A Hybrid Convolutional Neural Network and Recurrent Neural Network based Classification Method for Cyber Threat Detection Analysis

T.Elangovan[*]

[*]Ph.D. Research Scholar, Dept. of Computer Science Erode Arts and Science College, Erode-638 009, Tamilnadu, India. E-mail: elangovaneasc@gmail.com

Dr.S.Sukumaran

Associate Professor in Computer Science, Erode Arts and Science College, Erode-638 009, Tamilnadu, India. E-mail: prof_sukumar@yahoo.co.in

Abstract

The classification of cyber threat detection is found to be one of the emerging areas in computer vision. The main goal is to improve the accuracy of the classification of cyber threat detection. The principle of conventional IDS is to detect attempts to attack a network and to identify abnormal activities and behaviors. The reasons, including the uncertainty in searching for types of attacks and the increasing complexity of advanced cyber-attacks, IDS calls for the need for integration of methods such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) more precisely Long Short Term Memory (LSTM). In this work, CNN and RNN are employed for the development of discriminative characteristics and sequential-labels respectively. The CNN and RNN learn input traffic data in sync with a gradually increasing granularity such that both spatial and temporal features of the data can be effectively extracted. A convolutional recurrent neural network is used to create a deep learning based hybrid ID framework that predicts and classifies malicious cyber-attacks in the network. To assess the efficacy of the HCRNN proposed method, experiments were done on publicly available Intrusion Detection data, specifically the NSLKDD dataset. The experiments prove that the proposed HCRNN substantially outperforms current ID methodologies attaining high malicious attack detection rate accuracy.

Keywords: Network Intrusion Detection, Convolutional Neural Network, Recurrent Neural Network, Long Short Term Memory, HCRNN, Cyber Threats.

1. INTRODUCTION

Cyber-attacks are becoming increasingly common, which is a significant security concern [1,2]. For this reason, technicians and network security specialists are paying increasing attention to identifying network attacks. Networked computing becomes indispensable to people's life. From daily communications to commercial transactions, from small businesses to large enterprises, all activities can be or will soon be done through networked services [3]. Any vulnerabilities in the networked devices and computing platforms can expose the whole network under various attacks and may bring about disastrous consequences. Hence, effective network intrusion detection (NID) solutions are ultimately essential to the modern society. Initial NID designs are signature-based, where each type of attacks should be manually studied beforehand, and the detection is performed based on the attack's signatures. This kind of approaches are, however, not suitable to the fast growing network and cannot

cope with attacks of increasing volume, complexity and volatility [5]. For the security of such a large scale and ever-expanding network, we need an intrusion detection system that is not only able to quickly and correctly identify known attacks but also adaptive and intelligent enough for the unknown and evolved attacks, which leads to AI-based solutions [6].

The artificial intelligence gained from machine learning enables the detection system to discover network attacks without much need for human intervention [7]. So far, investigations on the AI-based solutions are mainly based on two schemes: anomaly detection and misuse detection. The anomaly detection identifies an attack based on its anomalies deviated from the profile of normal traffic. Nevertheless, this scheme may have a high false positive rate if the normal traffic is not well profiled, and the profile used in the detection is not fully representative [8]. Furthermore, to obtain a fully representative normal traffic profile for a dynamically evolving and expanding network is unlikely possible. The misuse detection, on the other hand, focuses on the abnormal behaviour directly. The scheme can learn features of attacks based on a labelled training dataset, where both normal and attacked traffic data are marked. Given sufficient labelled data, a misuse-detection design can effectively generate an abstract hypothesis of the boundary between normal and malicious traffic [9].

The hypothesis can then be used to detect attacks for future unknown traffic. Therefore the misuse detection is more feasible and effective than the anomaly detection and has been adopted in real-world systems [10]. The existing misuse detection designs still present a high false positive rate, which significantly limits the in-time detection efficiency, incurs large manual scrutiny workload, and potentially degrades the network-wide security [11]. This paper addresses this issue, and aim for an improved misuse detection design.

2. RELATED WORKS

Yin et al. [21] proposed a model for intrusion detection using Recurrent Neural Networks (RNNs). RNNs are especially suited to data that are time dependent. This model contained of forward and back propagation stages. Forward propagation calculates the output values, although back propagation passes residuals accumulated to update the weights. The model contained of 20 hidden nodes, with Sigmoid as the activation function and Softmax as the classification function. Performance results using the NSL-KDD data set presented the accuracy values was 83.28% and 81.29% for binary and multiclass classification.

Deep learning and traditional machine learning techniques can be hybridized to increase intrusion detection accuracy. A combination of sparse auto encoder and SVM was proposed by Al-Qatf et al [1]. The sparse auto encoder used to capture the input training data set, whereas the SVM used to build the classification model. This model trained and evaluated using the NSL-KDD data set. The obtained accuracy values were 84.96% and 80.48% for two-class and five-class classification, respectively.

Altwaijry et al. [2] developed an intrusion detection model using DNN. The proposed model consisted of four hidden fully connected layers and trained using NSL-KDD data set. The DNN model acquired accuracy values of 84.70% and 77.55% for the two class and five-class classification difficulties. The proposed model outperformed traditional machine learning algorithms, including NB, Bagging, and Adaboost in terms of accuracy and recall.

F. Farahnakian et al. [6] proposed to use Deep Auto-Encoder (DAE) as one of the most well-known deep learning models. The DAE model is formed in an avid layer way to avoid overflow and local optimum. The experimental results of the KDD-CUP 99 dataset show that our approach makes for substantial improvements over other approaches based on in-depth learning in precision, detection rates and false alarm rates.

S. Seo et al. [14] defined RBM as a type of unsupervised learning that does not use class labels. RBM is a probabilistic generative model that composes new input data based on formed probability. The new data compiled by RBM shows that noise and outliers are removed from the input data. When newly composed data is applied to the network intrusion detection model, the negative effects of noise and outliers on learning are eliminated. They offer noise and outlier values in KDD Cup99 Data are removed by applying the data to RBM and composing a new data. Then use the results between the existing data and the data from which the noises and outliers are removed.

R. Vinayakumar et al. [18] describes sequential data modelling is a relevant cyber security task. In addition, Stacked Recurring Neural Networks (S-RNN) have the potential to quickly learn complex temporal behaviors, including sparse representations. To do this, the network traffic as a time series, especially Transmission Control Protocol/Internet Protocol (TCP/IP) packets in a predefined time range with a supervised learning method, using millions of known good and bad network connections. To discover the best architecture, the authors complete a comprehensive review of various RNN architectures with its network parameters and network structures. They use the login records of the Kddcup-99 challenge dataset.

Kasun et al. [3] proposed a methodology to generate offline and online feedback to the user on the DNN-IDS decision-making process. Offline, the user is reported the input features that are most relevant to detect each type of intrusion by the trained DNN-IDS. Online, for each detection, the user is reported the input features that contributed the most to the detection. This can be binomial where the data indicates the presence or absence of an attack, where it can be multinomial where the input record can be from a specific attack group.

3. EXISTING METHOD

Support Vector Machine (SVM)

The Support Vector Machine (SVM) is typically used to describe classification with support vector methods and support vector regression is used to describe regression with support vector methods [4]. SVM is a useful technique for data classification. SVM is an innovative approach to constructing learning machines that minimize generalization error. The classification problem can be restricted to consideration of the two-class problem without loss of generality. In this problem the goal is to separate the two classes by a function which is induced from available examples. The goal is to produce a classifier that will work well on unseen examples, i.e. it generalizes well. Here there are many possible linear classifiers that can separate the data, but there is only one that maximizes the margin (maximizes the distance between it and the nearest data point of each class). These linear classifiers termed the optimal separating hyper plane. The system in SVMs is to discover a maximum edge partition hyperplane in the n-measurement highlight space. SVMs can accomplish satisfying outcomes even with limited scope preparing sets in light of the fact that the partition hyperplane is resolved simply by few help vectors. In any case, SVMs are delicate to commotion close the hyperplane.

4. PROPOSED METHODOLOGY

Convolutional Neural Network (CNN)

Convolutional Neural Network **(**CNN) main operations are convolution and pooling. Convolution transforms input data, through a set of filters or kernels, to an output that highlights the features of the input data, hence the output is usually called feature map [19]. The convolution output is further processed by an activation function and then down-sampled by pooling to trim off irrelevant data. Pooling also helps to remove glitches in the data to improve the learning for the following layers.

CNN learns the input data by adjusting the filters automatically through rounds and rounds of learning processes so that its output feature map can effectively represent the raw input data.

Since the network packet is presented in a 1D format, then use 1D convolution, as illustrated in equation (1) that specifies the operation to the input vector g with a filter f of size m.

$$(f * g)(i) = \sum_{j=1}^{m} g(j) \cdot f(i - j + m/2) \quad \ldots (1)$$

Where i is the position of different values in the sequence data.

Because the rectified linear unit (ReLU): f(z) = max(0, z) is good for fast learning convergence, therefore, choose it as the activation function. Also use the max pooling operation, as commonly applied in other existing designs [20].

Batch Normalization

One problem with using the deep neural network is that the input value range dynamical changes from layer to layer during training, which is also known as covariance shift. The covariance shift causes the learning efficiency of one layer dependent on other layers, making the learning outcome unstable. Furthermore, because of the covariance shift, the learning rate is likely restricted to a low value to ensure data in different input ranges to be effectively learned, which slows down the learning speed. Batch normalization can be used to address this issue.

Normalization scales data to the unit norm in the input layer and has been used to accelerate training deep neural network for image recognition.

Batch normalization is used to adjust the CNN output for RNN. The normalization subtracts the batch mean from each data and divides the result by the batch standard deviation, as given in equation (2).

$$\hat{x} = \frac{x - \mu_B}{\sqrt{\delta_B^2 + \epsilon}} \quad \ldots (2)$$

where x is a value in the input batch, and μ_B and δ_B are, respectively, the batch mean and variance. The ϵ is an ignorable value, just to ensure the denominator in the formula non-zero.

Based on the normalized $\hat{x}$, the normalization produces the output y as given in equation (3), where the γ and β will be trained in the learning process for a better learning outcome.

$$\hat{y} = \gamma\hat{x} + \beta \quad \ldots (3)$$

Recurrent Neural Network (RNN)

Recurrent neural network (RNN) is an enhanced version of feed-forward neural networks that can memorize data at each step for subsequent outputs. In an RNN, the output of neurons is connected to the input of other neurons and themselves. Therefore, RNNs can model data sequences and time series using their internal memory [5]. In Intrusion Detection Systems, RNNs are generally employed to extract temporal correlations between security attacks and malicious behaviors (temporal features), whereas CNNs extract spatial features. RNNs are implemented in different architectures; among them, the Long Short-Term Memory (LSTM) model is well known and widely used in IDSs.

RNN is widely used for learning from sequential training data. It trains the model by using the back-propagation methodology. Recurrent networks are different than the Multi-Layer Perceptron (MLP) since they not only consider the current input but also take into account what has happened previously. The three layers of RNN with dropout followed by the output layer using a fully connected layer.

Long Short Term Memory (LSTM)

Different from CNN that learns information on an individual data record basis, RNN can establish the relationship between data records by feeding back what has been learned from the previous learning to the current learning, and hence can capture the temporal features in the input data [21].

However, the simple feedback used in the traditional RNN may have a learning error accumulated in the long dependency. The accumulated errors may become large enough to invalid the final learning outcome. Long Short-Term Memory (LSTM), a gated recurrent neural network, mitigates such a problem. It controls the feedback with a set of gate functions such that the short lived errors are eventually dropped out and only persistent features are retained. Therefore, we use LSTM for RNN.

LSTM can be abstracted as a connection of four sub networks (denoted as p-net, g-net, f-net and q-net in the diagram), a set of control gates, and a memory component. The input and output values in the diagram are vectors of the same size determined by the input x(t). The state, s(t), saved in the memory, serves as the feedback to the current learning. All the sub nets in LSTM have a similar structure, as specified in equation (4).

$$b + U \times x(t) + W \times h(t-1) \quad \ldots(4)$$

where x(t), h(t − 1), b, U and W are, respectively, current input, previous output, bias, weight matrix for the current input, and recurrent weight matrix for the previous output. Each of the four nets has a different b, U, and W.

The outputs from the sub nets (p(t), g(t), f(t) and q(t)) are then used, through two types of controlling gates (σ and tanh) to determine the feedback s(t) from the previous learning and the current output h(t), as given in equations (5) and (6), respectively.

$$s(t) = \sigma(f(t)) * s(t-1) + \sigma(p(t)) * \tanh g(t) \quad \ldots(5)$$

$$h(t) = \tanh s(t) * \sigma(q(t)) \quad \ldots(6)$$

LSTM learns the inputs by adjusting the weights in those nets and the σ value such that the temporal features between the input data can be effectively generated in the output.

Dimension Reshape

The learning granularity changes from one HCRNN level to another, the output size of one level is different from the input size expected at the next level. Therefore, add a layer to reshape the data for the next block.

Overfitting Prevention

One typical problem when learning big data using a deep neural network is over fitting namely, the network has learned the training data too well, which restricts its ability to identify variants in new samples. This problem can be handled by Dropout. Dropout randomly removes some connections from the deep neural network to reduce overfitting.

Final Layers

Finally, an extra convolution layer and a global average pooling layer are used to extract further spatial temporal features learned the blocks. The final learning output is generated by the last layer, a fully-connected layer.

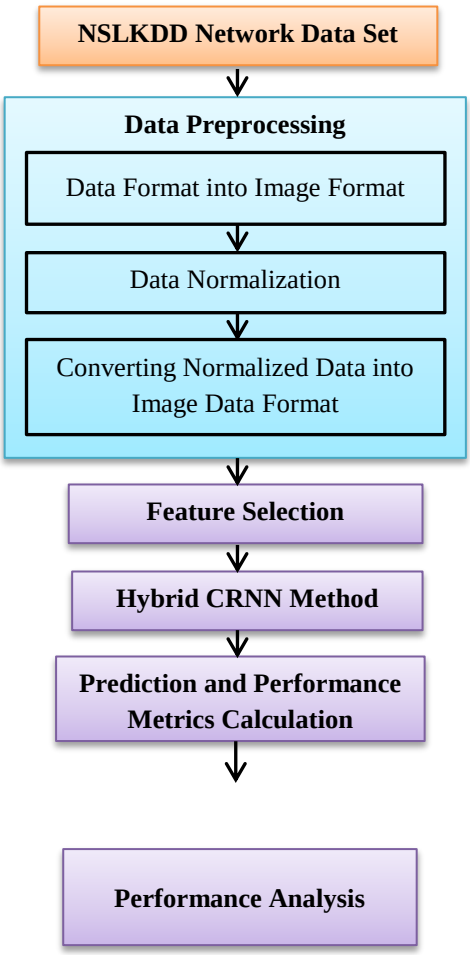

Figure 1. Block Diagram of Proposed HCRNN Method

Algorithm for HCRNN

Step 1: Start the Process
Step 2: Select the NSLKDD Network Dataset
Step 3: Perform Data Preprocessing
Step 4: Transfer symbolic features into numerical value
Step 5: Apply min-max Normalization
Step 6: Design a fully connected neural network with a sigmoid activation function.
Step 7: Input S^S and build attention weight matrix A.
Step 8: Calculate $S^A = S^f.A$
Step 9: Initialize w^k and b^k randomly.
Step 10: Add three 1D CNN layers with leaky LSTM activation.
Step 11: Add 3 LSTM layer with the number of neurons as 128, 64, and 64 respectively.
Step 12: Add one dense layer.
Step 13: For i = 1 to no-1 do
Calculate feature map
Calculate temporal features
Find out the error value
Fine tune the hyper parameters using back propagation.
End for
Step 14: Finally Classified Data

5. EXPERIMENTS AND RESULTS

The KDD dataset is constructed from the data gathered by the IDS evaluation program of DARPA'98. NSL-KDD is a new version of the KDD dataset that was proposed as a solution to the problems encountered in the KDD dataset. The dataset overcomes and removes the redundant data of the KDD dataset. It contains 125,973 and 22,544 instances for training and testing respectively [18]. The number of features is 41. The data is either labeled as normal or a particular attack type. There are 39 different attack types that are grouped into four main categories of attacks: DoS, Probing, Remote to Local (R2L), and User to Root (U2R) [19].

For the performance evaluation, four metrics are used: Accuracy, TPR, FPR, and F-Measure, which are all commonly used for learning-based methods in the field of cyber threat detection. TPR is used to estimate the systems performance with respect to its threat detection. FPR is used to evaluate misclassifications of normal data. F-measure is the harmonic mean of the precision and TPR (recall), where Precision = TP / (TP+FP) is the percentage of true attacks among all attacks classified.

The explanations for accuracy, TPR, FPR, and F-measure are offered below:

$$\text{TPR (Recall)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad \ldots (7)$$

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}} \quad \ldots (8)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \quad \ldots (9)$$

$$\text{F} - \text{measure} = 2 * \frac{\text{Precison} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad \ldots (10)$$

Table 1. Comparison Table of Proposed Work

Methods/Parameters	Accuracy	Precision	Recall	F-Measure
SVM	92.22	93.58	94.19	93.25
CNN	93.27	94.44	95.01	93.98
RNN	94.85	96.33	95.04	94.17
HCRNN	99.75	99.36	99.28	99.71

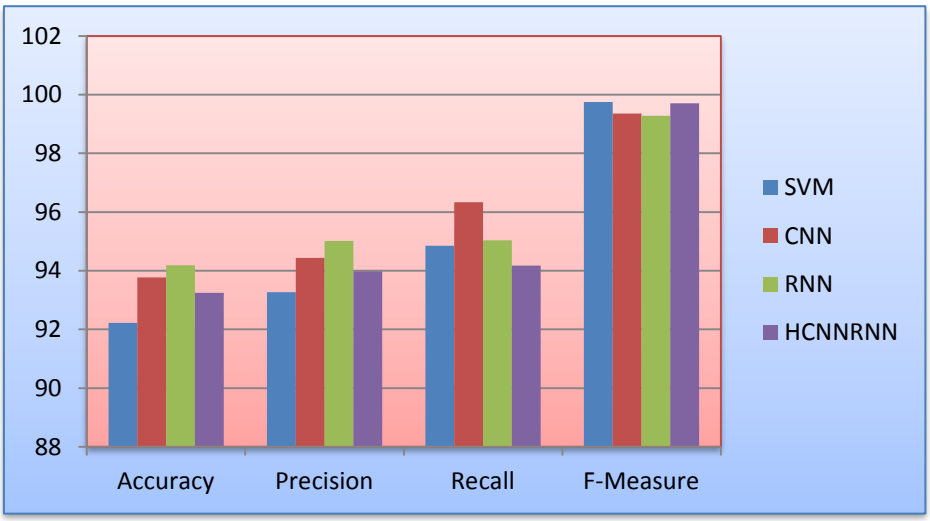

Figure 2. Comparison Graph for Proposed HCRNN

6. CONCLUSION

A new deep learning-based approach suggested in this paper for developing a cyber-threat detection method. A new approach based on learning algorithms that makes it possible to detect an attack in order not to allow the same aggression to recur. Detection allows the identification of a certain characteristic that violates security policies. This allowed us to cyber threat detection using HCRNN, which provided an instant update of a new malware sample following its introduction into the classification system. The experimental results suggest that the HCRNN is able to accurately identify the cyber threats. Apart from CNN and RNN, more complex neural network architectures will be considered as part of our future analysis. Despite the relative abundance of extant works addressing cyber threat detection, there is still space for experimentation, and the discovery of new insights on the nature of cyber threat detection may lead to more efficient and accurate models. Traditional models can also be beneficial if they are combined with task specific feature techniques.

REFERENCES

[1] Al-Qatf M, Lasheng Y, Al-Habib M and K Al-Sabahi, "Deep Learning Approach Combining Sparse Auto Encoder with SVM for Network Intrusion Detection", IEEE Access, Vol. 6, Pp. 52843–52856, 2018.

[2] Altwaijry N, Alqahtani A, and Al-Turaiki I. "A Deep Learning Approach for Anomaly-Based Network Intrusion Detection", First International Conference on Big Data and Security, Nanjing, China: Springer, 2019.

[3] Amarasinghe, K. and Manic, M., "Improving User Trust on Deep Neural Networks Based Intrusion Detection Systems", IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, Washington DC, Pp. 3262-3268, 2018.

[4] Christiana Ioannou and Vasos Vassiliou, "Network Attack Classification in IoT Using Support Vector Machines", Journal of Sensor and Actuator Networks, Vol. 10, No. 58, Pp. 1-19, 2021.

[5] Dusan Nedeljkovic and Zivana Jakovljevic, "Cyber-attack detection method based on RNN", 7th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN 2020), at Belgrade, 2020.

[6] Farahnakian, F. and Heikkonen, J., "A Deep Auto-Encoder Based Approach for Intrusion Detection System", 20th International Conference on Advanced Communication Technology (ICACT), Korea (South), PP. 247-252, 11-14 February 2018.

[7] R. H. Hwang, M. C. Peng, C. W. Huang, P. C. Lin, and V. L. Nguyen, "An unsupervised deep learning model for early network traffic anomaly detection", IEEE Access, Vol. 8, Pp. 30387–30399, 2020.

[8] Javaid A, Niyaz Q, Sun W, and Alam M, "A Deep Learning Approach for Network Intrusion Detection System", 9th EAI International Conference on Bio-inspired Information and Communications Technologies, Pp. 21–26, 2016.

[9] Justin, V., Marathe, N. and Dongre, N., "Hybrid IDS Using SVM Classifier for Detecting DoS Attack in MANET Application", International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), Palladam, 775-778, 2017.

[10] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "A Novel Two Stage Deep Learning Model for Efficient Network Intrusion Detection", IEEE Access, Vol. 7, Pp. 30373–30385, 2019.

[11] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of Intrusion Detection using Deep Neural Network," IEEE International Conference on Big Data and Smart Computing, pp. 313–316, 2017.

[12] Kumari, U. and Soni, U., "A Review of Intrusion Detection Using Anomaly Based Detection". 2nd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, 19-20 October, Pp.824-826, 2017.

[13] Park, Y.S., Choi, C.S., Jang, C., Shin, D.G., Cho, G.C. and Kim, H.S., "Development of Incident Response Tool for Cyber Security Training Based on Virtualization and Cloud", International Workshop on Big Data and Information Security (IWBIS), Indonesia, Pp. 115-118, 2019.

[14] Seo, S., Park, S. and Kim, J., "Improvement of Network Intrusion Detection Accuracy by Using Restricted Boltzmann Machine", 8th International Conference on Computational Intelligence and Communication Networks (CICN), Pp. 413-417, 23-25 December 2016.

[15] Shadi Aljawarneh, Monther Aldwairi and Muneer Bani Yassein, "Anomaly-based Intrusion Detection System through Feature Selection Analysis and Building Hybrid Efficient Model", Journal of Computational Science, Vol. 25, Pp. 152-160, 2018.

[16] S. Sheikhi, M. Kheirabadi, and A. Bazzazi, "An Effective Model for SMS Spam Detection Using Content-based Features and Averaged Neural Network," International Journal of Engineering, Vol. 33, No. 2, Pp. 221-228, 2020.

[17] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, and S. Venkatraman, "Robust Intelligent Malware Detection using Deep Learning," IEEE Access, Vol. 7, Pp. 46717-46738, 2019.

[18] Vinayakumar, R., Soman, K. and Prabaharan, P., "Evaluation of Recurrent Neural Network and Its Variants for Intrusion Detection System (IDS)", 2020.

[19] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks", IEEE Access, Vol. 7, Pp. 42210–42219, 2019.

[20] H. Yang and F. Wang, "Wireless Network Intrusion Detection based on Improved Convolutional Neural Network", IEEE Access, Vol. 7, Pp. 64366–64374, 2019.

[21] Yin C, Zhu Y, Fei J and He X, "A Deep Learning Approach for Intrusion Detection using Recurrent Neural Networks", IEEE Access, Vol. 5, Pp. 21954–21961, 2017.

Authors Profile

T.Elangovan received the Bachelor of Computer Technology degree from the Anna University, Coimbatore, TN, India, in 2010 and the Master of Computer Sciecne (M.Sc) degree from the Bharathiar University, Coimbatore, TN, India, in 2013. He also recieved the M.Phil degree from the Bharathiar University, Coimbatore, in 2015. He is pursuing Ph.D degree in computer science at Bharathiar University. His research interests include Advanced Networks.

Dr. S. Sukumaran graduated in 1985 with a degree in Science. He obtained his Master Degree in Science and M.Phil in Computer Science from the Bharathiar University. He received the Ph.D degree in Computer Science from the Bharathiar University. He has 34 years of teaching experience starting from Lecturer to Associate Professor. At present he is working as Associate Professor of Computer Science in Erode Arts and Science College, Erode, Tamilnadu. He has guided for more than 57 M.Phil research Scholars in various fields and guided 19 Ph.D Scholars. Currently he is Guiding 6 Ph.D Scholars. He is member of Board studies of various Autonomous Colleges and Universities. He published around 115 research papers in national and international journals and conferences. His current research interests include Image processing, Network Security and Data Mining.