Implementation of Intrusion Detection System based on Long Short Term Memory Architecture Recurrent Neural Network

Siddhant Waghjale

Abstract - **Everyday, almost 2.5 quintillion bytes of data is shared and created every day.Due to advancement in communication, more and more data is created and shared across various networks throughout the world.This however, gives malicious people even more opportunities to exploit networks for their gain.To reduce and prevent such attacks, Intrusion Detection Systems (IDS) is used.In this paper, we create an Intrusion Detection System by applying Long Short Term Memory (LSTM) Architecture to a Recurrent Neural Network. We will be using KDD Cup 1999 dataset to train the IDS and will be able to detect which attack is taking place based on other features. By measuring its performance, we then conclude that it is one of the most effective ways of implementing an IDS.**

Index Terms-Intrusion Detection System, Long Short Term Memory, Recurrent Neural Network

I.INTRODUCTION

An Intrusion Detection System can be defined as software that monitors and tracks a network for malicious activity which can then be reported. If it further acts on the malicious activity by removing or rejecting it, it is known as an Intrusion Prevention System (IPS). Our focus however in this paper in IDS. A typical intrusion detection system performs many functions for the network.It monitors and analyses system activities.It performs auditing of system files and files in the network.It assesses integrity of the system data and files.It implements analysis of patterns in the network based on already known attacks.It detects errors and vulnerabilities in system configurations.An IDS alerts and cautions the user in case an attack is taking place as well.

IDS' can be classified on the basis of analyzed activity into Network intrusion detection systems(NIDS) and Host Intrusion detection systems (HIDS).A network intrusion detection system (NIDS) is placed at strategic points within the network, where it can monitor in bound and out bound traffic to and from all the devices on the network.Host intrusion detection systems (HIDS) runs on individual hosts on the network.A HIDS would monitor inbound and outbound data from that device only and alerts the admin if any suspicious activity is spotted.

Another way of classifying IDS' is to distinguish them based on Detection methods.In this aspect they can be classified as Signature-based IDS and Anomaly-based IDS.A Signature based IDS detects attacks by looking for specific patterns,for example,an anti-virus software.They are able to detect known attacks but cannot detect new attacks,which is why regular updating is necessary.

An Anomaly based IDS improves on this by being able to detect unknown attacks.The basic approach is to use machine learning to create a model of trustworthy activity, and then compare new behavior against this model.

In this paper,we implement an Intrusion Detection System by applying Long Short Term Memory architecture to a Recurrent Neural Network.We will train the IDS using a dataset known as KDD Cup 1999 dataset and measure the performance using detection rate and false alarm rate.

II. DATASET

The dataset we have used to train the IDS is called KDD Cup 1999 dataset. Its original form contains about four gigabytes of compressed binary TCPdump data from seven weeks of network traffic from a U.S Navy Forces headquarters. This was processed into about five million connection records[3].
It contains information for thousands of packets with the last column being the outcome,which signifies whether it is an attack or a normal packet.It contains 41 features followed by the outcome column[2].There are 22 types of attacks classified into 4 categories. Denial of Service(DoS) is an attack in which the attacker makes some computing or memory
resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.User to Root Attack (U2R) is a class of attack in which the attacker starts out with access to a normal user account on the system (perhaps gained by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system. Remote to Local Attack (R2L) occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account

on that machine exploits some vulnerability to gain local access as a user of that machine.Probing Attack is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls.

Category	Name of Attack
DoS	back, land, neptune, pod, smurf, teardrop
R2L	ftp-write, guess-passwd, imap, multihop, phf, spy, warezclient, warezmaster
U2R	buffer-overflow, loadmodule, perl, rootkit
Probe	ipsweep, nmap, portsweep, satan

Table 1 : Categories and Names of Attacks

However,we will be using a much smaller subset of this which contains 10% of the original dataset called KDD Cup 1999 10 percent data.However,in this dataset,the number of DoS attacks greatly dwarfs the attacks from the remaining 3 categories.This would mean that it would be able to detect Normal and DoS attacks easily while not working as well with U2R,R2L and Probing attacks.

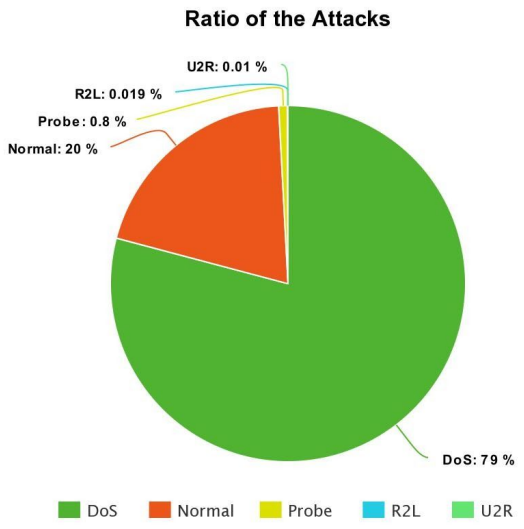

Figure 1. Ratio of distribution of attacks

In order to solve this problem,we take 500 instances of each type of attack.Those having less than 500,their whole data is taken.

III.RECURRENT NEURAL NETWORK AND LONG SHORT TERM MEMORY

A Recurrent Neural Network (RNN) is a modification of a typical feed-forward neural network.A feedforward neural network is an artificial neural network wherein connections between the nodes do not form a cycle.A Recurrent Neural Network, however,is way more equipped to handle sequential data due to its capability of having cyclic connections.Recurrent networks, on the other hand, take as their input not just the current input example they see, but also what they have perceived previously in time.

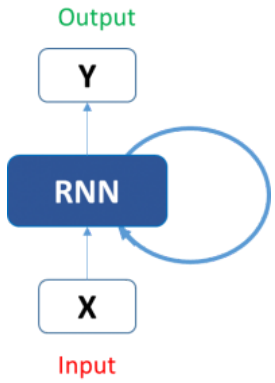

Figure 2. Recurrent Neural Network

Recurrent networks rely on an extension of backpropagation called backpropagation through time, or BPTT. Time, in this case, is simply expressed by a well-defined, ordered series of calculations linking one time step to the next, which is all backpropagation needs to work.

Recurrent Neural Networks suffer from a problem known as Vanishing Gradient Problem.During neural network training with backpropagation, the (local) minimum of the error function is found by iteratively taking small steps in the direction of the negative error derivative with respect to networks weights (i.e. gradients). With each subsequent layer the magnitude of the gradients gets exponentially smaller (vanishes) thus making the steps also very small which results in very slow learning of the weights in the lower layers of a deep network.To combat this,we have used Long Short Term Memory Architecture.

In 1997,a variation of Recurrent Neural Networks was developed by German researchers Sepp Hochreiter and Juergen Schmidhuber as a solution to the vanishing gradient problem.They termed this as the Long Short Term Architecture.

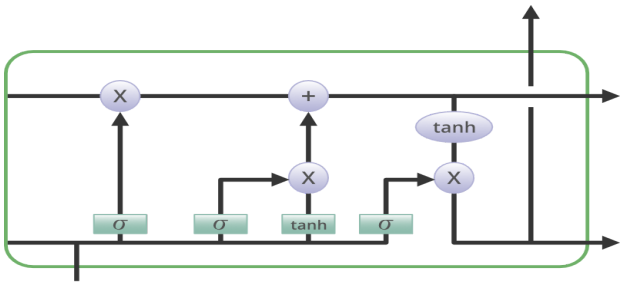

Figure.3 Architecture of LSTM

LSTMs help preserve the error that can be back-propagated through time and layers. By maintaining a more constant error, they allow recurrent networks to continue to learn over many time steps (over 1000), thereby opening a channel to link causes and effects remotely.LSTMs contain information outside the normal flow of the recurrent network in a gated cell. Information can be stored in, written to, or read from a cell, much like data in a computer's memory. The cell makes decisions about what to store, and when to allow reads, writes and erasures, via gates that open and close.

LSTM contains 3 gates,forget gate,input gate and output gate.

1.Forget Gate

The information that no longer useful in the cell state is removed with the forget gate. Two inputs *x_t* (input at the particular time) and *h_t-1*(previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state the output is 0, the piece of information is forgotten and for the output 1, the information is retained for the future use.

2.Input Gate

Addition of useful information to the cell state is done by input gate. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs *h_t-1* and *x_t*. Then, a vector is created using *tanh* function that gives output from -1 to +1, which contains all the possible values from h_t-1 and *x_t*. Atlast, the values of the vector and the regulated values are multiplied to obtain the useful information

3.Output Gate

The task of extracting useful information from the current cell state to be presented as an output is done by output gate. First, a vector is generated by applying tanh function on the cell. Then, the information is regulated using the sigmoid function and filter the values to be remembered using inputs *h_t-1* and *x_t*. Atlast, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell.

IV. EXPERIMENTAL ANALYSIS

After filtering of the dataset, so that overfitting of only some attacks is not training the model normalization of the dataset is done.

1.Normalization: Normalization converts all the values of the instances in the range 0 to 1.As the range of the values of the features gets narrowed down to a particular range because of

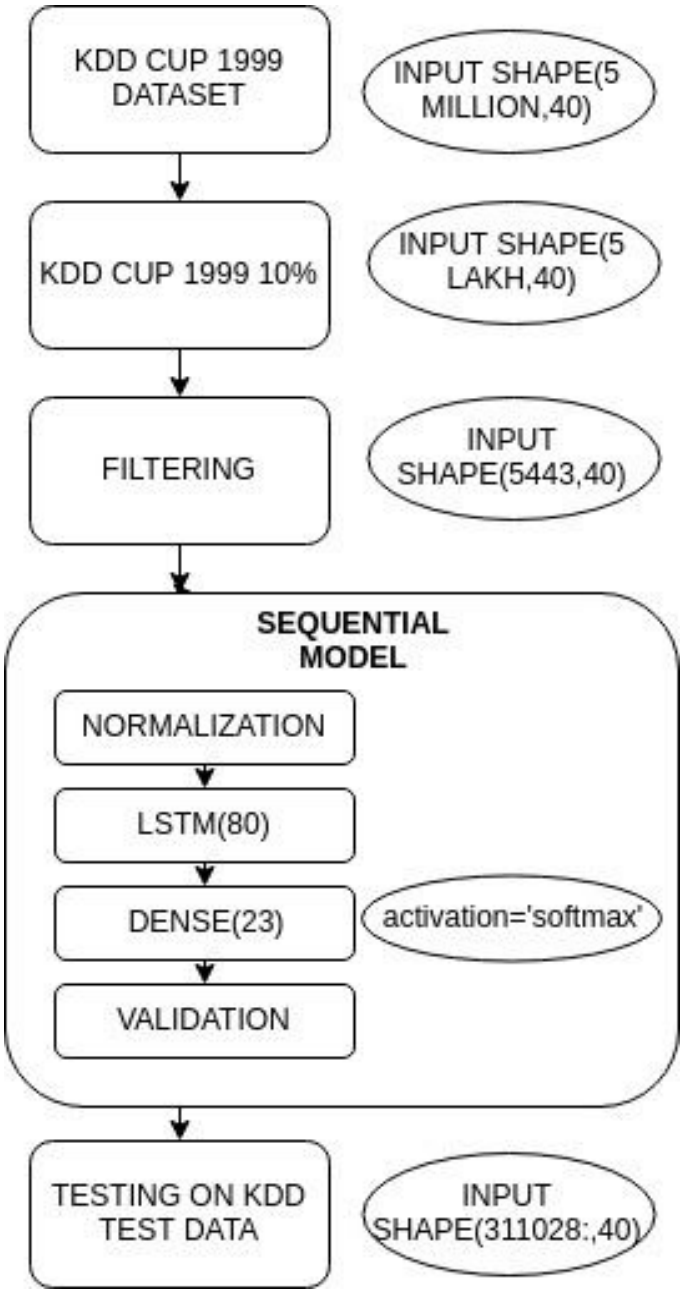

Figure 4. Work flow chart

normalization, its easy to perform computations over a smaller range of values. So, usually, the model gets trained a bit faster.

2.The IDS model set up :

The input vector is having 41 features in which the last one is the 'outcome' and the output vector is having 22 attacks and 1 non-attack(normal). So the input dimension is 40 and the output dimension is 23.

Instead of categorizing the attacks in 4 categories(DoS, R2L,U2R,Probe), we build the model with 23 output dimension so that the precision of the model is more and the model won't interpret one attack as other belonging to the same category.

The model we built is a Sequential model with 40 input dimension. Then we apply LSTM architecture to the hidden layer. Some of the parameters initially decided are:

Time-step size : 100

Batch-size : 50

Hidden-layer size : 80

Learning rate: 0.01

Activation: Softmax

Hyper-parameters are parameters for model initiation. Depending on the value of hyper-parameter, the performance is changed. We found that the number of epoch and loss function, optimizer has a great effect on the performance.These hyper-parameters are independent of each other and do not have any impact on each others performance.

Accuracy is used as a parameter to measure the performance

3.Experiment 1 : Number of Epochs

An **epoch** is a measure of the **number** of times all of the training vectors are used once to update the weights. Choosing an optimal number of epochs is very important.If the number of epochs is very less then the model will not get trained properly and the prediction of the model won't be accurate. If the number of epochs is very large then the time required for training will increase and the values will start overfitting and model will end up memorizing the training dataset.

So the value of the number of epochs plays a major role in determining the accuracy and efficiency of the model.For our dataset we started changing number of epochs from range 10 to 250.We observed that initially the accuracy is increasing as the number is increasing but after 180, the value of accuracy and loss function almost remains constant which is an indication of overfitting.

For changing the values of epochs the loss function used is 'categorical_crossentropy' and optimizer used is 'rmspop'.

Number of epochs	Accuracy	Loss
10	77.03	70.30
30	90.19	30.50
50	93.41	21.74
80	96.12	15.32
100	97.118	12.69
120	98.07	10.36
140	98.46	8.60
160	98.60	7.32
180	99.36	3.70
200	99.20	3.72
220	99.17	4.12
250	99.15	3.72

Table 2. Comparison of Accuracy and Loss wrt Epochs

After experimenting we found that the accuracy is increasing exponentially initially from 77.03 to 90.19,but after that there is a gradually slow increase in the accuracy and we found that the maximum accuracy is obtained at 180 epochs and after that the value almost remains constant.So number of epochs=180 is the optimal number of epochs for our dataset and model

3.Experiment 2: Loss function and optimizer

The choice of Optimisation Algorithms and Loss Functions for a deep learning model can play a big role in producing optimum and faster results.Loss functions are helpful to train a neural network. Given an input and a target, they calculate the loss, i.e difference between output and target variable.The different loss function we tried experimenting with are : 'Mean Square Error' and 'categorical_crossentropy'.

Optimisation Algorithms are used to update weights and biases i.e. the internal parameters of a model to reduce the error.The different types of Optimizers used are : 'RMSprop','adam','Stochastic Gradient Descent'.

Now we fix the number of epochs to 180 and change the loss function and optimizer.

Loss Function	Optimizer	Accuracy
Mean Squared Error	SGD	30.55
categorical_cross entropy	RMSprop	98.92
categorical_cross entropy'	Adam	99.36

Table 3. Comparison of Accuracy wrt Optimizer and Loss Function

For loss function,Mean Squared error is used when the target variable is continuous and categorical_crossentropy is used when the output is categorized in different features and is discrete.So in our case categorical_crossentropy performed better as we have different categories.

For Optimization algorithm,

Stochastic Gradient Descent was much faster than the other algorithms but the results produced were far from optimum.

RMSprop is an adaptive learning optimization algorithm, in case of RNN we use because of model of neural network is sequential.

Adam stands for Adaptive Moment Estimation. It also calculates different learning rate. Adam works well in practice, is faster, and outperforms other techniques.

As categorical_crossentropy and 'adam' optimizes the model better than other functions .As a result, the best algorithm for the loss function and optimizer are categorical_crossentropy and 'adam' respectively.

V. CONCLUSION

In this paper,we have implemented Intrusion Detection System using Long Short Term Memory Recurrent Neural Network.We used KDD Cup 1999 dataset to train the IDS.We further improvised by filtering 500 instances of each specific attack for better accuracy.Our IDS was quite accurately able to declare the category of attack and specific attack from the testing dataset.A comparison of accuracy with respect to increase in the number of epochs was done.We also compared different Loss Function and Optimizer to decide which one was giving the most accurate results.

We shall further improve on the model in the future so that it predicts attacks with extremely few instances accurately as well.Also we shall work on continuing to improve precision and recall,which in turn will result in better efficiency.

REFERENCES

[1] Jihyun Kim, Jaehyun Kim, Huong Le Thi Thu, and Howon Kim,"*Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection"*,2016 IEEE

[2] Tavallaee, Mahbod; Bagheri, Ebrahim; Lu, Wei; Ghorbani,"*A Detailed Analysis of the KDD CUP 99 Data Set"*

[3] KDD Cup 1999. Available on: http://kdd.ics.uci.edu/databases/kddcup 99/kddcup99.html

[4] Depren, Ozgur, et al., "*An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks*", Expert systems with Applications 2005

[5] What is IDS? https://searchsecurity.techtarget.com/definition/intrusion-detection-system

[6] Manuscript Templates for Conference Proceedings https://www.ieee.org/conferences/publishing/templates.html

[7] Alex Sherstinsky,"*Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network*",2018

[8] Depren, Ozgur,"*An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks*" Expert systems with Applications 2005

[9] Bai, Yuebin, and Hidetsune Kobayashi, "*Intrusion detection systems:technology and development*", 2003