International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

June 2022 | Issue 51 | Volume 14 | Number 2 | Pages 275-283

OSS-RF: INTRUSION DETECTION USING OPTIMIZED SINE SWARM BASED RANDOM FOREST CLASSIFIER ON UNSW-NB15 DATASET

J. Vimalrosy S. Brittorameshkumar

Department of Computer Science, St. Joseph's College, Bharathidasan University, Trichy, India
rschlrvimalarosy@yahoo.com, dr.sbritto@outlook.com

Abstract- An IDS is termed an Intrusion Detection System and is considered a security management system. The network traffic is monitored, the system is scanned for malicious activities, and the security administrator is alerted. Intrusion detection systems in cloud environments suffer from lower detection accuracy, higher computational time, and higher false-positive rates. Hence, this paper proposes an effective mechanism for detecting and classifying the attacks taken from the UNSW-NB15 dataset. Specifically, to enhance the accuracy, the feature selection process is performed using the Optimized Sine Swarm algorithm (OSS), which selects only the significant features. Finally, the intrusion classification is performed using the RF (Random Forest) classification. The proposed OSS-based RF classifier was evaluated using different performance metrics and compared with various existing models to prove its efficiency. The research results show that the proposed model is very good at detecting and classifying threats.

Keywords: Intrusion Detection, UNSW-NB15 Dataset, Optimized Sine Swarm Algorithm, Random Forest.

1. INTRODUCTION

Privacy and security are two of the most pressing issues while using a cloud computing service. The purpose should be to identify IDS in the cloud. Cloud attacks include business data misuse, unauthorized access to legal resources, dangerous software or worms, and intellectual property theft. Putting an IDS on a cloud platform has certain drawbacks. As a result of multiple customers sharing the same physical resources and cloud servers, virtual machines have become a significant source of traffic congestion for internet providers. The primary concern is with IDS' dynamic performance. Large volumes of data are becoming increasingly unsuitable for traditional intrusion detection systems (IDS). As a result, additional research into IDS platform performance difficulties is required [1].

Particular specific challenges need to be handled in IDS deployment in the cloud platform. There is a need for researchers to increase detection rates and devise new detection methods. Anomaly or biometrics detection mechanisms are both viable options for IDS.

Known attack patterns detect only the signature-based IDS, which cannot identify zero-day attacks or unseen attacks. The signature definition is considered a cumbersome task for all previously identified attack patterns and requires expert knowledge. Anomaly-based IDS has detected new attacks based on observed instances with behavioral analysis [2].

There are multiple challenges related to IDS approaches in a cloud platform like normal dynamic behavior, public attack dataset unavailable, and hackers discovering specific ways to show the attack packet as a regular packet. For the effective detection of intrusion, machine learning techniques can be implemented in cloud platforms that deal with cloud services' dynamic behavior changes. Machine learning algorithms considered Neural Network, c4.5 Decision tree, and Naïve Bayes in cloud environments considered open research areas. Furthermore, cloud servers process massive amounts of data every minute. The IDS mechanism must be improved to detect suspicious activities quickly. A cloud environment can be augmented by parallel processing approaches [3]. Thus, this study concentrated on two issues: IDS performance and cloud platform detection mechanism.

A unique OSS feature selection approach with a random forest classifier is suggested to handle the above challenges. Thus, the significant contribution of the study involves,

- The established IDS system should be adequate for bulk data processing without intrusion into the cloud environment. To attain this purpose, the Random Forest Classifier is used for classification and to increase the accuracy of classification rate efficient feature selection method used, namely Optimized Sine Swarm (OSS) algorithm. It further returns the best solution among global optimum.
- To analyze the classification performances, the UNSW-BoT Dataset is utilized.
- The performance has been related to other existing models to measure the proposed IDS system's effectiveness in detecting malicious attacks in the cloud environment.

2. INTERCONNECTED WORKS

In [4], the developed model has been inserted into the cloud environment, which permits capturing the incoming network traffic to the physical layers' router of the edge network. Every cloud router captures pre-processing time based on the sliding window technique for network traffic. Using the Naive Bayes classifier further passes through the malicious detection framework. To MapReduce and Hadoop, commodity server nodes are available in every malicious detection framework in usage when an increase in network congestion is observed. Ensemble learning classifiers can be utilized for the random forest for multi-class classification performance for every attack type identification.

Further in [5], the IDS- intrusion detection system for MLP- Multilayer perceptron network, fuzzy clustering, and ABC- artificial bee colony have been designed. MLP detected the abnormal and normal network traffic packets in which the ABC algorithm performed the MLP training through linkage biases and weights optimization. For the verification NSL-KDD dataset and cloudSim simulator has used. The performance metrics considered were RMSE, kappa statistic, and mean absolute error. The better performance resulted by compared with existing methods.

Moreover, in [6], SVM- support vector machine and FCM-fuzzy c means clustering have been utilized for accuracy enhancement for malicious detection in the cloud environment. NSLKDD dataset used to verify the hybrid context shows that the proposed model detects malicious activities with lesser false alarm rates and higher accuracy than state-of-the-art algorithms. In [7], the CS-PSO algorithm developed for IDS is used in user activities prevention over the cloud environment. Informing the whole technique, CS-PSO performed a significant role using the NSL-KDD dataset. From a higher dimensional dataset, feature selection is considered a better technique based on memory storage, and training time shows the efficiency improvement in IDS.

Similarly, in [8], an effective malicious detection model was developed based on the cloud environment. For intrusion detection and profile training, SVM has been used. Better feature set technique-based optimized NSL-KDD dataset developed for information gain ratio obtained lesser false alarm rate and 96.24% accuracy. SVM approach shows major benefits for IDS evolution in inspiring, challenging environments. Consequently, in [9], IDS in cloud platforms has focused because of its distributed nature and extensive usage, considered constant unknown attacks and new targets. IDS is in charge of detecting and monitoring suspicious activities in any network. Most of the conventional IDSs shows risks of new kind of attacks. Maintaining the balance between the lesser false positive rate and higher accuracy shows incapability. Cloud IDS framework with deep reinforcement learning has been developed, which handles the problematic situations, and it further performs new and challenging attacks detection and classification.

The dataset used was the UNSW-NB15 dataset shows lesser FPR and higher accuracy compared with the existing algorithm. Specifically, the DDoS attack has focused in [10] on the proposed self-adaptive evolutionary extreme learning machine model. It has been improved because the combination of two features better crossover operator has been used. Hidden layer neurons found themselves. The properties improve categorization and learning. UNSW-NB15, ISCXIDS 2012, CICIDS 2017, and NSL-KDD datasets have been used, and the accuracy resulted as 89.1%, 98.9%, 99.9% and 86.8 correspondingly. The developed attack detection model shows better results compared with existing techniques.

Snort has used signature-based detection, various feature selection, and machine learning algorithms focused on anomaly detection. Intrusion detection accuracy improved in this model while minimizing the false alert. The developed framework's feasibility and performance have been investigated based on attacks performance using the UNSW-NB15 dataset [11]. In [12], HLDNS- hypervisor level distributed network security has developed deployed on cloud computing's every processing server. The BBA-binary bat algorithm with novel fitness functions from cloud network traffic. The derived features were applied to a random forest classifier for intrusion detection and generating alerts. Distributed attacks are identified through intrusion, generating a new attack signature. The developed security context has been evaluated on a testbed of cloud network using CICIDS-2017 and UNSW-NB15 datasets. For cloud network security fulfillment, a comparative analysis was performed.

In [13], a complex network environment to deal with malicious data and cyber-attacks, the machine learning technique has been proposed to be utilized in the intrusion detection model. For the model training random forest method, further accuracy has been verified using the test set. 0.94 F1 score has resulted while performing intrusion data detection. Cloud computing is the technology used to securely store and maintain users' information at a low cost. Nowadays, a new cloud computing model is seeking attention known as mobile cloud computing (MCC). They are used on mobile devices for accessing their cloud environment. The estimation of IDS networks that use the computational intelligence (CI) model in MCC has been represented [14]. IDS with swarm intelligence was deployed in the needed networks to handle the attacks. Whale Pearson hybrid feature selection wrapper developed to help with IDS. This algorithm is a better version of the binary Whales Optimization Algorithm (WOA). The outcome was having an accuracy of 80%, which is an 8% increase related to the KNN algorithm [15]. The number of cyberattacks in the cloud environment is expanding in parallel with the internet and related technology expansion. This work reviews the collaborative measures taken against the alerts from outsiders. The ability to detect anomalies by IDS in the early stage is more crucial than other criteria.

This work gives an outlook on the various detection system engaged in a cloud system to avoid damages suggested by National Intrusion Detecting Systems (NIDS) [16]. Security for the data in a cloud environment has expanded for cloud environment. This virtual system is still facing some challenges regarding this security factor. A recurrent convolutional neural network (RCNN) has been suggested to overlook this issue. This RCNN is used to find if text data is an intrusion or not. Elliptical curve cryptography (ECC) is engaged in increasing the security performance of the system. The output of the above approach is highly securing the data and does not get affected by attacks and warnings [17]. Among the security issues for cloud environments, IDS is the most frequently searched term on the internet. This issue is creating a sensation in cloud servers. Decision trees to perform binary and multiclass classification. The result of this work has created robustness of dataset features similar to Azure machine learning. The statistical result for the work involving other classifiers is also represented [18].

3. PROPOSED METHODOLOGY

In this section, the newly proposed IDS model in the Cloud environment, namely OSS-based Random Forest classifier, has been elaborated, and the overall flow is shown in Figure 1. Initially, the UNSW-NB15 dataset is loaded, and the pre-processing is performed. Further to enhance the detection accuracy, the feature selection process is performed using the Optimized Sine Swarm (OSS) algorithm, selecting only the significant features. Finally, the classification of the anomalies is performed using the random forest classification. In the prediction phase, it detects all attack types. Finally, the proposed OSS-based Random Forest model evaluated different performance metrics and compared them with various existing models to prove its efficiency.

3.1. IDS Model in Cloud Environment

Initially, the packet sniffer evaluates the streams of data packets that flow between the network and monitors the network traffic. Further PCAP files are presented to create files comprised of packet data of the network used to evaluate the characteristic of the network. If the packets are not malicious, they are stored in logs or passed the IDS model and identify the attacks using the proposed model. Further detection engine gets an alert and new rule generated for intrusion log generator and sends to cloud administrators, and further, the packets are dropped. The overall flow in a cloud environment is shown in Figure 2.

Cloud invasions can be detected using machine learning methods. Machine learning is an intrusion detection method for tenants' behavior in a cloud platform. A cloud administrator should be alerted to any unusual patterns. It aids in learning each tenant network's profile information in a dynamic setting. The significant benefit is that the retraining classifiers can identify the regular behavior changes while the new kind of application is installed in the cloud platform.

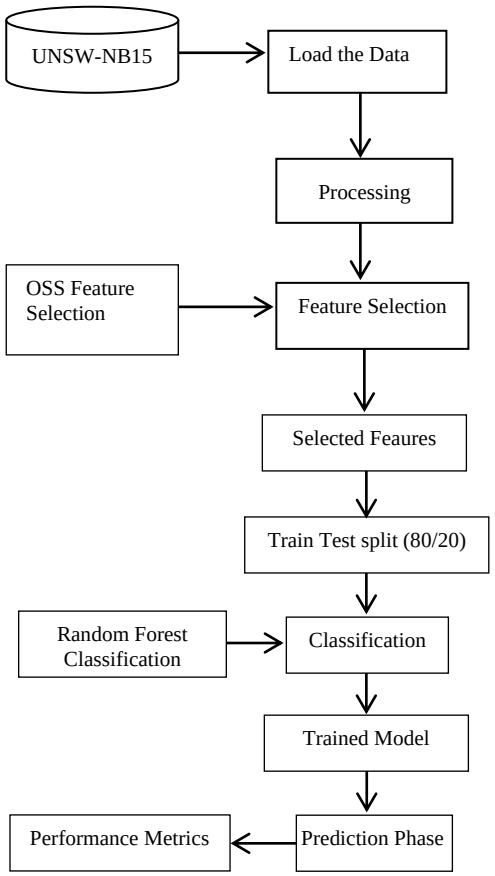

Figure 1. Proposed workflow of IDS model

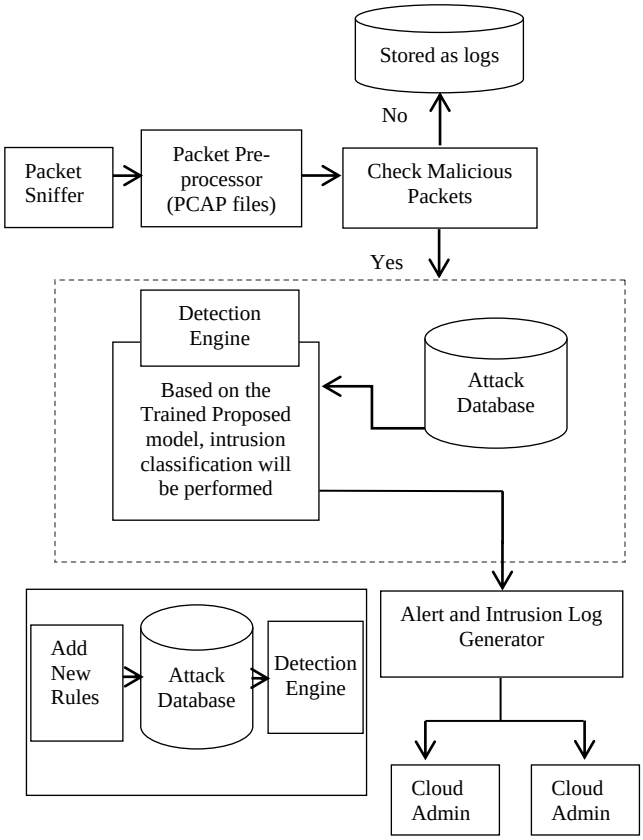

Figure 2. IDS model in cloud environment

However, specific methods exhibited higher processing time due to the classifier's false alarms and expected behavior. A machine learning algorithm is easy to modify in a dynamic cloud environment for identifying malicious/intrusions. Appropriate machine learning algorithms selection needs available datasets and expertise knowledge for training. In Figure 3, a network node with a trained IDS is depicted. A dataset is used to assess the classification algorithm's performance during the training and testing phases of the IDS system. The acquired dataset is utilized to learn traffic patterns' normal behavior in the training phase. During testing, the trained classifier sorts the test cases.

Malicious activities in the system are associated with learned behavior deviations. The cloud administrator is responsible for traffic validation. Traffic behaviors are monitored in the network node, packets form into test classes, moving to the stored trained classifier. If there is no signal, then malicious activities are not present. If the alert signal comes, the cloud initiates malicious application deletes or virtual machine generating application is isolated, and the classifier is retrained for new behavior.

3.2. Feature Selection Using Optimized Sine Swarm Algorithm

The basic principle of Particle swarm optimization is derived from the social life of birds and fishes who live in groups. Every particle is the solution to the selected optimization problem and passed using the position and velocity vectors in design space to identify a new solution. Initially, the velocity and position vectors of all particles are randomly generated, and every particle passes in the design space using a better position. in earlier implementing stages, the velocity and position vectors of all particles are randomly generated as $A_i = \{A_{i1}, A_{i2}, \ldots, A_{id}\}$ and $V_i = \{V_{i1}, V_{i2}, \ldots, V_{id}\}$.

In design space, every particle moves using the best position experience from that particle $\vec{A}_{\text{best}_i}$ and better solution attained by all particles $\vec{A}_{g\text{best}_i}$. After the position and velocity of each particle [20],

$$\vec{V_i}(q+1) = wV_i(q) + s_1k_1\left(\vec{A}_{p\text{best}_i} - \vec{A_i}(q)\right) + \\ + s_2k_2\left(\vec{A}_{g\text{best}_i} - \vec{A_i}(q)\right) \quad (1)$$

$$\vec{A_i}(q+1) = \vec{A_i}(q) + \vec{V_i}(q+1) \quad (2)$$

From Equations (1) and (2), $\vec{V_i}(q+1)$ and $\vec{A_i}(q+1)$ are considered as velocity and position vector of i particle at $q+1$ iterations. Personal and learning factors are s_1 and s_2 in interval k_1 and k_2 are the random numbers. The adaptive coefficient is expressed as

$$w(q) = w_f + \left(\frac{1+\cos\left(\frac{\pi q}{q_{\max}}\right)}{2}\right)^d \left(w_i - w_f\right) \quad (3)$$

From Equation (3), initial and final weight factor values are w_i and w_f. The maximum no. of repetition is $q_{\max}$ The current iteration number is q, and the constant number is d, indicating the decreasing weight factor intensity. If s_1 and s_2 treated as decreasing and incremental monotonic functions during optimization process, it resulted to exploration capability at initial stage and at final stage exploitation capability is resulted, defined as,

$$s_1(q) = s_{1\min} + \left(\frac{q_{\max} - q}{q_{\max}}\right)\left(s_{1\max} - s_{2\min}\right) \quad (4)$$

$$s_2(q) = s_{2\min} + \left(\frac{q_{\max} - q}{q_{\max}}\right)\left(s_{2\min} - s_{2\max}\right) \quad (5)$$

From Equations (4) and (5), $s_{1\min}$ and $s_{2\min}$ are the personal and social minimum values and maximum values are $s_{1\max}$ and $s_{2\max}$ correspondingly.

Sine Cosine algorithms - SCA generates multiple random solutions and leads to optimal equations through sine and cosine functions. The distance among the solutions and the best member of population affects the movement and sine cosine equations capable of generating balance among exploration and exploitation by implementing far fewer operators than other algorithms. This method applies following relationship for exploitation and exploration solution update as,

$$\vec{A_i}(q+1) = \vec{A_i}(q) + k_1 \sin(k_2)\left|k_3P^t - \vec{A_i}(q)\right| \quad (6)$$

$$\vec{A_i}(q+1) = \vec{A_i}(q) + k_1 \cos(k_2)\left|k_3P^t - \vec{A_i}(q)\right| \quad (7)$$

From Equations (6) and (7), $\vec{A_i}(q)$ and $\vec{A_i}(q+1)$ are the present solutions positions in and $q+1$ iterations respectively. The random numbers are k_1, k_2 and k_3 and the best member position of the population is P^q, || is the absolute value or iteration q destination point. By combining these equations, the solution updates are then

$$\vec{A_i}(q+1) = \begin{cases} \vec{A_i}(q) + k_1 \sin(k_2)\left|k_3P^q - \vec{A_i}(q)\right|, & k_4 < 0.5 \\ \vec{A_i}(q) + k_1 \cos(k_2)\left|k_3P^q - \vec{A_i}(q)\right|, & k_4 \geq 0.5 \end{cases} \quad (8)$$

From Equation (8), random value is $k_4 \in [01]$ and performs as switching factor among the equations. Four effective parameters are k_1, k_2, k_3 and k_4 for new solution position determination. Next position specified from k_1 parameter.

$$k_1 = a\left(1 - \frac{q}{q_{\max}}\right) \quad (9)$$

From Equation (9), q and $q_{\max}$ are considered as current and maximum iteration correspondingly and

constant number is *a*. Further levy flight in various optimization algorithms used for generating random step size and it is considered as random process with the non-Gaussian distribution expressed as, $L(b) \sim |b|^{-1-\beta}$. The Levy flight distribution mathematical formula is defined as,

$$L(b,\gamma,\mu)=\begin{cases}\sqrt{\frac{\gamma}{2\pi}}\exp\left(-\frac{\gamma}{2(b-\mu)}\right)\frac{1}{(b-\mu)^{\frac{3}{2}}}, & 0<\mu<¥\\ 0 & \text{otherwise}\end{cases} \quad (10)$$

For this distribution, the transmission parameter and samples are μ and b. Levy flight distribution scale control by γ parameter and in Fourier transform it is defined as,

$$F(k)=\exp\left(-\alpha|k|^{\beta}\right),\ \beta\in(0,2] \quad (11)$$

From Equation (11), scale parameter is α. The analytical form is measured for certain exceptional cases β. Based on Mantegna algorithm, step length *b* can be determined,

$$b=u/|v|^{\frac{1}{\beta}} \quad (12)$$

From Equation (12), Gaussian distribution determined from *u* and *v* parameters and are attained as,

$$u\sim N\left(0,\sigma_u^2\right),\ v\sim N\left(0,\sigma_v^2\right) \quad (13)$$

$$\sigma_u=\left\{\frac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left[\frac{(1+\beta)}{2}\right]\beta\times 2^{\frac{(\beta-1)}{2}}}\right\}^{\frac{1}{\beta}},\ \sigma_v=1 \quad (14)$$

The step size is measured from,

$$Stepsize=scale\times b \quad (15)$$

Basically, in Particle Swarm Optimization (PSO) shows superiority in higher computational speed, standard implementation and small no. of parameters, however, it exhibited two significant drawbacks as falling to local minima and premature convergence. The researchers have introduced different PSO algorithm versions by merging position and velocity b updating the Levy flight based PSO equations. It resulted in highly effective search in search space with higher jumps and different solutions are generated.

For generating more different solutions through the design space and exploration capability increased through sine cosine algorithm and further Levy flight distribution included to sine cosine algorithm as Equation (16), which $Levy_{walk}\left(\vec{A}_i(q)\right)$ expression added as Equation (17).

$$\vec{A}_i(q+1)=\begin{cases}Levy_{walk}\vec{A}_i(q)+k_1\sin(k_2)\left|k_3\vec{A}_{gbest}-\vec{A}_i(q)\right|, & k_4<0.5\\ Levy_{walk}\vec{A}_i(q)+k_1\cos(k_2)\left|k_3\vec{A}_{gbest}-\vec{A}_i(q)\right|, & k_4\geq 0.5\end{cases} \quad (16)$$

$$Levy_{walk}\left(\vec{A}_i(q)\right)=\vec{A}_i(q)+\overrightarrow{step}\overrightarrow{nrandom}(size\left(\vec{A}_i(q)\right)) \quad (17)$$

$$\overrightarrow{step}=stepsize\oplus\vec{A}_i(q) \quad (18)$$

The below algorithm 1 shows the OSS algorithm. According to this pseudo code, population size is n_{pop}, max iteration is Max_{it}, in search space, $A_{\min}$ and $A_{\max}$ are lower and upper bound and problem dimension is dim. The velocity and position of particle are randomly generated initially. Further the objective function is measured, and for every particle A_{pbest} is measured and for whole population A_{gbest} is measured. Hence, using Equations (3) to (5), the parameters value s_1, s_2 and *w* have been measured in every iteration. before the ever particle velocity vector updated the particle limit value is validated. If it is smaller compared with the predetermined value, using Equation (1) and (2), the particle's position and velocity are updated. Or else using Equation (16) and (17), the next position of the particle has identified. To that particle, random value is allocated. Using Equation (16), the position of particle measured, if rand(0.5). The new attained position is in range of ($A_{\min}$, $A_{\max}$). If the range exceeds then

$$A=\min(A,A_{\max})$$

$$A=\max(A,A_{\min})$$

Algorithm 1. Optimized sine swarm (OSS)

Initialize parameters (npop, MaxIt, $A_{max,dim}$, A_{min}) Trial (for every particle, maintain limit value) = 0
Initialize the random positions particles (A_i) and (V_i) random velocity within the range of initialization
  Assess the fitness
  Set A_i as A_{pbest}
  Set best fitness to particle as A_{gbest}
  For q = 1, Maxit
  Measure $s_1(q), s_2(q)$ and w(q) for i = 1: npo
if tri(i) < limit
    Using Equation (1), Velocity v_i of particle update
    Using Equation (2), position A_i of particle update
   Else
if rand() < 0.5
     Using Equation (16), update A_i
    else
     Using Equation (17), update A_i
    end if
   end if
   To the boundary value the position value is set, while the value is passed out of search space boundary
   For new particle A_i, Assess the fitness value,
if $f(A_i(q)) < f(A_{pbest})$
     tri(i) = 0
   $A_{pbest} = A_i$
 else
   Tri(i) = tri(i) + 1
end if
     if $(A_{pbest}) < f(A_{gbest})$
     $A_{gbest} = A_{pbest}$
    end if
  end for
 end for

3.3. Classification using Random Forest Classifier

Random forest algorithm is introduced by Breiman, to idea of forest and an election. Every tree in forest identifies as voter. The set of percentages and votes is the standard for making the end decision. Bagging method is used for every tree by RF method for random training dataset generation. By RF, splitting features are selected semi-randomly. A random subset of particular ratio is provided through the possible feature space splitting.

Algorithm 2. RF (Random Forest Detection)

```
To create n classifiers
for i=1 to n do
Rrandom sample of the training data R t is replaced by R_i
Put R in a root node,X_i
Invoke Build_Tree(X_i)
end for
Build_Tree(X):
if only one class is present in T, then
return
else
Pick at random x% of T's possible splitting features
A high-information gain feature F should be selected for splitting.
 Create b child nodes of X,X_1…….X_b, where, B has b  possible values(B1…… Bf)
              for b=1 to f do
                Set the contents of X_i to R_i , where, R_i  is all instances in T that Match B_i
                Call BuildTree(X_i)
              end for
end if
```

4. FINDINGS AND ANALYSIS

The suggested OSS feature selection-based RF classifier method used to detect the intrusion on UNSW-NB15 dataset. The UNSW-NB15 dataset raw network packets have been generated by UNSW Canberra cyber range lab for synthetic contemporary behavior attack and generating activities of real modern normal. It was created utilizing the IXIA Perfect Storm tool at ACCS-Australian Centre for Cyber Security's cyber range lab to create realistic modern regular activity hybridization and from network traffic, simulated current attack behaviors focusing on the UNSW-NB15 dataset [19]. A TCP dump tool utilized for record network traffic of 100GB. Bro-IDS and Argus tool were utilized and for extracting features 12 models have been developed using [20]. Due to the fact that they are classic single classifiers with little training data and a narrow hypothesis space, neural networks, support vector machines, and decision trees (DT) are simple to acquire the local optimal value.

Table 1. Sample data distribution

Category	Training Data	Testing Data
Normal	56001	37001
Analysis	2001	678
Backdoor	1,747	584
DoS	12,265	4090
Exploits	33,394	11,131
Fuzzers	18,185	6063
Generic	40,001	18.872
Reconnaissance	10,490	3,497
Shellcode	1,134	379
Worms	131	45
Total Records	175,351	85,342

Table 1 explains the generation of training and testing set from the selected UNSW-NB15 dataset. A part of dataset observed and records have divided with 60:40 percent ratio approximately as training and testing set. No redundant records between the training and testing set are recorded in attaining the IDS evaluation authenticity.

4.1. Performance Analysis

From Table 2, the performance of proposed IDS model is depicted to F_1-score, precision, recall, and accuracy. Using the OSS feature selection algorithm, accuracy, precision, recall and F_1-score were obtained as 98.15, 98.24, 97.36 and 97.19 respectively whereas without feature selection algorithm, the accuracy, precision, recall, and F_1-score obtained as 95.68, 95.87, 94.36 and 94.83 respectively. Hence the proposed OSS feature selection algorithm performed well in enhancing the accuracy values of IDS model.

Table 2. Performance analysis of proposed IDS model (with/without feature selection)

Metrics	Feature Selection	Without Feature Selection
Accuracy	98.15	95.68
Precision	98.24	95.87
Recall	97.36	94.36
F1-Score	97.19	94.83

Table 3. Total features selected

Total Features	Selected Features	Total Selected Features
43	1, 3, 5, 6, 8, 9, 12, 15, 19, 25, 29, 31, 35, 37	14

From Table 3, it shows the total features identified as 43, from that the selected optimal features are 1, 3, 5, 6, 8, 9, 12, 15, 19, 25, 29, 31, 35, 37 and thus the total no. of selected features is 14 used for classification.

4.2. Performance Comparison

Predictions are obtained from all the existing algorithms, and the predictions are multiplied with the corresponding detection levels of each of the models The Figure 3 depicts the Performance Comparison of proposed classification of malicious activities in cloud environment with the existing algorithms DT [17], ANN [18], LR [19], SVM [20] and KNN [21]. It is defined as the prediction accuracy of the correct detections of the Intrusion Detection System. It is the ratio of the total number of correct detections to the total number of detections. High prediction accuracy refers to the high identification rate and precise detection mechanism. The assessment shows the suggested method accuracy outperformed the accuracy values of existing studies as shown in Figure 3, in detecting malicious activities in cloud environment using OSS-based RF classifier.

The Figure 4 provides the accuracy prediction of the nefarious activity in the cloud with the existing methods CNN CE [23], FL-NIDS [24], and CNN SMOTE [22]. Both precision and recall represent malicious IDS detection levels of the model in identifying intrusions. The proposed algorithm also indicates the reduce false

positive rate while the existing algorithm has high false positive rate. The results show the proposed method accuracy outperformed the accuracy values of existing models as shown in Figure 4, in detecting malicious activities in cloud environment using OSS-based RF classifier.

Figure 3. Performance metrics of various existing methods and the proposed method

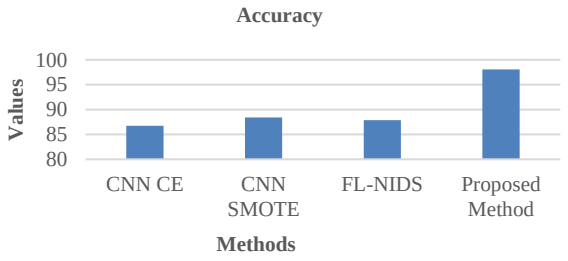

Figure 4. Comparison of proposed OSS based RF and existing methods

The Figure 5 shows the comparative analysis of proposed classification of malicious activities in cloud environment with the existing methods ELM [25], DNN [19] and MSCNN [26]. The proposed algorithm is verified using the UNSW-NB15 dataset. From Figure 5, it is observed that the existing algorithms have a poor detection rate and lack the efficiency of high accuracy detection. The results show the proposed method accuracy, precision, recall and F-measure values are outperformed the values of existing studies as shown in Figure 5, in detecting the malicious activities in cloud environment by using OSS based RF classifier.

5. CONCLUSION

Ensemble learning can increase generalization performance by increasing the learning effect of the algorithm on unbalanced data. This is accomplished by mixing many classifiers in a single training set. A classical ensemble learning model like RF can use bootstrap sampling, which is a random choose sample combined with randomly selected base learner features, to successfully balance data set errors, which is advantageous in preventing overfitting when dealing with imbalanced classification.

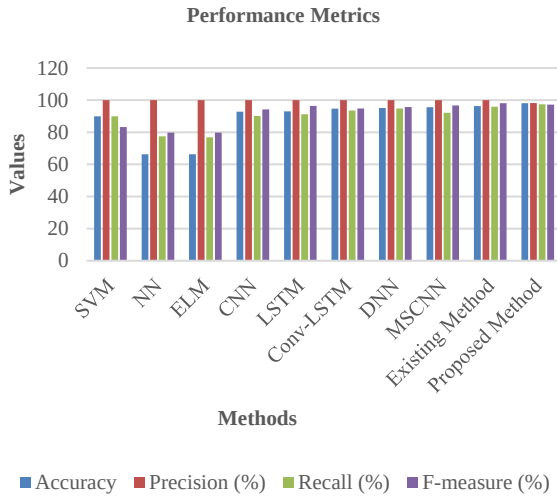

Figure 5. Comparison of precision, recall and F-measure of OSS Based RF classifier with existing methods

This research uses the Optimized Sine Swarm for feature selection to select the UNSW-NB15 optimum characteristics of the dataset. The appropriate classification is made using the Random Forest classification. With a precision rate of 98.15 percent, the proposed OSS-RF system achieved the greatest accuracy rate for UNSW-NB15. When it comes to classifying and detecting various attacks, the suggested method beats the existing methodologies, as demonstrated by the dataset and success rate as a result, the system is effectively restored in the shortest amount of time.

REFERENCES

[1] W. Wang, X. Du, N. Wang, "Building a Cloud IDS Using an Efficient Feature Selection Method And SVM", IEEE Access, Vol. 7, pp. 1345-1354, 2018.

[2] Z. Chiba, N. Abghour, K. Moussaid, M. Rida, "Intelligent Approach to Build a Deep Neural Network based IDS for Cloud Environment Using Combination of Machine Learning Algorithms", Computers and Security, Vol. 86, pp. 291-317, 2019.

[3] C. Zouhair, N. Abghour, K. Moussaid, A. El Omri, M. Rida, "A Review of Intrusion Detection Systems in Cloud Computing", Security and Privacy in Smart Sensor Networks, pp. 253-283, 2018.

[4] M. Idhammad, K. Afdel, M. Belouch, "Distributed Intrusion Detection System for Cloud Environments Based on Data Mining Techniques", Procedia Computer Science, Vol. 127, pp. 35-41, 2018.

[5] B. Hajimirzaei, N.J. Navimipour, "Intrusion Detection for Cloud Computing Using Neural Networks and Artificial Bee Colony Optimization Algorithm", ICT Express, Vol. 5, No. 1, pp. 56-59, 2019.

[6] A.N. Jaber, S.U. Rehman, "FCM-SVM Based Intrusion Detection System for Cloud Computing Environment", Cluster Computing, Vol. 23, No. 4, pp. 3221-3231, 2020.

[7] P. Ghosh, A. Karmakar, J. Sharma, S. Phadikar, "CS-PSO Based Intrusion Detection System in Cloud

Environment", Emerging Technologies in Data Mining and Information Security: Springer, pp. 261-269, 2019.
[8] S. Krishnaveni, P. Vigneshwar, S. Kishore, B. Jothi, S. Sivamohan, "Anomaly-Based Intrusion Detection System Using Support Vector Machine", Artificial Intelligence and Evolutionary Computations in Engineering Systems, Springer, pp. 723-731, 2020.
[9] K. Sethi, R. Kumar, N. Prajapati, P. Bera, "Deep Reinforcement Learning Based Intrusion Detection System for Cloud Infrastructure", International Conference on Communication Systems and Networks (COMSNETS), pp. 1-6, 2020.
[10] D. Kadam, R. Patil, C. Modi, "An Enhanced Approach for Intrusion Detection in Virtual Network of Cloud Computing", Tenth International Conference on Advanced Computing (ICoAC), pp. 80-87, 2018.
[11] R. Patil, H. Dudeja, C. Modi, "Designing an Efficient Security Framework for Detecting Intrusions in Virtual Network of Cloud Computing", Computers and Security, Vol. 85, pp. 402-422, 2019.
[12] X. Guo, H. Huang, X. Meng, "Random Forest for Intrusion Detection of Cloud Manufacturing Platform", The 26th International Conference on Automation and Computing (ICAC), pp. 1-6, 2021.
[13] V. Ravindranath, S. Ramasamy, R. Somula, K.S. Sahoo, A.H. Gandomi, "Swarm Intelligence Based Feature Selection for Intrusion and Detection System in Cloud Infrastructure", Congress on Evolutionary Computation (CEC), pp. 1-6, 2020.
[14] O. Alkadi, N. Moustafa, B. Turnbull, "A Review of Intrusion Detection and Blockchain Applications in the Cloud: Approaches, Challenges and Solutions", IEEE Access, Vol. 8, pp. 104893-104917, 2020.
[15] V. Prabhakaran, A. Kulandasamy, "Integration of Recurrent Convolutional Neural Network and Optimal Encryption Scheme for Intrusion Detection with Secure data Storage in the Cloud", Computational Intelligence, Vol. 37, No. 1, pp. 344-370, 2021.
[16] S. Rajagopal, P.P. Kundapur, K. Hareesha, "Towards Effective Network Intrusion Detection: From Concept to Creation on Azure Cloud", IEEE Access, Vol. 9, pp. 19723-19742, 2021.
[17] L. Catherine, R. Pathak, V. Vaidehi, "Efficient Host Based Intrusion Detection System Using Partial Decision Tree and Correlation Feature Selection Algorithm", International Conference on Recent Trends in Information Technology, pp. 1-6, 2017.
[18] N. Pandeeswari, G. Kumar, "Anomaly detection System in Cloud Environment Using Fuzzy Clustering Based ANN", Mobile Networks and Applications, Vol. 21, No. 3, pp. 494-505, 2016.
[19] S. Naseer, Y. Saleem, S. Khalid, M.K. Bashir, J. Han, M.M. Iqbal, K. Han, "Enhanced Network Anomaly Detection Based on Deep Neural Networks", IEEE Access., Vol. 14, No. 8, August 2015.
[20] S. Mukkamala, G. Janoski, A. Sung, "Intrusion Detection Using Neural Networks and Support Vector Machines", International Joint Conference on Neural Networks, pp. 1702-1707, August 2002.
[21] N. Moustafa, J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)", Military Communications and Information Systems Conference (MilCIS), pp. 1-6, 2015.
[22] N. Moustafa, J. Slay, "The Evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 Data Set and the Comparison with the KDD99 Data Set", Information Security Journal: A Global Perspective, Vol. 25, No. 1-3, pp. 18-31, 2016.
[23] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, F. Ahmad, "Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches", Transactions on Emerging Telecommunications Technologies, Vol. 32, p. e4150, 2021.
[24] Y. Liu, S. Liu, X. Zhao, "Intrusion Detection Algorithm Based on Convolutional Neural Network", DEStech Transactions on Engineering and Technology Research, 2017.
[25] G.B. Huang, Q.Y. Zhu, C.K. Siew, "Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks", International Joint Conference on Neural Networks, No. 4, pp. 985-990, 2004.
[26] Z. Gong, P. Zhong, Y. Yu, W. Hu, S. Li, "A CNN With Multiscale Convolution and Diversified Metric for Hyperspectral Image Classification", Transactions on Geoscience and Remote Sensing, Vol. 57, No. 6, pp. 3599-3618, June 2019.
[27] D. Zheng, C. Qin, P. Liu, "Adaptive Particle Swarm Optimization Algorithm EnsembleModel Applied to Classification of Unbalanced Data", Scientific Programming, Vol. 2021, pp. 1-13, 2021.
[28] S.M. Kasongo, Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset", Journal of Big Data, Vol. 7, No. 1, pp. 1-20, 2020.
[29] M. Mulyanto, M. Faisal, S.W. Prakosa, J.S. Leu, "Effectiveness of Focal Loss for Minority Classification in Network Intrusion Detection Systems", Symmetry, Vol. 13, No. 1, p. 4, 2020.
[30] P.R. Kanna, P. Santhi, "Unified Deep Learning Approach for Efficient Intrusion Detection System Using Integrated Spatial-Temporal Features", Knowledge-Based Systems, Vol. 226, p. 107132, 2021.
[31] J.N. Victor, "A Profit Driven Stacking Model for Effective Churn Prediction", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 50, Vol. 14, No. 1, pp. 89-94, March 2022.
[32] R. Samsami, "Comparison between Genetic Algorithm Particle swarm optimization and Ant Colony Optimization Techniques for NO_X Emission Forecasting in Iran", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 16, Vol. 5, No. 3, pp. 80-85, September 2013.

BIOGRAPHIES

Jesurathinam Vimalrosy was born in 1980 at Madurai district, Tamil Nadu, India. She has completed her Master degree in Philosophy Computer Science, and is currently pursuing her Ph.D. in Computer Science in the Field of Cloud Computing. She currently serves as a Head and Assistant Professor at Department of Computer Science, Soka Ikeda College of Arts and Science for Women, Chennai, Tamil Nadu, India.

Swakkin Brittorameshkumar was born in Madurai district, Tamil Nadu, India. He has a obtained his Master degree from Madurai Kamaraj University, India and his doctorate from Bharathidasan University, India. Now he is an Assistant Professor of Computer Science at St. Joseph's College (Autonomous), Tiruchirappalli, India. His research interests include software architecture, wireless and mobile technologies, information security and Web Services. He has published many journal articles and book chapters on the topics of mobile payment and data structure and algorithms. He was awarded as the best researcher for the year 2008 in Bishop Heber College, Tiruchirappalli, India. He has completed a minor research project.