Indian Journal of Computer Science and Technology
Volume 3, Issue3 (September-December 2024), PP: 17-19.
www.indjcst.com **ISSN No: 2583-5300**

Enhanced Cybersecurity via Deep LSTM Networks for Intrusion Detection

Sohaib Ansari[1], Rajat Kamble[2], Shishir Mishra[3], Abhishek Piperde[4]
[1,2,3] Students, Department of Computer Science Engineering JIT, Nagpur, Maharashtra, India.
[4]Assistant.Professor, Department of Computer Science Engineering JIT, Nagpur, Maharashtra, India.

To Cite this Article: *Sohaib Ansari[1], Rajat Kamble[2], Shishir Mishra[3], Abhishek Piperde[4], "Enhanced Cybersecurity via Deep LSTM Networks for Intrusion Detection", Indian Journal of Computer Science and Technology, Volume 03, Issue 03 (September-December 2024), PP: 17-19.*

Abstract: *Deep Long Short-Term Memory Recurrent Neural network (LSTM-RNN) methodology is consists of pre-processing, training and testing phases. The raw data attributes consist of numerical and non-numerical values. Non-numerical values need to be conversion of numerical values because the LSTM-RNN model requires numerical attribute values as input. The numericalization process can be done with one-hot encoding. One-hot encoding assigns unique feature values to the non-numerical features. Some numerical zed data attributes consist of large feature value and some attributes consist of minimum value. The difference between minimum feature value and maximum feature value is very large. This difference affects the original feature values. The normalization process avoids the effectiveness of the original feature values. Normalization could be done with min-max normalization.*

Key words: *Neural network, LSTM, LSTM-RNN*

I.INTRODUCTION

The normalized datais given as an input to the proposed LSTM-RNN method. The method is evaluated with seven optimizers such as adam, adamax, Stochastic Gradient Descent (SGD), Adagrad, RMSprop, Nadam, and Adadelta. The goal of this chapter will be the following; 1) To propose a Deep Long Short Term Memory Recurrent Neural Network for intrusion detection system; 2) To test it on a real dataset NSL-KDD dataset; (3) To provide obtained results and (4) To provide evidence of why this procedure can outperform the existing classification techniques. The experiment evaluation consists of the following phases.

- Selected datasets are pre-processed using the techniques numericalization and normalization. The numericalization is used to transform categorical features into numerical features. Normalization can be normalized the large feature set values between 0 and 1.
- LSTM-RNN classifier trains the dataset with specified optimizers individually.
- The classifier tests the test dataset with specified optimizers individually.
- The classifiers results are evaluated and compared with existing classifiers results.

II.RECURRENT NEURAL NETWORK (RNN) ARCHITECTURE

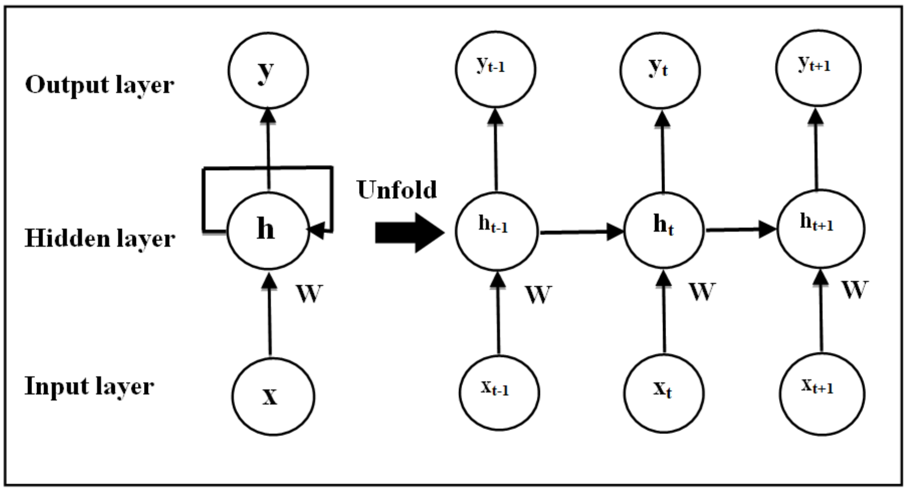

Figure 1 Weight sharing across time steps is encapsulated in the recurrent

RNN is a variant of artificial neural network (ANN), wherein the connection between the nodes resembles the neurons of human brain. Neural network connections can transmit signal to other nodes like synapses in a biological brain. The artificial neuron then processes the received signal and transmits it to other connected nodes. The connections and neurons typically have weights to adjust the learning process. The weights can vary to adjust the strength of the signals as the signal travels from the input layers to the output layers. An ANN contains hidden layers between the input and output layers. RNN should have at least three hidden layers. The basic architecture of RNNs contains input unit, hidden units, and output units evaluating all the calculations by weight adjustment to produce the outputs. It has a one-way data flow from the input units to the hidden units and a directional loop that compares the error of this hidden layer to that of the previous hidden layer, and adjusts the weights between the hidden layers. The following figure depicts a simple RNN architecture.

An RNN is an extension of traditional feed-forward neural network (FFNNs). The information moves the forward direction i.e. from the input nodes through the hidden nodes to the output nodes. There are no cycles in the network and hidden nodes are optional in FFNNs. Conventional RNN contains of input layer, and a recurrent layer as shown in above figure. They comprises of series of weight matrices and activation functions. We assume an input vector sequence, a hidden vector sequence, and an output vector sequence denoted by X, H, and Y respectively. An input vector sequence given as $X = (x1, x2, ... , xT$). Atraditional RNN calculates the hidden vector sequence $H = (h1, h2, ... , hT$)and output vector sequence $Y = (y1, y2, ... , yT$) with $t = 1$ to T as follows

$ht = (Wxhxt + Whhht{-}1 + bh)$……………..1

$yt = Whyht + by$………………………2

Where σ is the logistic function, b is bias vector, and W is weight matrix and b is bias term. In equation 3.1, *ht* is the hidden layer output at *t*-time steps, and *ht-1* denotes the previous hidden layer's output. Standard RNNs are not able to establish more than 5-10 time steps. A vanishing gradient problem arises in RNN when gradient-based learning methods are used for updating the weights. Weights receive an updated proportion of the partial derivative of the error function in each training iteration. In some cases, the gradient will be very small. These error signals may either blow-up or vanish, which prevents the weight from changing value. These vanishing error signals may cause the weights to fluctuate. The model learning takes unacceptable time with vanishing errors or does not work at all.

RNNs can be used for supervised classification learning [89, 90]. Due to vanishing and gradient, exploding RNNs are difficult to train and improperly assigns weights (i.e. assigned very high or very low values). To overcome the training issues LSTM with forget gates are often combined with RNN.

III.LSTM OPTIMIZERS

Gradient descent is one of the most popular methods to perform optimization and the most common way to optimize neural networks. Gradient descent is a way to minimize a cost function *J(θ)* by updating the parameters in the reverse direction of the gradient of the cost function $\nabla\theta$ *J(θ)* with respect to the parameters [94]. There are three types of gradient descent which differ in how much data we use to compute the gradient of the object function. They are batch gradient descent, stochastic gradient descent and mini-batch gradient descent. Vanilla mini-batch gradient descent does not guarantee good convergence, however, offers two challenges that need to be addressed two issues. The first is selecting a proper learning can be difficult. Learning rate schedules try to adjust the learning rate during training. The second is to minimize non-convex error functions for neural networks. This section summarizes some gradient descent optimizes to understand how to adjust the learning rate briefly.

IV. DATASET DESCRIPTION AND DETAILS

DARPA initiative IDS-events at MIT Lincoln LAB in 1998. Later from DARPA network dataset files, KDD99 dataset was created by Lee and Stolfo , who were participants in DARPA team. The KDD99 can be easily used in machine learning dataset. However it is much more used in IDS than DARPA dataset. The KDD dataset 38 attacks are divided into five main categories, such as Dos (Denial of Services), Probe, R2L (Root to Local), U2R (User to Root), and normal. The training and testing dataset contains 24, and 14 attacks respectively. The researchers identified several short comings in KDD dataset.

- It is heavily imbalanced i.e 80
- U2R and R2L are rare in dataset
- Redundant datasets in both train and test
- It has large dataset, most of the studies used small percentage of it.

V. DATASET FEATURES

To reduce the inadequacy of KDD99 dataset, Tavallaee et al. proposed NSL-KDD dataset. It has been generated by removing redundant instances and decreasing of size dataset. The NSL-KDD dataset had 41 features, which are either continuous or discrete. features are grouped into three categories, such as basic features Content features , and traffic features Basic features encapsulates all the attributes extracted from TCP/IP connection. Content features comprices of suspicious behavior data, i.e number of failed login attempts. Traffic category is categorized as same host with same service in current connection with respect to window interval.The NSL KDD dataset has sufficient number of records in the train and test dataset, which is reasonably rational and enables to execute experiments on the complete set. The number of selected records from each group is inversely proportional to the percentage of records in the original KDD dataset. The removal of redundant records enable the classifiers to produce an un-biased results.

VI.LSTM CLASSIFICATION ALGORITHM

```
Input
Train and test intrusion dataset x
Output
Multi-attack classification
Step 1: Normalize the dataset (Di) into values from 0 to 1
Step 2: Setup input units, LSTM units, output units to define LSTM
Step 3: Select training batch size and organize (Di) accordingly
Step 4: for n epochs and batch size do
Step 5: Train the network LSTM
Step 6: end for
Step 7: Run predictions using LSTM
Step 8: Classify the multi-attacks with soft-max activation function
Step 9: Stop.
```

VII.EXPERIMENTS AND RESULTS

NSL-KDD dataset is used to evaluate the performance of deep learning approach Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) which has been discussed in this chapter. At the pre-processing phase, the dataset features are encoded using One-Hot-Encoding to transform categorical features into numerical feature. The dataset features have been scaled using Min-Max normalization technique.

VIII. CONCLUSION

Deep neural network architecture, Long Short Term Memory Recurrent Neural Network (LSTM-RNN) to improve the accuracy and detection rate, and minimize the false alarm rate. The LSTM-RNN model is implemented with seven optimizers such as adadelta, adagrad, adamax, nadam, SGD, RMSprop, and adam optimizers individually with 100 hidden layer sizes. Our experiment is evaluated on NSL-KDD dataset. Compared with the other state-of –the-art techniques, LSTM-RNN model with adamax optimizer can significantly improve the validation accuracy, detection rate and minimized false positive rate for network intrusion detection. It must be pointed out that LSTM-RNN classifier does not accurately classify the minor number of attacks in training dataset such as R2L and U2R. The model train and test is comparatively very high.

Reference

1. *Malik, N.; Sardaraz, M.; Tahir, M.; Shah, B.; Ali, G.; Moreira, F. Energy-efficient load balancing algorithm for workflow scheduling in cloud data centers using queuing and thresholds. Appl. Sci.* ***2021****, 11, 5849.*
2. *Baiyere, A.; Topi, H.; Venkatesh, V.; Wyatt, J.; Design, R.; Donnellan, B. Communications of the Association for Information Systems Internet of Things (IoT)—A Research Agenda for Information Systems.*
3. *Lone, A.N.; Mustajab, S.; Alam, M. A comprehensive study on cybersecurity challenges and opportunities in the IoT world. Secur. Priv.* ***2023****, 6, e318*
4. *Dahou, A.; Abd Elaziz, M.; Chelloug, S.A.; Awadallah, M.A.; Al-Betar, M.A.; Al-Qaness, M.A.; Forestiero, A. Intrusion Detection System for IoT Based on Deep Learning and Modified Reptile Search Algorithm. Comput. Intell. Neurosci.* ***2022****, 2022, 6473507.*
5. *Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. Neural Comput.* ***1997****, 9, 1735–1780.*