Improved Chicken Swarm Optimized Recurrent Neural Networks for Big Data Intrusion Detection System

P.Sudha, R.Gunavathi

Abstract: ***Cyber security threats are an ever increasing, frequent and complex issue in the modern information era. With the advent of big data, incremental increase of huge amounts of data has further increased the security problems. Intrusion Detection Systems (IDS) were been developed to monitor and secure the cyber data systems and networks from any intrusions. However, the intrusion detection is difficult due to the rapid evolution of security attacks and the high volume, variety and speed of big data. In addition, the shallow architectures of existing IDS models lead to high computation cost and high memory requirements, thus further diminishing the efficiency of intrusion detection. The recent studies have suggested the use of data analytics and the deep learning algorithms can be effective in improving the IDS. An efficient IDS model is developed in this study by using the improved Elman-type Recurrent Neural Networks (RNN) in which the Improved Chicken Swarm Optimization (ICSO) optimally determines RNN parameters. RNN is an efficient method for classifying network traffic data but its traditional training algorithms are slow in convergence and faces local optimum problem. The introduction of ICSO with enhanced global search ability significantly avoids those limitations and improves the training process of RNN. This optimized deep learning algorithm of RNN, named as ICSO-RNN, is employed in the IDS with Intuitionistic Fuzzy Mutual Information feature selection to analyze larger network traffic datasets. The proposed IDS model using ICSO-RNN is tested on UNSW NB15 dataset. The final outcomes suggested that ICSO-RNN model has high performance in intrusion detection, with minimum training time and is proficient for big data.***

Keywords: ***Deep learning, Intrusion Detection Systems, Improved Chicken Swarm Optimization, Recurrent Neural Networks.***

I. INTRODUCTION

Cyber security is the important necessity for computers, data and networks to protect against the attackers. For ensuring cyber security, a set of technologies have been applied including the anti-virus software, firewall and IDS [1]. Detecting the anomalies and intrusions is through a hardware/software that monitors and detects the unauthorized logins, intruders, data corruptions, destruction and anomalous processes. Traditional IDS were developed for tackling these issues in normal network environments [2]. Recently, the advanced developments in communication networks and Internet-of-Things (IoT) has led to the huge amount of data generated each day in all major fields with varying volumes and types to form the big data [3]. These big data environment create maximum possible scenarios of abnormality, making the detection of the attacks very difficult compared to that of the traditional IDS becomes complex and inefficient in dealing big data [4].

Many studies were conducted to develop efficient IDS for network traffic. Most authors employed the machine learning techniques [5]. Decision tree [6] and Naïve Bayes [7] were the most common traditional algorithms. Recent studies also used Support Vector Machine (SVM) [8], Random forests [9], Extreme learning machine (ELM) [10], etc. for intrusion detections. However, after extensive research, it has been found that the generalized machine learning techniques were not efficient to tackle the big data intrusion detection. The main limitations were due to the shallow architecture of these machine learning algorithms [11]. The studies also suggested the use of deep learning methods for this problem can be viable solution [12]. Hence this paper focuses on utilizing an improved and hybrid deep learning algorithm for building competent IDS.

For the proposed objective, the deep learning Recurrent Neural Networks (RNN) [13] has been selected as it is considered ideal for real-time big data classification and dynamic system modeling like the IDS. However, when analyzed, the RNN has been found to be difficult to implement due to the complexity of training with presence of feedback connections. Even the improved Elman type RNN does not overcome this limitation. Hence an efficient automated algorithm must be developed to determine the optimal structure and parameter values. The proposed ICSO algorithm overcomes the limitations of existing CSO algorithm and efficiently trains the RNN for the intrusion detection. For ideal feature selection, Intuitionistic fuzzy mutual information feature subset selection [14] process is applied. The selected features are fed to the ICSO-RNN for training and then applied on the testing data. The remainder of this manuscript is structured as: Related works in section 2. Explanation of the proposed ICSO-RNN in section 3 followed by the evaluation results in section 4. Section 5 concludes this article.

Revised Manuscript Received on April 30, 2020.
***** Correspondence Author
Mrs. P.Sudha*, Department of Computer Science, Sree Saraswathi Thyagaraja College, Pollachi - 642 107, Tamil Nadu, India. Email: sudha.sabariananth@gmail.com
Dr. R.Gunavathi,,Head, Master of Computer Science and Applications, Sree Saraswathi Thyagaraja College, Pollachi - 642 107, Tamil Nadu, India. Email: hodmca@stc.ac.in

II. RELATED WORK

Recent studies have employed intrusion detection systems using deep learning models. Some prominent methods have developed optimized and hybrid deep learning based IDS. Yin et al. [15] suggested an IDS using deep learning RNNs. This approach considered the features of NSL-KDD dataset and normalized them to increase RNN performance. The RNN based IDS provided high detection accuracy in both binary and multi-class classification. But the limitation of using RNN for intrusion detection is that the training time is high due to slow acceleration. Shone et al. [16] designed an IDS using deep learning approach of non-symmetric deep auto-encoder (NDAE) based shallow learning. This approach of IDS used the Random forest and provided higher accuracy when applied on KDD Cup '99 and NSL-KDD datasets. However, this model cannot detect the zero-day attacks.Al-Qatf et al. [17] presented a network IDS using the combination of sparse auto-encoder and SVM. This approach utilized the self-taught learning of sparse auto-encoder for feature extraction to build SVM for classification. This hybrid approach was applied on NSL-KDD dataset and it provided better detection accuracy. However, the training and testing time of this model is high when applied for the whole dataset.

Wu et al. [18] developed IDS using convolutional neural networks (CNN) applied on NSL-KDD dataset for large networks. This CNN based model selects the traffic features automatically and accurately classifies the traffic into normal and abnormal classes with effective solution to the imbalance dataset problem. However, this model consumes more time for larger networks. Abdulhammed et al. [19] proposed an IDS model using the Deep Neural Networks (DNN) along with Random Forest algorithm and Stacking machine learning classifiers. This model effectively identifies the intrusion and also solves the imbalance network dataset problem. The evaluation results obtained on CIDDS-001 dataset showed that this hybrid model has higher accuracy and fewer errors with efficient solution to imbalance class problem. However, the other machine learning algorithms used in this approach namely Voting and Variational Auto-Encoder (VAE) have practical limitations to be used in this approach as they increase the computational complexity. Marir et al. [20] designed distributed IDS using deep belief network (DBN) along with the ensemble SVM classifiers. This hybrid deep learning model employs the DBN for dimensionality decline and SVM for network traffic data. Then the Hadoop HDFS system predicts the abnormal classes. The evaluations were performed on many intrusion datasets and the results showed that this hybrid IDS provided high detection accuracy for all considered intrusion data. However, the training time is high in this model for large datasets.

Khan et al. [21] proposed two-stage deep learning IDS using stacked auto-encoder and soft-max classifier. This two-stage model initially classifies the normal and abnormal traffic in first stage based on probability score. Then in second stage, the normal and abnormal class attacks are detected using deep neural networks. The evaluations performed on KDD99 and UNSW-NB15 datasets showed that high attack recognition rates are obtained for both datasets. However, this model is trained only on binary classes but not with multiple attack classes. Al Jallad et al. [22] developed Networking Chat-bot for intrusion detection by using Long Short Term Memory (LSTM). This intrusion detection model utilizes the concepts of distributed deep learning along with the natural language processing, context analysis, and big data to identify the abnormal traffic. This model was tested on MAWI dataset and the results showed promising detection rate with low false positive rate. However, this model was evaluated only on smaller data subsets and not on the complete dataset due to hardware limitations.

Garg et al. [23] developed IDS using hybrid deep learning model of Improved Convolutional Neural Network (ImCNN) and Improved Grey Wolf Optimization (ImGWO) for intrusion detection. The ImGWO, designed by improving exploration, ex ploitation, and random population generation, is employed for feature selection while revamped dropout layer ImCNN classifies the data to identify the network anomalies. The results obtained on DARPA'98 and KDD'99 datasets showed that this model improved the detection rate with higher accuracy. The only limitation of this model is that it has design complexities for real-world implementation. Hassan et al. [24] introducedIDS using deep learning CNN for relevant feature extraction and a Weight-Dropped, Long Short-Term Memory (WDLSTM) network for classify the big traffic data. The main advantage of this model is the use of drop-connect regularization and hyper-parameter optimization to avoid the over fitting problem which is a very challenging problem in big data analysis. Although high accuracy of intrusion detection is obtained, the computation complexity and time complexity of this model is significantly higher.

The IDS models in literature showed that the deep learning algorithms have provided efficient performance. However, they also suffer from some limitations namely computation complexity, time complexity and implementation limitations. Based on these inferences, this proposed research study aims at developing efficient IDS for network traffic attacks using ICSO-RNN. This has been developed in the visualization of achieving highly accurate intrusion detection with less complexity and time consumption.

III. METHODOLOGY

The proposed ICSO-RNN model for intrusion detection has been modeled by using the Elman type RNN whose structure and parameters are optimized by the improved CSO algorithm. The working model of the proposed ICSO-RNN is shown in Fig.1. Initially, the data are normalized and then the features are extracted. A total of 49 features are extracted and then the Intuitionistic Fuzzy Mutual Information Feature Subset Selection is performed. The final selected features are fed to the classifier for training and then applied on the testing data. The ICSO optimally selects the parameter values and the ideal structure for this work.

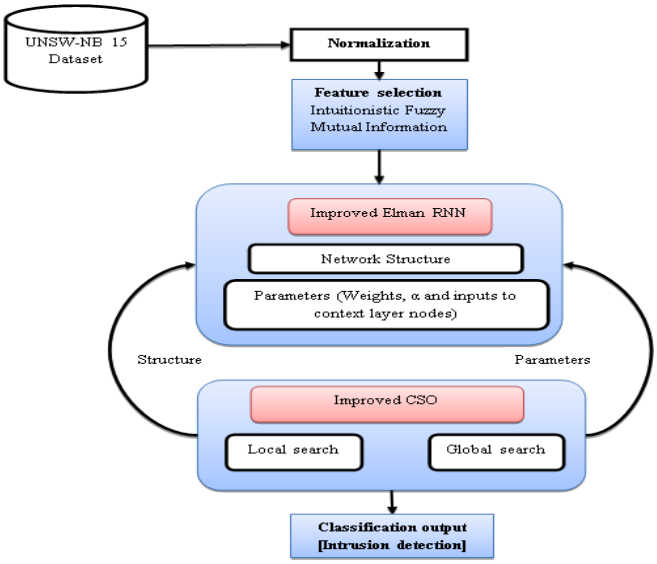

Fig. 1.Working model of ICSO-RNN

A. Architecture of Elman Recurrent Neural Networks

A RNN is a deep learning artificial neural network with feedback relations between its nodes. The most effective model of RNN is the Elman-type RNN. It is a partial RNN with an input, a hidden, a context and an output layers. Context layer nodes are used as memory units and the hidden layer nodes get the outputs of context and input layer nodes. Although efficient, the Elman RNN can only support single-order dynamic systems which are insufficient for the intrusion detection models. Hence an improved Elman RNN [25] is considered in this work. This improved Elman RNN has self-feedback connections based on context nodes with fixed coefficients and is must suitable for multi-order dynamic systems like IDS. This RNN model has introduced self-feedback coefficients with enhanced memorization ability which greatly increases the convergence accuracy at minimized learning time. The basic functions of each neuron layers of this improved Elman RNN are defined based on the propagation of data signals.

Input layer: The input layer neurons obtain the input from external sources and provide output which are represented as

$$x_i^I(k) = f\left(net_i^I(k)\right) \quad (1)$$

$$net_i^I(k) = u(k) \qquad i = 1,2,...,I \quad (2)$$

Here k denotes the k-th iteration number, represent an input linear function, denote the output and indicate the input of the layer, denotes the neuron processed objective.

Hidden layer: In hidden layer, the input and output neurons are denoted based on the context layer and input layer. They are given as

$$x_i^H(k) = S(net_J^H(k)) \quad (3)$$

$$net_j^H(k) = \sum_i W_{ji}^{HI}(k) \times x_i^I(k) + \sum_i W_{jr}^{HC}(k) \times x_r^C(k) \quad (4)$$

Here $net_j^H(k)$ and $x_i^H(k)$ represent the input and output of the hidden layer, respectively. $x_r^C(k)$ denote the output of the context layer, W_{jr}^{HI}, represent the weight matrix amongst input with hidden layer, W_{jr}^{HC} denote the weight matrix amongst the context and hidden layer and S denote the sigmoid function given by $S(x) = 1/(1+e^{-x})$

Context layer: In context layer, the input and output nodes are signified as

$$x_r^C(k) = \alpha x_r^C(k-1) + x_i^H(k-1) \quad (5)$$

$$0 \le \alpha \le 1$$

Here α denote the self-feedback coefficient. When $\alpha = 0$ the improved Elman RNN resembles the original Elman RNN and hence this case is very rarely utilized.

Output layer: In output layer, the input and output nodes are signified as

$$y_q(k) = g(net_q^O(k)) \quad q = 1,2,...,n \quad (6)$$

$$net_q^o(k) = \sum_i W_{qi}^{OH}(k) \times x_j^H(k) \quad (7)$$

Here, $net_q^o(k)$ indicate the input and $y_q(k)$ indicate the output nodes. W_{qi}^{OH}, denotes the weight matrix amongst the hidden & the output **layers**, and g denote a linear function that determines the output layer.

B. Improved Chicken Swarm Optimization

CSO Algorithm

CSO [26] is based on the social hierarchy characteristics of the farm chickens. In the farms, the chickens flock together as a group. A male chicken or rooster acts as the leader of this group while hens and chicks are also members of this group. Each group select the leader male chicken based on its ability to search best food source. Likewise, the hens and chicks are also positioned based on their social characters. Chicken Swarm Optimization is modeled based on this hierarchy. In the initialization phase, the chicken swarm of size X is initialized. RX denotes the male chicken. The hens, the young chickens and the mother hens are indicated as HX, CX and MX, respectively. Each rooster in the group acts as an agent and it is accountable for the motion of chicken and its location update towards the food source. The best fitness chickens are only assigned as leader roosters, while the worst fitness chickens will become chicks. The remaining chickens take the role of hens. All X chickens are represented by their locations $x_{i,j}^t (i \in [1,2,...,X], j \in [1,2,...,D]$ at time phaset %G (t is a fraction of G time phases) and search foodstuff in a D-dimensional area. Based on the movement, the location of rooster is updated using

$$x_{i,j}^{t+1} = x_{i,j}^t * \left(1 + Randn(0,\sigma^2)\right) \quad (8)$$

$$\sigma^2 = \begin{cases} \exp\left(\frac{(f_s - f_i)}{|f_i| + \in}\right), otherwise, s \in [1,2,...,X, s \ne i] \end{cases} \quad (9)$$

Where $x_{i,j}^t$ is the best position until previous iteration t, $Rand\left(0,\sigma^2\right)$ is a Gaussian function with zero mean and standard deviation σ^2 . $\in$ is the least constant number, s is a male chicken's index which is arbitrarily nominated from the chicken group, f is the qualification value of the equivalent x. Here, if $f_i \le f_s$,then $\sigma^2 = 1$.

Similarly, the hens'location is updated by

$$x_{i,j}^{t+1} = x_{i,j}^{t} + S1 * Rand * \left(x_{r1,j}^{t} - x_{i,j}^{t}\right) + S2 * Rand * \left(x_{r2,j}^{t} - x_{i,j}^{t}\right) \quad (10)$$

Here $s1 = \exp(f_i - f_{r1})/(abs(f_i) + \in)$ and $s2 = \exp(f_{r2} - f_i)$. Rand is a uniform arbitrary number [0, 1]. $r1 \in [1,2,...X]$ is an index of the rooster, while $r2 \in [1,2,...X]$ is an index of the randomly selected hen. $r1 \neq r2$.

The location of the chicks are modeled around the hens as in the following equation

$$x_{i,j}^{t+1} = x_{i,j}^{t} + FL * \left(x_{m,j}^{t} - x_{i,j}^{t}\right) \quad (11)$$

Where $x_{m,j}^{t}$ is the location of the i-th younger chicken's mother $(m \in [1, X])$. $FL(FL \in (0,2))$ denotes a uniform random distribution parameter. It is set between 0 and 2 to progress the variety of the population and obtain better convergence rate.

C. ICSO Algorithm

As the CSO has limitations in convergence speed and local optimum problems, especially in training the RNNs, the improved CSO is developed. In the ICSO algorithm, the convergence of the algorithm is improved along with the speed through the modification of the location update equations of the hens. As the chicks are completely dependent on the hens, their position update equation is also modified are modified. After initialization of the chicken swarm population randomly, the rooster positions are updated. Then the location of each hen is updated using new equations that include the adaptive weight factors. The location of the hens is updated by the weight values so that the global and local search ability is significantly improved.

$$x_{i,j}^{t+1} = w(t).x_{i,j}^{t} + s1 * Rand * \left(x_{r1,j}^{t} - x_{i,j}^{t}\right) + S2 * Rand * \left(x_{r2,j}^{t} - x_{i,j}^{t}\right) \quad (12)$$

$$w(t) = w_{\max}(w_{\max} - w_{\min}).\frac{(t_{\max} - t)}{t_{\max}} \quad (13)$$

Where $w_{\max}$ and $w_{\min}$ are the extreme and least weights. $t_{\max}$ denote the maximum iterations and indicate current iteration number. The weight values are adaptively determined using the above equation such that w(t), $w_{\max}, w_{\min} \in [0,1]$. Through extensive simulation, the best results are obtained when the values of $w_{\max}$ and $w_{\min}$ are determined as 0.9 and 0.4, respectively.

The locations of the small chicks are updated by

$$x_{i,j}^{t+1} = x_{i,j}^{t} + C * \left(x_{m,j}^{t} - x_{i,j}^{t}\right) \quad (14)$$

Here C replaces the uniform random distribution parameter FL in order to increase the global and local exploration capability of the chicks. C is also a coefficient but its value is calculated as in the following equation instead of setting its value directly.

$$C = 0.1 \times Rand + 0.4(w(t)) \quad (15)$$

These modifications made to the CSO system increases the convergence speed of the algorithm and enhances the possibility of obtaining global optimum solution. The best solution is achieved at the end of maximum iterations with less time and less computation complexity for training the RNN.

D. Designing ICSO-RNN based IDS model

The proposed ICSO-RNN for identifying the anomalies is characterized by optimizing the structure and the parameters of the improved Elman RNN using the interleaved ICSO algorithm. For enabling this design, the ICSO formulates the structure and parameter function for evaluating the fitness of the model. Initially, the ICSO explores for the quantity of hidden layer nodes and then optimizes the parameters namely weight values, inputs to context layer neurons and the value of for each structure (chicken) in the ICSO. At maximum iterations, the best solution is obtained which includes the optimal arrangement and constraints for the RNN.

Encoding scheme: The ICSO optimization algorithm determines the optimal structure and values for the parameters of RNN in a simultaneous process. For achieving this outcome, two kinds of chicken swarms are initialized: each one for structure and the parameters. The structure chicken swarm is encoded as a binary string i.e. the value can be either zero when the hidden nodes are absent or one when the hidden nodes are present. The length of this binary string is determined by limited specified during runtime. The parameter chicken swarm is determined as a real integer coded vector. As the Elman RNN consists of l input level nodes, m hidden level nodes and context level nodes, and n output level nodes, the parameter chicken represent the weights, self-feedback constant and the preliminary inputs of context nodes. Hence the sum of elements in the parameter chicken is given by $m + m + ml + mn$ .This process simplifies the optimal structure and parameter generation process by ICSO.

Fitness evaluation: For each chicken in the swarm, the fitness must be estimated according to fitness function.

$$MSE = \frac{1}{n,N}\sum_{t=1}^{N}\sum_{i=1}^{n}\left(y_{ti(k)} - \underset{tt}{\tilde{y}(k)}\right)^2 \quad (16)$$

In the ICSO-RNN, the fitness function evaluation is performed with the help of mean squared error (MSE) equation to estimate the mean output error.

Here N represent the training samples, n denote the network outputs, $y_{ti(k)}$ denote the expected outcome and $\underset{tt}{\tilde{y}(k)}$ denote the actual outcome of the network of the t samples at iteration k. The appropriate objective function for the proposed ICSO-RNN is estimated using the following equation by considering the system's magnitude and convergence rate.

$$fitness = MSE + \beta(m / \max_{-m}) \quad (17)$$

Here β denote a control constant for the system size, m denote the amount of hidden nodes. Based on this objective function, the ability of the ICSO-RNN is evaluated and the best solution is obtained. The main constraints that the ICSO considers are that the consideration of least hidden nodes for minimizing the complexity and improve the network performance.

Implementation: The proposed ICSO-RNN is implemented by initializing the structure population. For each structure, the parameter chicken is initialized and the fitness values are computed to sort the best parameter values. Then the validation is performed to

obtain fitness values for each network and evaluate the optimal structure.

Algorithm: ICSO-RNN

i. Begin
ii. Initialize the RNN and CSO parameters
iii. Initialize the structure chicken population, number of hidden nodes
iv. Iteration = 0;
v. For each structure
a. Initialize the parameter chicken population
b. Generate initial parameter chickens randomly
c. Estimate the fitness of each parameter chicken
d. Sort the chickens in ascending order
e. Return best parameter chicken
f. Until maximum iterations
vi. End for
vii. Iteration = iteration +1;
viii. Estimate fitness for each network by validation
ix. Represent number of hidden nodes as structure chicken
x. For each structure chicken
a. Add hidden nodes operator
b. Estimate the computation time
c. Return optimal structure chicken
d. If new solution = best;
i. Terminate the algorithm
e. Else go to step v
xi. End if
xii. End for
xiii. End

IV. RESULTS AND DISCUSSION

A. Simulation environment and datasets

The proposed ICSO-RNN model is simulated using MATLAB 2016b (version 9.1) installed on 64-bit Windows system with Intel core i3 processor, 4GB RAM and hard drive of 500GB. The simulation is performed using the UNSW-NB15 dataset. UNSW-NB 15 dataset was created by the IXIA PerfectStorm tool for creating a hybrid of real fresh normal events and synthetic modern attack patterns.It contains49 features and nine types of attacks viz.Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. A total of 2,540,044 records are present in the consolidated dataset from which175,341 records forms the training set while the remaining 82,332 records form the testing set.

B. Performance evaluation

The performance of the proposed ICSO-RNN is evaluated and compared with the prevailing Naïve Bayes [7], SVM [8] and Random forests [9] based IDS models. Accuracy, precision, recall, f-measure and time are considered for the performance estimation.

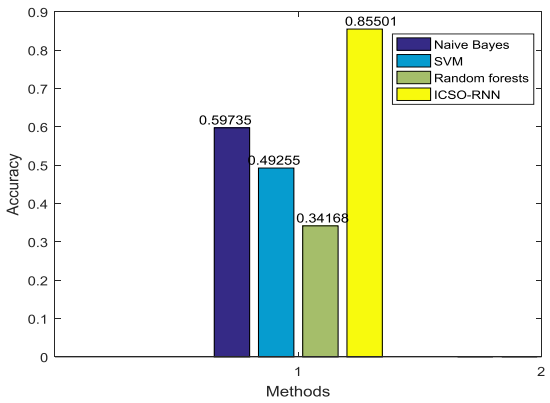

Fig.2. Accuracy comparison

Fig.2 displays the accuracy evaluation of the suggested ICSO-RNN against the prevailing IDS models. From the comparison, it is apparent that the suggested ICSO-RNN has higher accuracy than the existing models. ICSO-RNN has accuracy of 0.855 and it is 25.8% greater than Naïve Bayes, 36.3% larger than SVM and 51.4% higher than Random forests for the UNSW-NB 15 dataset.

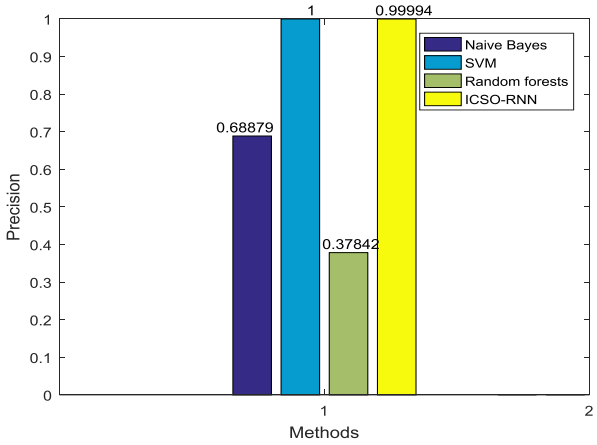

Fig.3. Precision Comparison

Fig.3 shows the precision evaluation of the suggested ICSO-RNN against the existing models. The precision results for the UNSW-NB15 dataset shows that the ICSO-RNN has high precision which is almost equivalent of SVM and greater than other models. ICSO-RNN has precision of 0.99994 which is 31% superior to Naïve Bayes, and 62.1% greater than Random forests based IDS models.

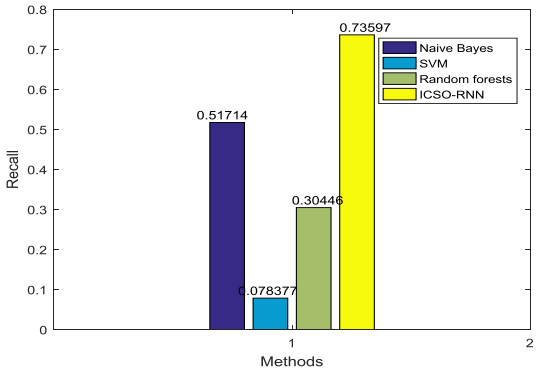

Fig.4. Recall Comparison

Fig. 4 shows the recall evaluation of the suggested ICSO-RNN against the prevailing IDS models. The results for the UNSW-NB15 dataset show that the ICSO-RNN has high recallthan other models.ICSO-RNN has recall 0.73597 which is 21.8% greater than Naïve Bayes, 65% superiorto SVM and 43% larger than Random forests based IDS models.

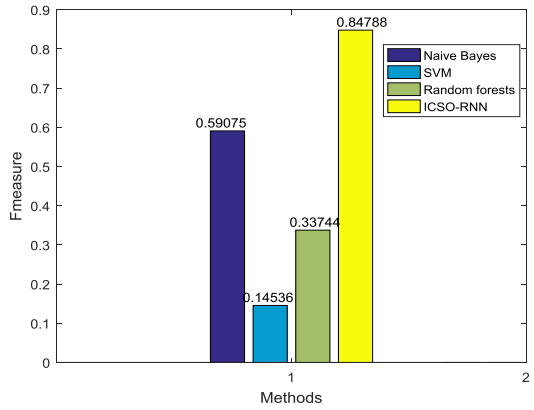

Fig.5. F-measure Comparison

Fig. 5 shows the f-measure evaluation of the suggested ICSO-RNN based IDS with the present IDS models. The plot results for the UNSW-NB15 dataset shows that the ICSO-RNN has high f-measure than other IDS models. ICSO-RNN has f-measure of 0.84788 and it is 25.7% greater than Naïve Bayes, 70.2%larger than SVM and 51% superior to Random forests based model.

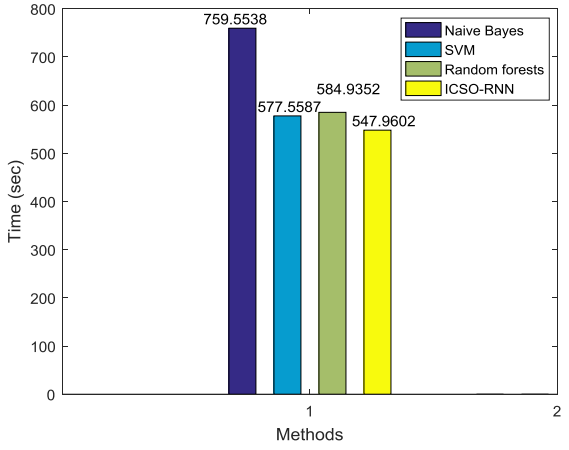

Fig.6. Processing time comparison

Fig.6 illustrates the processing time evaluation for the suggested IDS model against the prevailing models. The main evaluation of this research study is to decrease the time complexity. From the figure, it is proved that the time complexity of the suggested deep learning based IDS is very less. ICSO-RNN consumes 547.96 seconds for the intrusion detection from UNSW-NB15 dataset which is significantly less than the other compared IDS models. The main reason for this enhancement is the use of superior design approach of RNN with proficient training by ICSO.

V. CONCLUSION

Advanced IDS had been developed in this article by employing the optimized deep learning model of ICSO-RNN. First, the data is normalized and the features are selected using Intuitionistic Fuzzy Mutual Information. Then the ICSO-RNN is utilized to identify the accurate classes of the system traffic. The RNN is trained using the ICSO which optimizes the structure and parameters of the improved Elman RNN. This proposed IDS model improved the training process using RNN and increased the accuracy of intrusion detection with less time complexity. In future, the proposed ICSO-RNN can be applied to larger datasets of cyber security problems other than UNSW-NB15 and the efficiency will be investigated. In addition, the possibility of including parallel processing strategy to further reduce the training time will also be examined in future.

REFERENCES

1. H. J. Liao, C. H. R. Lin, Y. C. Lin and K. Y. Tung, “Intrusion detection system: A comprehensive review,” Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16-24, 2013.
2. S. X.WuandW. Banzhaf, “The use of computational intelligence in intrusion detection systems: A review,” Applied soft computing, vol. 10, no. 1, pp. 1-35, 2010.
3. R. Zuech, T. M. Khoshgoftaar andR. Wald, “Intrusion detection and big heterogeneous data: a survey,” Journal of Big Data, vol. 2, no. 1, p. 3, 2015.
4. S. Suthaharan, “Big data classification: Problems and challenges in network intrusion prediction with machine learning,” ACM SIGMETRICS Performance Evaluation Review, vol. 41, no. 4, pp. 70-73, 2014.
5. A. L. BuczakandE. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” IEEE Communications surveys and tutorials, vol. 18, no. 2, pp. 1153-1176, 2015.
6. S. S. S.Sindhu, S. Geetha andA. Kannan, “Decision tree based light weight intrusion detection using a wrapper approach,” Expert Systems with applications, vol. 39, no. 1, pp. 129-141, 2012.
7. S. Mukherjee andN. Sharma, “Intrusion detection using naive Bayes classifier with feature reduction,”Procedia Technology, vol. 4, pp. 119-128, 2012.
8. Z. ZhangandH. Shen, “Application of online-training SVMs for real-time intrusion detection with different considerations,” Computer Communications, vol. 28, no. 12, pp. 1428-1442, 2005.
9. N. Farnaaz andM. A. Jabbar, “Random forest modeling for network intrusion detection system,”Procedia Computer Science, vol. 89, no. 1, pp. 213-217, 2016.
10. C. Cheng, W. P. Tay andG. B. Huang, "Extreme learning machines for intrusion detection,” In 2012 International joint conference on neural networks (IJCNN), IEEE, pp. 1-8, 2012.
11. B. Dong andX. Wang, “Comparison deep learning method to traditional methods using for network intrusion detection,” In 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN), IEEE, pp. 581-585, 2016.
12. R. Vinayakumar, M.Alazab, K. P.Soman, P.Poornachandran, A. Al-NemratandS. Venkatraman, “Deep learning approach for intelligent intrusion detection system,” IEEE Access, vol. 7, pp. 41525-41550, 2019.
13. L. O.Anyanwu, J.Keengwe andG. A. Arome, “Scalable intrusion detection with recurrent neural networks,” In 2010 Seventh International Conference on Information Technology: New Generations, IEEE, pp. 919-923, 2010.
14. P. Sudha andR. Gunavathi, “Knowledgeable Handling of Impreciseness in Feature Subset Selection using Intuitionistic Fuzzy Mutual Information of Intrusion Detection. System,” International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 8, no. 12, pp. 1539-1544, 2019.
15. C. Yin, Y. Zhu, J.Fei andX. He, “A deep learning approach for intrusion detection using recurrent neural networks,” IEEE Access, vol. 5, pp. 21954-21961, 2017.
16. N. Shone, T. N. Ngoc, V. D. Phai andQ. Shi, “A deep learning approach to network intrusion detection,” IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41-50, 2018.
17. M. Al-Qatf, Y.Lasheng, M. Al-HabibandK. Al-Sabahi, “Deep learning approach combining sparse autoencoder with SVM for network intrusion detection,” IEEE Access, vol. 6, pp. 52843-52856, 2018.

18. K. Wu, Z. Chen and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," IEEE Access, vol. 6, pp. 50850-50859, 2018.
19. R. Abdulhammed, M.Faezipour, A.Abuzneid andA. AbuMallouh, "Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic," IEEE sensors letters, vol. 3, no. 1, pp. 1-4, 2018.
20. N. Marir, H. Wang, G. Feng, B. Li andM. Jia, "Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark. IEEE Access, vol. 6, pp. 59657-59671, 2018.
21. F. A. Khan, A. Gumaei, A.DerhabandA. Hussain, "A novel two-stage deep learning model for efficient network intrusion detection," IEEE Access, vol. 7, pp. 30373-30385, 2019.
22. K. Al Jallad, M. AljnidiandM. S. Desouki, "Big data analysis and distributed deep learning for next-generation intrusion detection system optimization," Journal of Big Data, vol. 6, no. 1, p. 88, 2019.
23. S. Garg, K. Kaur, N. Kumar, G.Kaddoum, A. Y. ZomayaandR. Ranjan, "A hybrid deep learning-based model for anomaly detection in cloud datacenter networks," IEEE Transactions on Network and Service Management, vol. 16, no. 3, pp. 924-935, 2019.
24. M. M. Hassan, A. Gumaei, A.Alsanad, M.AlrubaianandG. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," Information Sciences, vol. 513, pp. 386-396, 2020.
25. W. M. Lin andC. M. Hong, "A new Elman neural network-based control algorithm for adjustable-pitch variable-speed wind-energy conversion systems," IEEE transactions on power electronics, 26(2), 473-481, 2010.
26. X. Meng, Y. Liu, X. GaoandH. Zhang, "A new bio-inspired algorithm: chicken swarm optimization," In International conference in swarm intelligence, Springer, Cham, pp. 86-94, 2014

AUTHORS PROFILE

Mrs. P.Sudha has completed her M.Phil. in Computer Science and pursuing Ph.D in Bharathiar University. Her research area is Data Mining and Big Data Analytics. She has 13 years teaching experience. She is currently working as Assistant Professor in Computer Science, Sree Saraswathi Thyagaraja College, Pollachi. She has 10 years of research experience. She has published around 20 research articles in the refereed International Journals with high impact factor and also presented many research papers in the National and International level Conference.

Dr. R. Gunavathi has completed her Ph.D. in Computer Science in Mother Teresa Women's University, Kodaikanel, and her research is on "Efficient Cluster head selection algorithms to improve the Quality of service in Mobile Ad hoc networks". She has 20 years of teaching experience and currently working as Associate Professor and Head, Department of MCA at Sree Sarawthi Thyagaraja College, Pollachi. She has 15 years of research experience. Her current research interest in Mobile Ad hoc Networks, Vehicular Ad hoc Networks and Big Data Analytics. She has published around 30 research articles in the refereed International Journals with good impact factor and also presented 25 research papers in the National and International level Conferences. She has organized many National level seminars, workshops and conferences