan and SSH-patator. Training and testing sets are constructed in the ratio 70:30 from the dataset.

Table- I: Pre-processed dataset

Class Type	Total entries	Training set	Testing set
DDoS	4629	3239	1390
DoS	999	693	306
FTP-patator	7935	5584	2351
Infiltration	36	26	10
Normal / Benign	39276	27419	11857
PortScan	11189	7870	3319
SSH-patator	2974	2095	879
Total	**67038**	**46926**	**20112**

B. Classification results

The system is tested using classification algorithms such as RF, C5.0 and SVM. We have implemented our approach with three packages of R programming which are caret package for SVM, randomForest package for RF classifier and C50 package for C5.0 algorithm. 'caret' package in R programming is utilized for experimentation of SVM classifier. The traincontrol() method is called to set control parameter to five fold and number of repeats to 3. train() method is invoked with the parameters: preprocessed dataset, data as training partition, method as 'SVM linear kernel', traincontrol, preprocess as combination of 'center' and 'scale', and tunelength is set as '3'. Table II shows the confusion matrix of the SVM classifier on the test set. Winnowing is applied in the dataset to identify the features' importance for extracting only needed features. Boosting is applied using 'trials' option of C5.0() function, assigned to a value of 10. A decision tree and rule set is constructed for the training dataset. Table III shows the confusion matrix of the C5.0 classifiers' prediction on the test set. It depicts that C5.0 is better than SVM.

randomForest() function is invoked with the parameters preprocessed dataset, training partition, number of trees as 25 and M_{try} as 7. M_{try} parameter is selected using tuneRF() method which is used to find the best M_{try} value. It depicts the error rate is low when M_{try} value is 6 or 7. Fig.1 shows the relationship between tuned M_{try} value and OOB error rate. Table IV shows the confusion matrix of the RF classifier on the test set.

Table- II: Confusion matrix of SVM classifier

	DDoS	DoS	FTP- patator	Infiltration	Normal	Port Scan	SSH- patator
DDoS	1388	0	0	0	0	0	0
DoS	0	160	0	0	7	0	0
FTP-patator	0	0	2375	0	11	0	6
Infiltration	0	0	0	9	4	0	0
Normal	0	134	2	0	11535	20	6
PortScan	0	0	0	1	38	3336	3
SSH-patator	0	5	3	0	187	0	877

Table- III: Confusion matrix of C5.0 classifier

	DDoS	DoS	FTP- patator	Infiltration	Normal	Port Scan	SSH- patator
DDoS	1403	0	0	0	0	0	0
DoS	0	287	0	0	3	0	0
FTP-patator	0	0	2405	0	0	0	0
Infiltration	0	0	0	12	0	0	0
Normal	0	12	1	3	11742	3	2
PortScan	0	0	0	0	0	3370	0
SSH-patator	0	0	0	0	1	0	868

Table- IV: Confusion matrix of Random forest

	DDoS	DoS	FTP- patator	Infiltration	Normal	Port Scan	SSH- patator
DDoS	1390	0	0	0	0	0	0
DoS	0	304	0	0	1	0	0
FTP-patator	0	0	2351	0	0	0	1
Infiltration	0	0	0	10	0	0	0
Normal	0	2	0	0	11856	2	3
PortScan	0	0	0	0	0	3317	0
SSH-patator	0	0	0	0	0	0	875

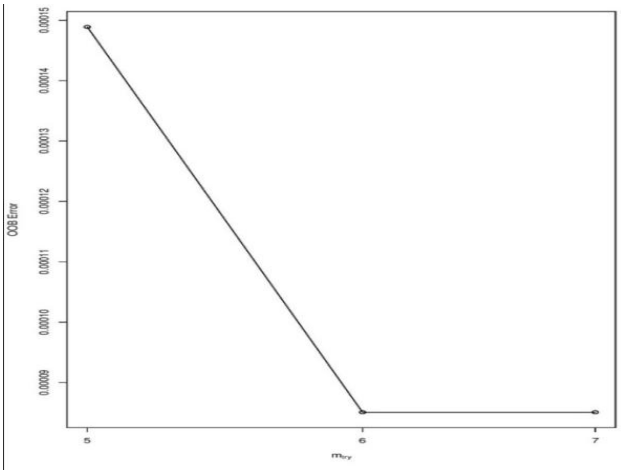

Fig. 1. Tuned M_{try} parameter

By viewing the confusion matrices, it can be concluded that RF is the best classifier comparing the other two ML algorithms. Table V shows the comparison of classification algorithms through the metrics accuracy, recall and precision. Even though increase in number of events leads RF to degrade its performance.

Table- V: Comparison of classifiers

Metrics	SVM	C5.0	RF
Accuracy	97.88	99.88	99.95
Precision	94.19	99.98	99.96
Recall	98.25	99.78	99.90

C. Tuning results

The initial temperature is set as 10000 and the cooling criteria level is 0.9999. The default parameter values are chosen as the initial solution of SA which is defined as:

$$HP_t = \{M_{try} = 7, node\ size = 1, sample\ size = 1, no.\ of\ trees=25$$

It is assigned as best solution and cost of it is found using cost function. The next parameter combination is selected from the set HP_t. The cost of this current solution is compared with best solution. This procedure is continued as per the algorithm procedure. 10 different values of each parameter is tested which gives 10^4 =10,000 combinations. Finally the optimal hyper parameter set values achieved is:

$$HP_t = \{M_{try} = 4, node\ size = 3, sample\ size = 0.7, no.\ of\ trees = 20\}$$

Now RF classifier is trained and tested with the tuned parameter values. The error rate of each class of prediction with the number of trees is traced. Table VI shows the error rate of each attack and normal class before and after tuning the RF parameters. Fig.2 and Fig.3 illustrate the error rate with the number of trees for each type of attack in the dataset. The error rate has been reduced after tuning the parameters. It depicts that the infiltration attack has more error rate compared to others. It also illustrates that the error rate is less when 'number of trees' is '20'. DDoS and FTP-patator attacks have gained 100% accuracy.

Table- VI: Classwise comparison of error rate before and after tuning

Class	Number of trees	Error rate	
		Before tuning	After tuning
DDoS	5	0.0001	0.0000
	10	0.0001	0.0000
	15	0.0000	0.0000
	20	0.0000	0.0000
DoS	5	0.0512	0.0442
	10	0.0421	0.0351
	15	0.0465	0.0345
	20	0.0231	0.0105
FTP-patator	5	0.0009	0.0007
	10	0.0087	0.0004
	15	0.0094	0.0000
	20	0.0005	0.0000
Infiltration	5	0.4446	0.3556
	10	0.3545	0.1651
	15	0.3178	0.2145
	20	0.3321	0.2083
Normal	5	0.0645	0.0445
	10	0.0556	0.0345
	15	0.0541	0.0341
	20	0.0004	0.0002
PortScan	5	0.0254	0.0001
	10	0.0045	0.0001
	15	0.0007	0.0001
	20	0.0005	0.0000
SSH-patator	5	0.0215	0.0012
	10	0.0421	0.0001
	15	0.0452	0.0055
	20	0.0044	0.0004

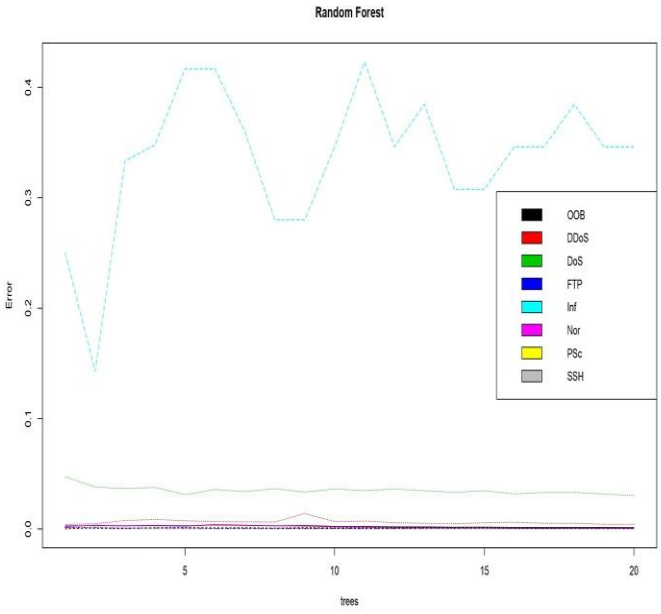

Fig. 2. No. of trees Vs Error rate(Before optimization)

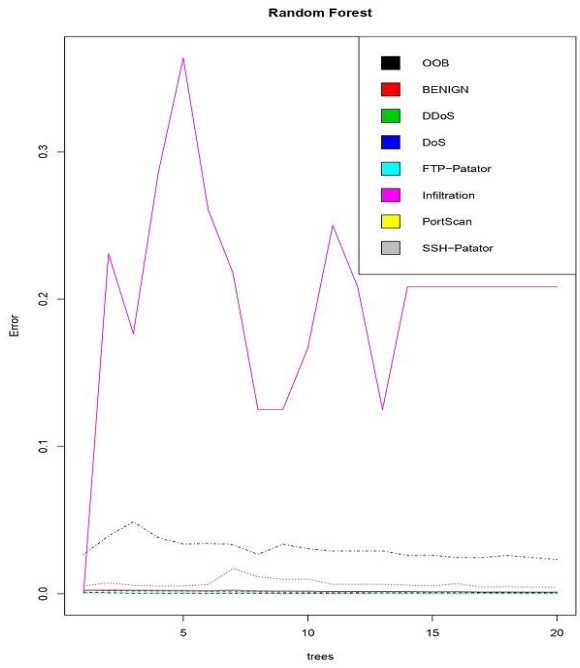

Fig. 3. No. of trees Vs Error rate(After optimization)

V. CONCLUSION

It has been proved that RF is the best classifier compared to SVM and C5.0. The most important parameters of RF are selected based on the factors randomness, splitting criteria and tree complexity. The selected hyper parameters are M_{try}, sample size, node size and number of trees. Then these parameters are optimized using probabilistic search algorithm SA. Generally tuning parameters of machine learning algorithms lead to achieve efficiency of the algorithm. The parameter value which minimizes the error rate is selected using cost function. In this paper, the efficiency of RF algorithm is improved by tuning its hyper parameters using SA. The proposed work recovers RF from bad split. Over-fitting and under-fitting problems of RF are avoided through parameter tuning. Also RF is tuned to gain performance efficiency even with an imbalanced dataset as used in this paper. In future, the proposed work can be evaluated using various performance metrics.

REFERENCES

1. Singh, R., Kumar, H., Singla, R.K., "An intrusion detection system using network traffic profiling and online sequential extreme learning machine". Expert Syst. Appl. 42, 8609–8624, 2015.
2. Ghanem, W.A.H.M., Jantan, A., "Novel multi-objective artificial bee colony optimization for wrapper basedFS in intrusion detection". Int. J. Adv. Soft Comput. Appl. 8, 70–81, 2016.
3. Moayedikia, A., Ong, K.L., Boo, Y.L., Yeoh, W.G., Jensen, R. "FS for high dimensional imbalanced class data using harmony search". Eng. Appl. Artif. Intell. 57, 38–49, 2017.
4. Al Haddad. Z, Hanoune. M and Mamouni. A, "A Collaborative Network Intrusion Detection System (C-NIDS) in Cloud Computing", International Journal of Communication Networks and Information Security, Vol.8, No.3, pp.130, 2016.
5. Watson. M.R, Marnerides. A.K, Mauthe. A and Hutchison. D, "Malware Detection in Cloud Computing Infrastructures", IEEE Transactions on Dependable and Secure Computing, Vol.13, No.2, pp.192-205, 2015.
6. Golmah. V, "An efficient hybrid intrusion detection system based on C5. 0 and SVM." International Journal of Database Theory and Application, Vol.7, no. 2, 2014, pp.59-70.
7. Farnaaz N and Jabbar M, "Random Forest Modeling for Network Intrusion Detection System", Procedia Computing Science, Vol.89, 2016, pp.213-217.
8. Hasan M A M, Nasser M, Ahmad S and Molla K I, "Feature Selection for Intrusion Detection Using Random Forest", Journal of information security, Vol.7, 2016, p.129.
9. Singh K, Guntuku S C, Thakur A and Hota C, "Big Data Analytics Framework for Peer-to-Peer Botnet Detection Using Random Forests", Inf Sci, 278, 2014, pp.488- 497.
10. Probst, P., Wright, M.N. and Boulesteix, A.L., 2019. "Hyperparameters and tuning strategies for random forest". Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9(3), p.e1301.
11. Ishwaran, H.,."The effect of splitting on random forests". Machine learning, 99(1), 2015, pp.75-118.
12. Scornet. E, "Tuning parameters in random forests", ESAIM: Proceedings and Surveys, Vol. 60, 2017, pp.144-162.
13. Jia H, Li. J, Song. W, Peng. X, Lang. C and Li. Y, "Spotted Hyena Optimization Algorithm with Simulated Annealing for Feature Selection", IEEE Access, Vol.7, 2019, pp.71943-71962.
14. Mafarja. M.M and Mirjalili. S, "Hybrid Whale Optimization Algorithm with Simulated Annealing for Feature Selection", Neurocomputing, Vol.260, 2017, pp.302-312.
15. Mentch, L. and Zhou, S., "Randomization as Regularization: A Degrees of Freedom Explanation for Random Forest Success". arXiv preprint arXiv:1911.00190, 2019.
16. Kang, S.-H., and Kim, K. J. A "FS approach to find optimal feature subsets for the network intrusion detection system", Cluster Computing, doi: 10.1007/s10586-015-0527-8, 2015.
17. Iman. S, Arash. H. L and Ali. A.G, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", Proceedings of 4th International Conference on Information System Security and Privacy, 2018, pp.108-116.

AUTHORS PROFILE

Mrs. C.Ambikavathi received her under graduate B.Sc. (Mathematics), post graduate MCA and M.Phil, from Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India. Currently, she is pursuing Ph.D. in Sathyabama University, Chennai, India. Her research domain is cloud security. She has 15 years of teaching experience in both under graduate and post graduate levels. At present, she is working as an assistant professor in Hindustan College of Arts & Science. She has presented research papers in three national conferences and three international conferences. She has also published four research papers in national and international journals. Her research interests are machine learning, cloud computing and network security.

Dr. S K Srivatsa received the Bachelor of Electronics and Telecommunication Engineering degree from Jadavpur University, Calcutta, India, Master's degree in Electrical Communication Engineering and Ph.D. from the Indian Institute of Science, Bangalore, India. He was a Professor of Electronics Engineering in Anna University, Chennai, India and has 39.5 years of Post-Doctoral teaching experience. He was a research associate at Indian Institute of Science. He has taught twenty eight different courses at under graduate and forty two courses at post graduate level. His current research activities pertain to computer networks, artificial intelligence and robotics. He has produced more than 100 Ph.D.'s and is the author of over 800 publications.