Intrusion Detection System using Long Short Term Memory Classification, Artificial Raindrop Algorithm and Harmony Search Algorithm

Meghana G Raj[1], Dr. Santosh Kumar Pani[2]
Assistant Professor[1], Professor[2]
Department School of Computer Engineering, Kalinga Institute of Industrial Technology (Deemed to be University),
Bhubaneswar, India

***Abstract*—Nowadays, various technological advancements in Intrusion Detection Systems (IDS) detects the malicious attacks and reinstate network security in the cloud platform. Cloud based IDS designed with hybrid elements combining Machine Learning and Computational Intelligence algorithms have been shown to perform better on parameters, such as Detection Rate, Accuracy, and the False Positive Rate. Machine Learning algorithms provide effective techniques for classification and prediction of network attacks, by analyzing existing IDS datasets. The main challenge is selection of appropriate data dimensions to be used for detection of attacks, out of the high number of data dimensions available. For the selected data dimensions, Computational Intelligence Algorithms provide effective techniques for hyper-parameter tuning, by optimizing on reiterative basis. The main challenge is selection of appropriate algorithm which offers optimal performance results. In this research, Hybrid Meta-heuristic approach, which combines a Long Short Term Memory (LSTM) classification model in dimension selection, with the application of Artificial Raindrop Algorithm- Harmony Search Algorithm (ARA-HSA) for hyper-parameter tuning, in order to achieve a high performance IDS in cloud environment. The performance validation of the hybrid LSTM-ARA-HSA algorithm has been carried out using a benchmark IDS data set and the comparative results for this algorithm along with other recent hybrid approaches has been presented.**

Keywords*—Artificial raindrop algorithm; cloud computing; harmony search algorithm; hybrid meta-heuristic algorithms; hyper-parameter tuning; intrusion detection system; long short term memory classification model

I. INTRODUCTION

The Cloud Computing (CC) platform, which is on-demand network access for several pattern of computing asset, such as servers, applications, networks, and services. Security is the major difficulty for the organizations to accept the cloud enabled solutions. Due to cloud infrastructure (open and fully distributed), which makes it vulnerable to attacks and threats. Thus, it creates incentives for intruders for initiating attack focused devices to permit the data stored in cloud. The threats are confidentiality, availability, and integrity of cloud resources and services. To solve the security challenges, the IDS should be integrated within the cloud environment [1].

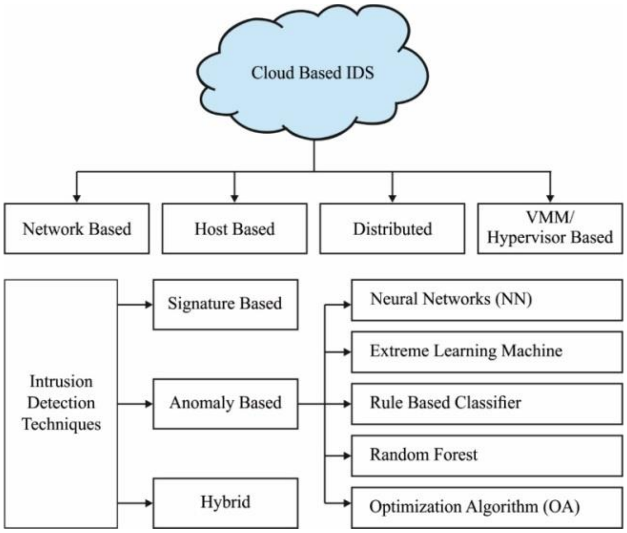

Fig. 1. Types of cloud based IDS

There are several kinds of attacks that could probably arise in a network, such as User to Root (U2R), Remote to Local (R2L), and the Denial of Service (DoS) attacks. In U2R attacks, a local user who is the attacker gets unauthorized access to the routing privilege and access control list [2]. In R2L attacks, the attacker is not a local user but remotely forwarded a set of packets to server or computer over the network, to attempt unauthorized access [3]. In DoS attacks, the attacker attempts to stops the standard service in the network by over-whelming the network with high number of request data packets [4].

Machine learning (ML) algorithms offer effective techniques to improve a efficiency and effectiveness of IDS, due to the capability to identify potential attacks through sophisticated classification of network states described by data dimensions, based on comparison with known network states during previous network attacks [5]. Several ML techniques [6], such as Extreme Learning Machine, Random Forest Classifiers, Multi-layer Perceptron Network etc. have been designed and their performance in detecting attacks evaluated [7]. Fig. 1 provides an overview of types of IDS, based on the different ML algorithms adopted.

The main challenges in design of IDS are:

- Selection of appropriate number data dimensions to be used for detection of attacks, out of the high number of data dimensions available
- Hyer-parameter tuning for the selected data dimensions
- Improving performance in classification

The challenge of feature selection is sought to be overcome by adoption of appropriate ML algorithms. The challenge of hyper-parameter tuning is sought to be overcome by adoption of appropriate Computational Intelligence algorithms. The challenge of improved performance is sought to be overcome by adoption of appropriate ML algorithms. The proposed solution therefore, consists of a hybrid approach, in which both ML and Computational Intelligence algorithms incorporated in order to deliver improved performance compared to existing approaches in the literature.

This paper presents a Hybrid Meta-heuristic approach which uses ML techniques- ExtraTrees Feature Selection Method and LSTM classification model- along with Computational Intelligence Algorithms- Artificial Raindrop Algorithm and Harmony Search Algorithm. For feature selection, ExtraTrees (ET) method is used to choose a proper set of data dimensions to be employed for classification [8][9] and prediction [10] of network attacks, out of maximal number of data dimensions available in the standard IDS data set of known attacks from the past. Artificial Raindrop Algorithm (ARA) and Harmony Search Algorithm (HSA) are used for hyper-parameter tuning (i.e. optimizing the parameter values to be used) for the data dimensions selected. Finally, classification is achieved using the LSTM classification model.

To estimate the resulting performance of IDS which uses hybrid combination of LSTM-ARA-HSA approach, the paper presents the results of simulations using the benchmark data set.

In summary, the paper's contributions can be summarized as follows.

- Use an ML technique- ExtraTrees Feature Selection method for selecting optimal set of data dimensions for classification and prediction of network attacks, out of the high number of data dimensions available for analysis.
    - Perform min-max based data normalization to achieve standardization of the different scales for the available data dimensions.
    - Use of ET Feature Selection method for the optimum selection of data dimensions, to enhance the efficiency of IDS.
- Adopt combination of ARA-HSA algorithms for hyper-parameter optimization of the data dimensions selected.
- Perform final classification using the LSTM classification model.
- Design a Hybrid Meta-heuristic approach for IDS using a combination of LSTM-ARA-HSA, as described above.
- Test the performance of Hybrid Meta-heuristic LSTM-ARA-HSA, using simulations on the benchmark IDS data set.

The paper is organized as follows. Section 2 illustrates a brief literature review of some of recent Hybrid Meta-heuristic approaches used by IDS. Section 3 provides an overview of LSTM-ARA-HSA methodologies. Section 4 discusses an overview of the results of testing a performance valuation of proposed hybrid approach. Section 5 lists the key findings and conclusions of this paper.

II. Literature Review

Review of recently developed IDS approaches for cloud environments is described below.

Sethi *et al.* [11] presented a DRL enabled adoptive cloud IDS model that implements fine-grained classification and precise detection of complex and new attacks.

Ji *et al.* [12] introduced a network IDS by integrating asymmetric convolution AE and RF. This method could integrate the benefits of Shallow and Deep Learning.

Singh and Ranga [13] deliberated a robust network driven IDS employing ensemble-based ML method with four classifications, namely voting scheme, RUSBooted, bagged tree, boosted tree, and subspace discriminant. The voting method was integrated with architecture for obtaining the final prediction.

Alkadi *et al.* [14] devised DBF method for offering privacy with blockchain and security-based distributed IDS by the smart contracts of IoT networks. IDS were utilized using BiLSTM-DL technique for managing the consecutive network data, which is computed by the datasets of BoT-IoT and UNSW-NB15.

Zhong *et al.* [15] presented a novel method using features of model. The method would gather features in network layer through tcpdump packet and a application layer through system routine. GRU and Text-CNN approaches are selected since they could process consecutive data as a language system.

Samriya and Kumar [16] proposed a hybridization method for IDS for improving entire security level of cloud computing platforms. Additionally, this technique assists in handling different kinds of security problems i.e., detection of fake identity and data leakage Phishing attacks to retain the security over cloud environment.

Abusitta *et al.* [17] presented an ML based IDS that effectively uses the previous feedback information to improve a capacity for proactive decision making. Especially, presented method was depended by Denoising Autoencoder (DA) that is employed as an element to create DNN method.

III. Proposed Model

In this research, use of a hybrid meta-heuristic approach-combining Machine Learning techniques, named ExtraTrees Feature Selection method and LSTM classification model with Computational Intelligence Algorithms named Artificial Raindrop Algorithm and Harmony Search Algorithm- is proposed for optimal classification and prediction of network attacks in the cloud environment.

The overall model encompasses the following processes:

1) Pre-processing of all available data dimensions in the data-set

2) Selection of appropriate data dimensions using ExtraTrees feature selection method

3) Hyper-parameter optimization for the selected data dimensions using Artificial Raindrop Algorithm and Harmony Search Algorithm (ARA-HSA)

4) Final LSTM based classification by IDS into Attack/ Not an Attack, based on above processes

Fig. 2 illustrates the overall working process of LSTM-ARA-HSA technique for IDS. The detailed working of each of the above processes is elaborated in the succeeding sections.

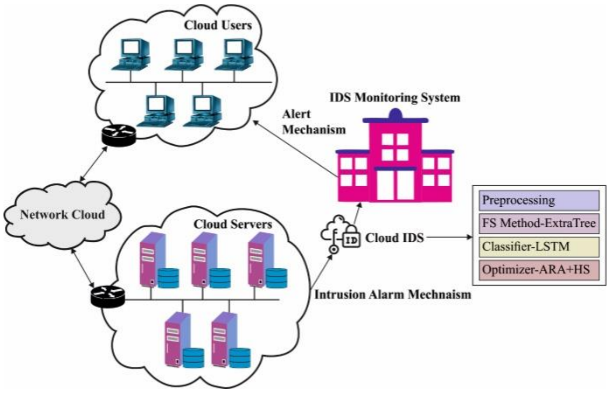

Fig. 2. Overall process of LSTM-ARA-HSA technique

A. Pre-Processing of All Available Data Dimensions

During this process, Min_max normalization was utilized to standardize the different scales for all the available data dimensions in the benchmark IDS data set. The normalization process transforms the data points with reference to the data range to which they belong, by implementing linear transformation. The value of each data point in a dimension was normalized to a standard range between zero and one, utilizing Min_max normalization as per the following formula [18]:

$$t = \frac{v - \min_d}{\max_d - \min_d}(tran\_\max_d - tran\_\min_d) + tran\_\min_d \quad (1)$$

where t refers to the transformed value of data value v from dimension d, $\min_d$ refers to the existing minimum value for the data range for dimension d, $\max_d$ refers to the existing maximum value for the data range for dimension d, $tran\_\min_d$ refers to the standardized minimum value (i.e. zero) for the transformed data range for dimension d, and $tran\_\max_d$ refers to the standardized maximum value (i.e. one) for the transformed data range for dimension *d*.

B. Selection of Appropriate Data Dimensions using ExtraTrees Feature Selection Method

ExtraTrees (ET) is ML type technique that aggregates several de-correlated decision trees together in "forest" to output it's classified results, presented by Geurts et al. [19]. ET is used during this process for the purpose of feature selection, i.e. choosing a sub-set of proper dimensional data from the full set of data dimensions.

Each Decision Tree (DT) in ExtraTrees Forest is created from raw training sample, and every node, every tree is given by random sample of the number of features to select from full set- say "k" features from full features-set where every tree must choose optimal feature to divide the data using defined mathematical criteria. Thus, this random feature led to the generation of several de-correlated decision trees. To fulfil the feature selection using process through this forest structure, the defined total minimization in the defined mathematical factors (such as the Gini coefficient) applied to the decision of feature of split is calculated for every feature throughout the forest construction [20].

The benefits of ET model are reduced variance of the DT and computation effectiveness. For each of the available data dimensions, the standardized reduction of the Gini coefficient utilized for splitting the feature decision is estimated. Afterward, the Gini coefficient is ranked in descending order, and the first k features can be chosen. In this way, the number of the sub-set of data dimensions- specified as k in number- can be selected out of the full set of available data dimensions in the benchmark IDS data set.

C. Hyper-Parameter Optimization for the Selected Data Dimensions using Artificial Raindrop Algorithm and Harmony Search Algorithm (ARA-HSA)

For the hyper-parameter optimization or tuning process, a combination of the ARA-HSA algorithms is proposed to be utilized.

The natural occurrence of rainfall served as the inspiration for the ARA, which has been proven to be an effective tool for resolving single-objective optimization issues [21]. The fundamental premise is to model the altitude of the raindrop from where it is initiated to a final state when has minimal altitude and hence has its minimal energy state. The state of minimal altitude is considered to be an optimal state or solution. ARA begins with initiation of the primary population of N raindrops as arbitrarily placed N vapors in the search space, with all vapors having a vapor place determined as follows:

$$Vapor_i = \left(x_i^{(1)}, \cdots, x_i^{(d)}, \cdots, x_i^{(D)}\right), i = 1,2,\cdots, N. \quad (2)$$

Where N refers to the population size, D refers to the number of dimensions selected for hyper-parameter optimization, and $x_i^{(d)}$ refers to the place of i^{th} vapor from the d^{th} dimension.

Since raindrops are created by constantly absorbing ambient water vapor naturally, the place of each raindrop is determined as follows:

$$Raindrop = \left(\frac{1}{N}\sum_{i=1}^{N} x_i^{(1)}, \cdots, \frac{1}{N}\sum_{i=1}^{N} x_i^{(d)}, \cdots, \frac{1}{N}\sum_{i=1}^{N} x_i^{(D)}\right) \quad (3)$$

If the effect of external factors is not taken into account, each raindrop falls from its initial altitude in the cloud, to the ground with free-fall due to gravity. This can be modeled as place of each Raindrop being modified to a new place, represented as $New_$Raindrop. Assuming that Raindro$p^{(d_i)}$ is the place of a Raindrop from $d_i th$ dimension, where $d_i(i = 1,2,3,4)$belongs to the set $\{1, 2, \cdots, D\}$,New_$Raindrop^{(d_1)}$ is attained by linear combination of Raindro$p^{(d_2)}$, $Raindrop^{(d_3)}$ and Raindro$p^{(d_4)}$, and determined as follows:

$$\begin{cases} New\_Raindrop^{(d)} = Raindrop^{(d_2)} + \varphi \cdot (Raindrop^{(d_3)} - \underset{Raindrop}{}^{(d_4)}), if\ d = d_1; \\ New\_Raindrop^{(d)} = Raindrop^{(d)}, otherwise. \end{cases} \quad (4)$$

where φ refers the arbitrary number from the range -1 and 1 , $d = 1,2,\cdots,D$. Once the $New\_$ Raindrops contact the ground, it is split to small raindrops due to speed as well as quality. Afterward, these smaller raindrops ($Small - Raindrop_i, i = 1,2,\cdots,N$) may move randomly to any direction. For this reason, the place of $Small\_Raindrop_i$ is modeled as follows:

$$Small_{Raindrop_i} = New_{Raindrop} + sign(\alpha - 0.5) \cdot \log(\beta) \cdot (New\_Raindrop - Vapor_k) (5)$$

where$k$refers to the arbitrarily selected index in the set $\{1, 2, \cdots, N\}$, α and β refer to two uniformly distributed arbitrary numbers from the range zero to one, and the sign $(\cdot)$ refers to the sign function [22].

During the act of gravity, the movement of $Small\_Raindrop_i(i = 1,2,\cdots,N)$ is from higher altitude to lower altitude, and they finally stop at places with minimal altitude (i.e., an optimum solution). The Raindrops Pool (RP) is modeled to track raindrops to the places of minimal altitude as follows:

1) RP is initiated randomly to initial places in the search space;

2) A better solution for the RP is rationalized at the end of every iteration;

3) Once the RP size increases the threshold specified initially, any less optimal solution is deleted from the RP, to maintain the RP's size stable and lower the computation required.

The movement of raindrop d_i for $Small\_Raindrop_i(i = 1,2,\cdots,N)$ is modeled as dependent upon linear group of 2 vectors $d1_i$and $d2_i$, where $d_i, d1_i$ and $d2_i$ are explained as:

$$d1_i = sign(F(RP_{k_1}) - F(Small\_Raindrop_i)) \cdot (RP_{k_1} - Small\_Raindrop) \quad (6)$$

$$d2_i = sign(F(RP_{k_2}) - F(Small\_Raindrop_i)) \cdot (RP_{k_2} - Small\_Raindrop) \quad (7)$$

$$d_i = \tau_1 \cdot rand1_i \cdot d1_i + \tau_2 \cdot rand2_i \cdot d2_i \quad (8)$$

where RP_{k_1} and RP_{k_2} refer to two candidate solutions from RP $(k_1, k_2 \in \{1,2,\cdots,|RP|\})$, τ_1 and τ_2 refer to two step parameters of $Small\_Raindrop_i$ flowing, $rand1_{\text{i}}$ and $rand2_i$ refer to two uniformly distributed arbitrary numbers from the range zero to one, F indicates the Fitness Function. Thus the outcome, $New\_Small\_Raindrop_i(i = 1,2,\cdots,N)$ is determined as:

$$New\_Small\_Raindrop_i = Small\_Raindrop_i + d_i \quad (9)$$

For improving the computational efficiency and convergence rate of ARA, the N optimum solutions in $New\_Small\_Raindrop \cup Vapor$ were chosen utilizing sort technique, as the next vapor population. The flowchart of ARA is described in Fig. 3 [22].

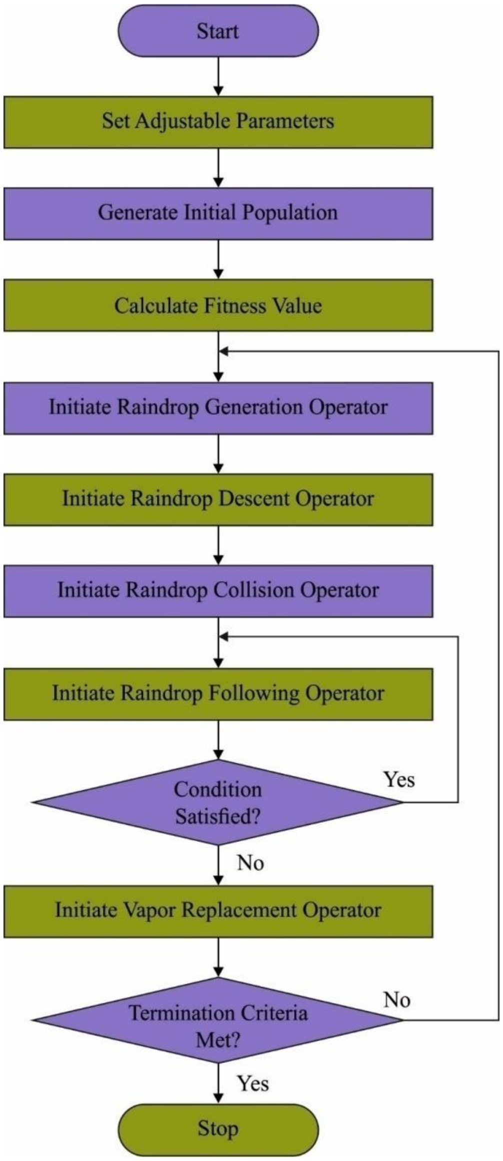
Start
Set Adjustable Parameters
Generate Initial Population
Calculate Fitness Value
Initiate Raindrop Generation Operator
Initiate Raindrop Descent Operator
Initiate Raindrop Collision Operator
Initiate Raindrop Following Operator
Condition Satisfied?
Yes
No
Initiate Vapor Replacement Operator
Termination Criteria Met?
No
Yes
Stop

Fig. 3. Flowchart of ARA

The performance of the ARA can be improved by updating every raindrop by the use of the individual model and Harmony Search Algorithm (HAS) based Harmony Search Memory (HM) operator.

HSA attempts several feasible optimal solutions based on use of the HM operator, similar to how multiple musicians attempt to achieve harmony in music based on their collective memory [23]. HSA has been simulated by employing rules of harmony improvisation [24]. It involves the steps as described below.

Step 1. Initializing of HS Memory (HM).

A primary HM has a particular number of arbitrarily created solutions for the optimized solution. For a n dimensional problem, the HM with size of N is modeled as:

$$HM = \begin{bmatrix} x_1^1, x_2^1, \dots, x_n^1 \\ x_1^2, x_2^2, \dots, x_n^2 \\ \vdots \\ x_1^{HMS}, x_2^{HMS}, \dots, x_n^{HMS} \end{bmatrix} \quad (10)$$

where $[x_1^i, x_2^i, \dots, x_n^i]$ $(i = 1, 2, \dots, HMS)$ is a solution candidate. Here, the HM was generally fixed between [50, 100].

Step 2. Improvising a solution $[x_1^{'}, x_2^{'}, \dots, x_n^{'}]$ in HM.

All the elements in the solution $x_j^{'}$ are attained dependent upon HM Considering Rate (HMCR). Here, the HMCR is determined as the possibility of choosing a module in HM member, and the 1-HMCR is, so, the possibility of creating it arbitrarily. Once $x_j^{'}$ appears into the HM, it can be selected in j^{th} dimension of arbitrary HM members and is mutated based on Pitching Adjust Rate (PAR) that defines the probabilities of the candidate in HM is mutated. Moreover, the improvisation of $[x_1^{'}, x_2^{'}, \dots, x_n^{'}]$ is related to the generation of issues from Genetic Algorithms (GA), and mutation as well as crossover functions. But, while GA generates novel chromosomes utilizing 1 (mutation) or 2 (simple crossover) pre-defined ones, the generation of novel solutions from HSA utilizes every HM member completely.

Step 3. Upgrading the HM. In this step, the Step 2 solution is estimated. Once it creates an optimum fitness which is worse than member from the HM, the existing member will be replaced. Else, the new member is discarded.

Step 4. Repeating Step 2 to Step 3 until a current end circumstance, i.e., a high number of iterations are met. The HSA is an arbitrary search approach and only employs a single search memory for evolving.

In order to optimize the hyper-parameter values for the selected data dimensions, the combination of ARA-HSA is used. The ARA-HSA combination computes a fitness function to accomplish optimization of the hyper-parameters. The value of the fitness function is a positive integer value, which represents the effective outcome of the candidate solution. In case of intrusion detection, the classifier error rate can be treated as the fitness function, as provided in Eq. (11) below. The optimal solutions hold minimum value of error rate and the non-optimal solutions hold maximum value of error rate.

$$fitness(x_i) = Classifier\ Error\ Rate(x_i)$$
$$\frac{number\ of\ misclassified\ instances}{Total\ number\ of\ instances} * 100 \quad (11)$$

D. LSTM based Classification

In this final process, the LSTM model can be utilized for the purpose of detecting and classifying the intrusions in cloud environment. LSTM is a kind of Recurring Neural Network (RNN) i.e., used for processing consecutive data and, it addressing a long-term memory problems of vanilla RNN. The LSTM expands the structure of RNN by a gating method and the standalone memory cell that normalizes the data flow in all over the networks. Here, a gating method includes output, input, and forget gates [25]. This gate controls the flow of data over the network to enables which data needed to continue or the period it would persevere afterward sensing it from the memory. The LSTM network is able to discard the insignificant data and preserves critical data. The memory cell provides a recurrent self-connected unit termed Constant Error Carousel (CEC), which offered a state vector to retain long-term dependency. Fig. 4 depicts the infrastructure of LSTM.

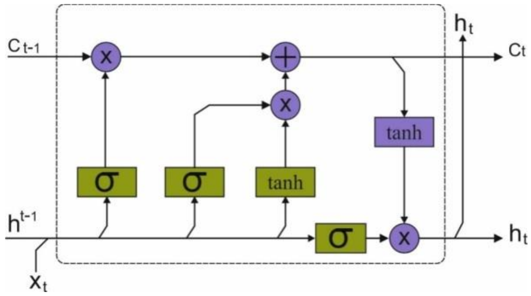

Fig. 4. LSTM structure

Then, to differentiate self-contained cell memory from state h_t in LSTM and, it is represented as c_t. The forget gate f_t obtains input x_t and h_{t-1} to define which data requires to be preserved in c_{t-1}. Activation function for gates i_τ, o_t and f_t are sigmoid layer whereas all the values are predicted among zero and one in which c_{t-1} provides the data preservation to determine the scale. But the above mentioned procedure is determined by Eqs. (12) to (16):

$$i_t = \sigma(W_{xi}x_\tau + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \quad (12)$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f\right) \quad (13)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_\tau + b_0) \quad (14)$$

$$c_t = f_\tau \circ c_{t-1} + i_t \circ \emptyset(W_{xc}x_\tau + W_{hc}h_{t-1} + b_c) \quad (15)$$

$$h_t = 0_\tau \circ \emptyset(c_t) \quad (16)$$

Now, $W_{xi}, W_{hi}, W_{ci}, W_{xf}, W_{hf}, W_{cf} W_{xo}, W_{ho}, W_{co}, W_{xc}$, and W_{hc} indicates weight matrices for the gate and cell memory state, whereas h_{t-1} indicates a preceding hidden state, and c_t denotes a cell state. The bias of the gate is denoted by b_i, b_o, b_f , and b_c whereas 0 indicates an element-wise multiplication process. Likewise, σ displays the logistic sigmoid function and $\emptyset$ signifies tangent function.

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (17)$$

$$\emptyset(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad (18)$$

IV. Experimental Validation

The experimental validation of the proposed hybrid meta-heuristic approach LSTM-ARA-HSA technique was done by two benchmark data sets; they are NSL-KDD and KDDCup'99 datasets. The NSL-KDD dataset holds 41 features, in which 2 features are symbolic records and other 39 features are numeric record. The data set contains Basic features: 1–9, Content features: 10–22, Traffic features: 23–31, and Host features: 32–41. The results were analyzed along different aspects. Comparative analysis of the proposed model with other recent techniques is also described below.

A. Results Analysis on NSL-KDD Dataset

This section elaborates the intrusion detection results of the proposed LSTM-ARA-HSA on the test NSL-KDD data set.

Fig. 5 showcases the FS result of the proposed model on the test NSL-KDD dataset. The figure displays the set of features elected by the proposed model.

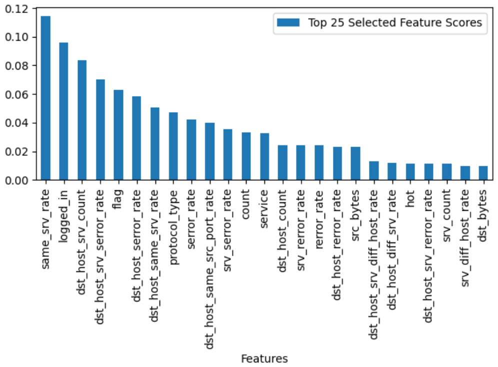

Fig. 5. Selected features and its scores of NSL-KDD dataset

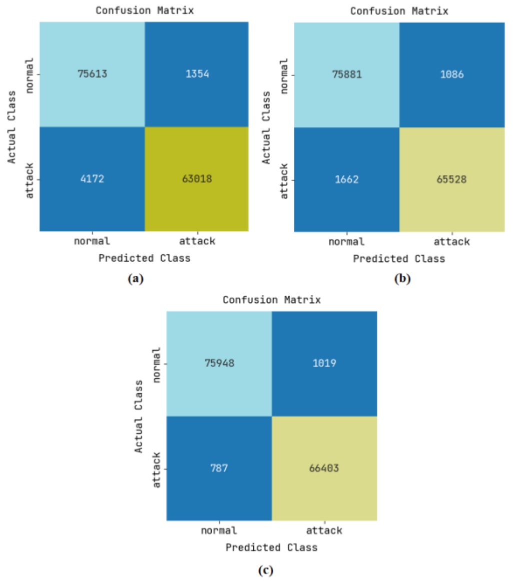

Fig. 6. Confusion matrix of NSL-KDD dataset a) LSTM b) FS+LSTM c) ARA+HS+FS+LSTM

Fig. 6 illustrates the confusion matrix in the proposed model with LSTM and FS+LSTM model. The figure indicates that the LSTM model classified a total of 75613 instances into Normal class and 63018 instances into Attack class. In addition, the FS+LSTM model classified a set of 75881 instances into Normal class and 65528 instances into Attack class. Lastly, the ARA+HS+FS+LSTM model classified a total of 75948 instances into Normal class and 66403 instances into Attack class.

The intrusion detection results of the proposed ARA+HS+FS+LSTM model on the NSL-KDD dataset are offered in Table I. From the table, it is proved that the LSTM model achieved an $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, AUC_{score} of 96.170%, 97.900%, 93.790%, 95.800%, and 96.020% respectively. The FS+LSTM model achieved slightly enhanced outcomes with the $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, AUC_{score} of 98.090%, 98.370%, 97.530%, 97.950%, and 98.060% respectively. Finally, the ARA+HS+FS+LSTM model achieved highest performance with the $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, AUC_{score} of 98.750%, 98.490%, 98.830%, 98.600%, and 98.750% respectively.

The ROC analysis of the ARA+HS+FS+LSTM model with its earlier versions on the test NSL-KDD dataset is shown in Fig. 7. From the figure, it is apparent that LSTM and FS+LSTM models provided reasonable ROC values of 0.9682 and 0.9886. The ARA+HS+FS+LSTM model provided ROC of 0.9875.

TABLE I. Result Analysis of ARA+HS+FS+LSTM Technique on NSL KDD Dataset

Methods	LSTM	FS+LSTM	ARA+HS+FS+LSTM
Accuracy	96.170	98.090	98.750
Precision	97.900	98.370	98.490
Recall	93.790	97.530	98.830
F1-Score	95.800	97.950	98.660
AUC Score	96.020	98.060	98.750

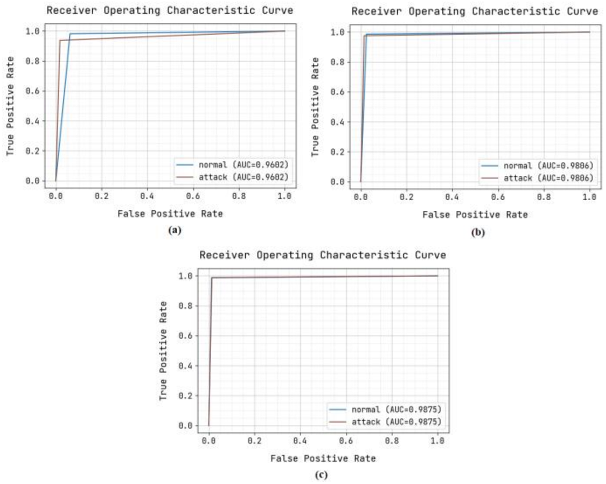

Fig. 7. ROC of NSL-KDD dataset a) LSTM b) FS+LSTM c) ARA+HS+FS+LSTM

A comprehensive comparative result analysis of the ARA+HS+FS+LSTM method with other existing techniques is made in Table II and Fig. 8 [26, 27]. The results indicated that DNN and DT models achieved the least intrusion detection performance. The PCA+DNN and RF models achieved moderately better intrusion detection performance. The XGBoost and Hybrid KPCA-SVM+GA techniques were next in terms of achievement. The Hybrid GA-SVM+PSO technique achieved near optimal outcomes. However, the ARA+HS+FS+LSTM technique achieved the highest performance results compared to other techniques, with the $accu_y, prec_y$, $reca_l$, and $F1_{score}$ of 98.75%, 98.49%, 98.83%< and 98.66%, respectively.

TABLE II. COMPARATIVE ANALYSIS OF ARA+HS+FS+LSTM TECHNIQUE WITH EXISTING APPROACHES ON NSL-KDD DATASET

Methods	Accuracy	Precision	Recall	F1-Score
ARA+HS+FS+LSTM	98.75	98.49	98.83	98.66
PCA+DNN	93.80	93.40	91.80	93.70
DNN Model	91.40	89.10	88.20	90.50
Random Forest	93.60	90.00	82.00	94.60
XGBoost	95.50	92.00	98.00	95.55
Decision Tree	92.89	90.20	85.00	94.50
Hybrid GA-SVM+PSO	96.38	94.90	97.77	96.11
Hybrid KPCA-SVM+GA	95.26	94.03	96.39	95.47

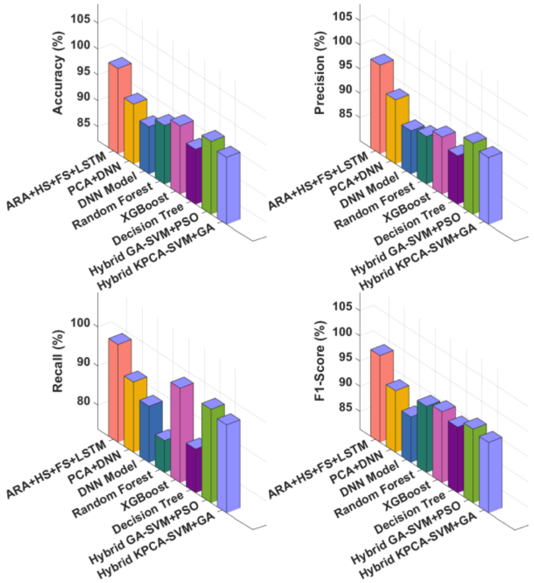

Fig. 8. Comparative analysis of proposed method on NSL-KDD dataset

B. Results Analysis on KDDCup Dataset

This section elaborates the intrusion detection results of the proposed LSTM-ARA-HSA on the test KDDCUP99 data set.

Fig. 9 depicts the FS results of the proposed technique in test KDDCUP99 dataset. The figure indicates the set of features chosen by the presented method.

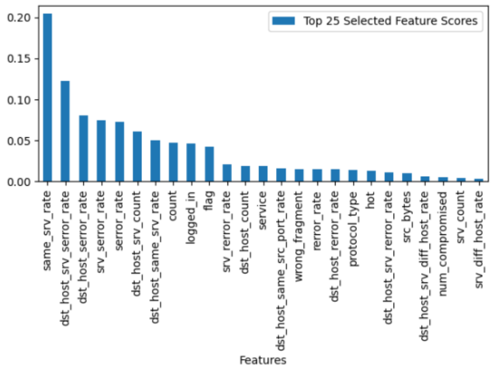

Fig. 9. Selected features and its scores of KDDCup99 dataset

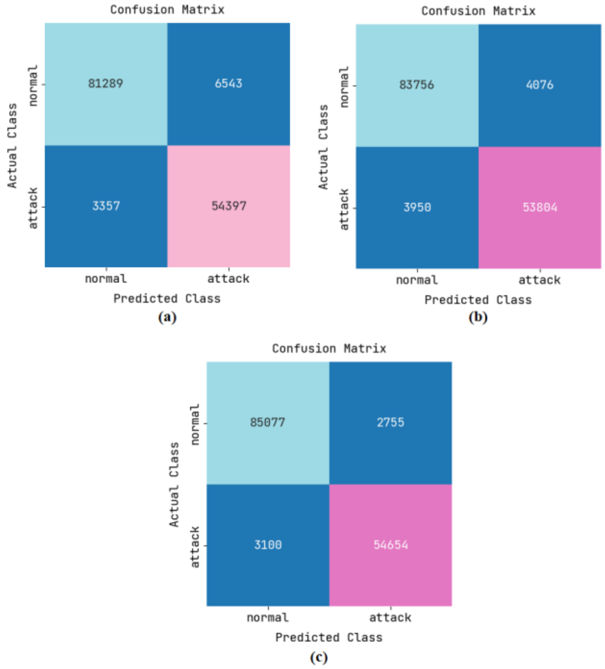

Fig. 10. Confusion Matrix of KDDCup 99 dataset a) LSTM b) FS+LSTM c) ARA+HS+FS+LSTM

Fig. 10 demonstrates the confusion matrix of the proposed algorithm with LSTM and FS+LSTM method. The figure showcased that the LSTM model has identified a total of 81289 instances into Normal class and 54397 instances into Attack class. Also, the FS+LSTM model has categorized a set of 83756 instances into normal class and 53804 instances into Attack class. At last, the ARA+HS+FS+LSTM model has classified a total of 85077 instances into Normal class and 54654 instances into attack class.

The intrusion detection outcomes of the proposed ARA+HS+FS+LSTM approach on the KDDCUP99 dataset is indicated in Table III. From the table, it is apparent that the LSTM model achieved $accu_y$, $prec_n, reca_l$, $F1_{score}$, AUC_{score} of 93.200%, 89.260%, 94.190%, 91.660%, and 93.370% correspondingly. The FS+LSTM method achieved slightly improved outcomes with $accu_y$, $prec_n, reca_l$, $F1_{score}, AUC_{score}$ of 94.490%, 92.960%, 93.160%, 93.060%, and 94.260% correspondingly. Finally, the

ARA+HS+FS+LSTM algorithm achieved the highest performance with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, AUC_{score} of 95.980%, 95.200%, 94.630%, 94.920%, and 95.750%, respectively.

TABLE III. RESULT ANALYSIS OF ARA+HS+FS+LSTM TECHNIQUE ON KDDCUP99 DATASET

Methods	LSTM	FS+LSTM	ARA+HS+FS+LSTM
Accuracy	93.200	94.490	95.980
Precision	89.260	92.960	95.200
Recall	94.190	93.160	94.630
F1-Score	91.660	93.060	94.920
AUC Score	93.370	94.260	95.750

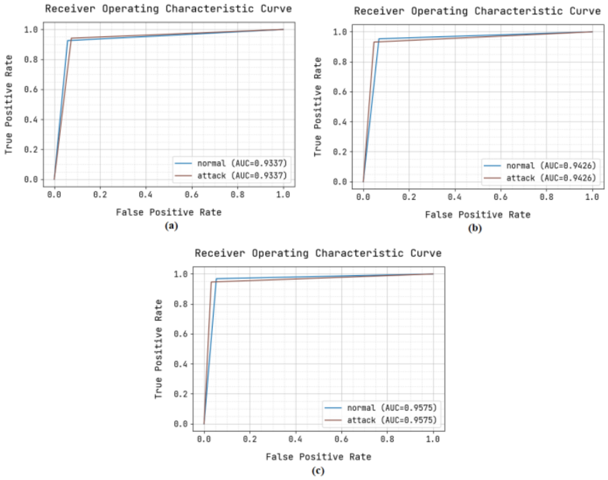

Fig. 11. ROC of KDDCup 99 dataset a) LSTM b) FS+LSTM c) ARA+HS+FS+LSTM

The ROC analysis of the ARA+HS+FS+LSTM technique along with other techniques on the test KDDCUP99 dataset is exhibited in Fig. 11.

From the figure, it is evident that the LSTM and FS+LSTM methods achieved ROC values of 0.9337 and 0.9426. The ARA+HS+FS+LSTM technique achieved ROC of 0.9575.

TABLE IV. COMPARATIVE ANALYSIS OF ARA+HS+FS+LSTM TECHNIQUE WITH OTHER EXISTING METHODS ON KDDCUP99 DATASET

Methods	Accuracy	Precision	Recall	F1-Score
ARA+HS+FS+LSTM	95.98	95.20	94.63	94.92
PCA+DNN	89.80	88.40	89.80	88.20
DNN Model	90.90	89.60	90.90	89.40
Random Forest	93.66	92.46	93.12	94.14
XGBoost	94.06	92.94	93.53	94.50
Decision Tree	93.00	91.28	92.48	92.62
IARADR	94.59	92.90	94.05	94.24
ML-SHNS	93.91	92.68	93.41	94.22

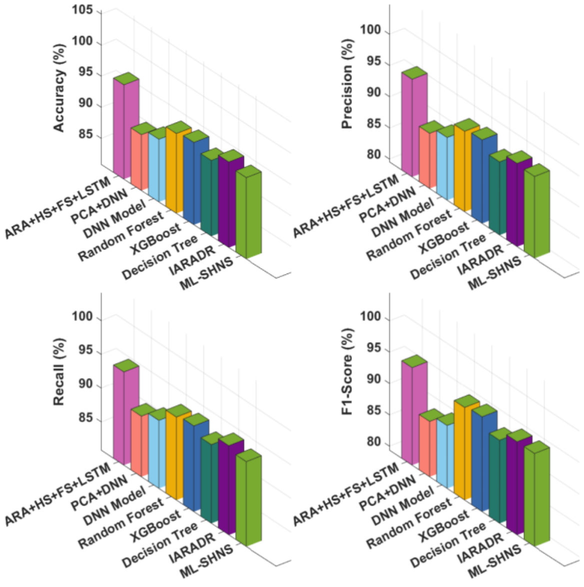

Fig. 12. Comparative analysis of proposed method on KDDCUP99 dataset

A detailed comparative result analysis of the ARA+HS+FS+LSTM method with other existing algorithms is made in Table IV and Fig. 12 [28, 29].

The results indicated that DNN and DT techniques achieved the least intrusion detection performance. The PCA+DNN and RF approaches achieved moderately improved intrusion detection performance. The XGBoost and IARADR techniques were next in terms of achievement. The ML-SHNS approach achieved near optimal outcomes. The ARA+HS+FS+LSTM technique achieved the highest performance results with $accu_y$, $prec_y$, $reca_l$, and $F1_{score}$ of 98.98%, 95.20%, 94.63%, and 94.92% correspondingly.

After evaluating the performance results of the proposed LSTM-ARA-HSA model along with other techniques and approaches, it is apparent that this approach is a feasible hybrid meta-heuristic approach that can be used by cloud-based IDS.

V. CONCLUSION

This paper has proposed a hybrid meta-heuristic approach combining ML techniques of ExtraTrees Feature Selection method and LSTM classification model with the Computational Intelligence Algorithms ARA-HAS, to develop the performance of cloud-based Intrusion Detection Systems and detect network attacks accurately and efficiently. These hybrid methods encompasses different processes such as Pre-processing of data dimensions, ET based feature selection, ARA-HSA based hyperparameter tuning, and LSTM based classification.

The adoption of ARA-HSA enables optimization of hyper-parameters for the dimensions selected using the ET Feature Selection method, in turn enabling more accurate LSTM classification.

The experimental outcome shows that the proposed LSTM-ARA-HSA has better performance than other techniques in terms of accuracy and efficiency. Therefore, this hybrid approach can be utilized by IDS for detecting and classifying the intrusions in the cloud environment. The objective of the current paper is to only establish that the proposed hybrid approach results in more accurate classification of attacks and the scope is limited to experimental validation using the two main datasets. As a part of future scope of research, the detection efficiency of this hybrid approach can be improved through appropriate design of clustering and outlier detection techniques.

REFERENCES

[1] K. Jain, S. Jain, A. Guha, and A. Patra, “An approach to early stage detection of atherosclerosis using arterial blood pressure measurements”, Biomedical Signal Processing and Control, vol.68, pp.102594, 2021.

[2] V.S.H. Rao, M.N. Kumar, “Novel approaches for predicting risk factors of atherosclerosis”, IEEE J. Biomed. Health Inform, vol.17, pp. 183–189, 2012.

[3] A.K. Gárate-Escamila, A.H. El Hassani, and E. Andrès, “Classification models for heart disease prediction using feature selection and PCA”, *Informatics in Medicine Unlocked*, no.*19*, pp.100330, 2020.

[4] O. Terrada, B. Cherradi, A. Raihani, and O. Bouattane, “A novel medical diagnosis support system for predicting patients with atherosclerosis diseases”, Informatics in Medicine Unlocked, vol.21, pp.100483, 2020.

[5] R. Rajagopal, and V. Ranganathan, “Evaluation of effect of unsupervised dimensionality reduction techniques on automated arrhythmia classification”, Biomed Signal Process Contr, vol.34, pp.1–8, 2017.

[6] U.R. Acharya, V.K. Sudarshan, J.E. Koh, R.J. Martis, J.H. Tan, S.L. Oh, A. Muhammad, Y. Hagiwara, M.R.K. Mookiah, K.P. Chua, et al. “Application of higher-order spectra for the characterization of coronary artery disease using electrocardiogram signals”, Biomed. Signal Process. Control, vol.31,pp.31–43,2017.

[7] M.A. Rahhal, Y. Bazi, H. Alhichri, N. Alajlan, F. Melgani, and R. Yager, “Deep learning approach for active classification of electrocardiogram signals”, InfSci, vol.345, pp. 340–54, 2016.

[8] K. Jain, S. Maka, and A. Patra, “Modeling of cardiovascular circulation for the early detection of coronary arterial blockage”, Math. Biosci. 304 (2018) 79–88.

[9] Y.K. Qawqzeh, M.M. Otoom, F. Al-Fayez, I. Almarashdeh, M. Alsmadi, and G. Jaradat, “A proposed decision tree classifier for atherosclerosis prediction and classification”, IJCSNS, vol. 19, pp.197, 2019.

[10] K. Jain, A. Patra, and S. Maka, “Modeling of the human cardiovascular system for detection of atherosclerosis”, IFAC-PapersOnLine, vol 51,pp.545–550, 2018.

[11] K. Sethi, R. Kumar, N. Prajapati, and P. Bera, “Deep reinforcement learning based intrusion detection system for cloud infrastructure”, In proceedings of International Conference on COMmunication Systems &NETworkS (COMSNETS), pp.1-6, 2020.

[12] S. Ji, K. Ye, and C.Z. Xu, “A Network Intrusion Detection Approach Based on Asymmetric Convolutional Autoencoder”, In proceedings of nternational Conference on Cloud Computing, Springer, Cham, pp. 126-140, September 2020.

[13] Singh P. and Ranga V. “Attack and intrusion detection in cloud computing using an ensemble learning approach”, International Journal of Information Technology, vol.13, no.2, pp.565-571, 2021.

[14] O. Alkadi, N. Moustafa, B. Turnbull, and K.K.R. Choo, “A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks”, IEEE Internet of Things Journal, vol.8, no.12, pp.9463-9472, 2020.

[15] M. Zhong, Y. Zhou, and G. Chen, “Sequential model-based intrusion detection system for IoT servers using deep learning methods”, Sensors, vol.21, no.4, pp.1113, 2021.

[16] J.K. Samriya, and N. Kumar, “A novel intrusion detection system using hybrid clustering-optimization approach in cloud computing”, Materials Today: Proceedings, 2020.

[17] A. Abusitta, M. Bellaiche, M. Dagenais, and T. Halabi, “A deep learning approach for proactive multi-cloud cooperative intrusion detection system”, Future Generation Computer Systems, vol.98, pp.308-318, 2019.

[18] N. Khare, P. Devan, C.L. Chowdhary, S. Bhattacharya, G. Singh, S. Singh, and B. Yoon, “Smo-dnn: Spider monkey optimization and deep neural network hybrid classifier model for intrusion detection”, Electronics, vol.9, no.4, pp.692, 2020.

[19] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees”, Mach. Learn, vol. 63, pp.3–42, 2006.

[20] E.K. Ampomah, Z. Qin, and G. Nyame, “Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement”, Information, vol.11, no.6, pp.332, 2020.

[21] Q. Jiang, L. Wang, X. Hei, G. Yu, and Y. Lin, “The performance comparison of a new version of artificial raindrop algorithm on global numerical optimization”, Neurocomputing, vol.179, pp.1-25, 2016.

[22] Y. Huang, and Y. Qiao, “Artificial raindrop algorithm for optimal parameter preference in digital IIR filters”, Adv. Model Anal. C, vol.72, pp.114-39, 2017.

[23] Z.W. Geem, J.H. Kim, and G.V. Loganathan, “A new heuristic optimization algorithm: harmony search”, simulation, vol.76, no.2, pp.60-68, 2001

[24] X.Z. Gao, V. Govindasamy, H. Xu, X. Wang, and K. Zenger, “Harmony search method: theory and applications”, Computational intelligence and neuroscience, 2015.

[25] N. Khan, I.U. Haq, F.U.M. Ullah, S.U. Khan, and M.Y. Lee, “CL-Net: ConvLSTM-Based Hybrid Architecture for Batteries”, State of Health and Power Consumption Forecasting, Mathematics, vol. 9,pp.33262021,2015.

[26] Moukhafi M, El Yassini K, Bri,S, “A novel hybrid GA and SVM with PSO feature selection for intrusion detection system”, Int. J.Adv. Sci. Res. Eng, vol.4, pp.129–134, 2018.

[27] F. Kuang, W. Xu, and S. Zhang, “A novel hybrid KPCA and SVM with GA model for intrusion detection. Appl. Soft Comput. J. 2014, 18, 178–184.

[28] M. Pozi, M.N. Sulaiman, N. Mustapha, and T. Perumal, “Improving Anomalous Rare Attack Detection Rate”, Neural Process. Lett., vol.44, pp.279–290, 2015.

[29] A. Khan, S. Abbas, A. Rehman, Y. Saeed, A. Zeb, M.I. Uddin, N. Nasser, and A. Ali, “A machine learning approach for blockchain based smart home networks security”, IEEE Netw. vol.35, pp. 223–2292021.