Applying Convolutional Neural Network for Network Intrusion Detection

Vinayakumar R*, Soman KP* and Prabaharan Poornachandran†
*Centre for Computational Engineering and Networking (CEN), Amrita School of Engineering, Coimbatore
†Center for Cyber Security Systems and Networks, Amrita School of Engineering, Amritapuri
Amrita Vishwa Vidyapeetham, Amrita University, India
Email: vinayakumarr77@gmail.com

***Abstract*—Recently, Convolutional neural network (CNN) architectures in deep learning have achieved significant results in the field of computer vision. To transform this performance toward the task of intrusion detection (ID) in cyber security, this paper models network traffic as time-series, particularly transmission control protocol / internet protocol (TCP/IP) packets in a predefined time range with supervised learning methods such as multi-layer perceptron (MLP), CNN, CNN-recurrent neural network (CNN-RNN), CNN-long short-term memory (CNN-LSTM) and CNN-gated recurrent unit (GRU), using millions of known good and bad network connections. To measure the efficacy of these approaches we evaluate on the most important synthetic ID data set such as KDDCup 99. To select the optimal network architecture, comprehensive analysis of various MLP, CNN, CNN-RNN, CNN-LSTM and CNN-GRU with its topologies, network parameters and network structures is used. The models in each experiment are run up to 1000 epochs with learning rate in the range [0.01-05]. CNN and its variant architectures have significantly performed well in comparison to the classical machine learning classifiers. This is mainly due to the reason that CNN have capability to extract high level feature representations that represents the abstract form of low level feature sets of network traffic connections.**

***Index Terms*—deep learning: convolutional neural network (CNN), recurrent neural network (RNN) long short-term memory (LSTM), gated recurrent unit (GRU), intrusion detection (ID) data sets: KDDCup 99, NSL-KDD.**

I. Introduction

Over the years, Information and communication technologies (ICT) systems have been continuously playing an important role in every aspect of organization and peoples lives. At the same time, the attacks to the ICT system are diverse and continuously growing gradually. As a result, yet ICT system looks towards a glaring need for an effective intergraded network security solution. An intrusion detection system (IDS) is a prominent and most widely used tool for identifying various types of attacks. A first significant work on intrusion detection (ID) was done by John Anderson in 1931 with a research paper Computer Security threat monitoring and surveillance [1]. Recently, [2] discussed in detail the various cyber-attacks and its techniques occurred from 2001 to 2013. With the characteristics of network behavior and network type the ID is classified into (1) network based IDS: looks at the contents of individual packets with the aim to detect the malicious activity in network traffic (2) host based IDS: looks at the information of log files such as sensors, system logs, software logs, file systems, disk resources and others in each system. In real-time, the organizations heavily adopted the hybrid of network and host based IDS. This has been used as an integral component of network security in ICT systems. However, the traditional methods to IDS are completely effective at identifying the foreseen attacks whereas they are entirely shows low performance in detecting novel threats. This paper puts focus towards the network based IDS.

The analysis and classification of collected network traffic data methods are broadly classified into signature detection, anomaly detection and state full protocol analysis. Signature detection uses predefined signatures and filters that effectively detect the known intrusions. Even though signature based IDS have the capability to attack known security threats more accurately, their applicability in real time adoption is very less in recent days due to they are completely ineffective in attacking the unknown malicious threats. These unknown threats can be attacked only after that they are identified as an attack by using some other techniques or in some cases manually and tagged a signature for them. Anomaly detection depends on heuristic approaches to detect the unknown intrusions. However, the performance is not effective and results in high false positive rate. To combat this, organizations use combinations of both the signature and anomaly detection approaches. State full protocol analysis relies on the predefined protocols and its specifications with the heuristic approaches of anomaly detection in finding the deviations of protocols and applications. This can result in low false positive rate mainly due to that, it can act on the network layer, application layer and transport layer. Indeed, the existing approaches to network intrusions in commercial markets use statistical measures or threshold computing methods. These methods rely on traffic parameters such as packet length, inter-arrival time, and flow size and so on to model the network traffic in a predefined time-range. This may not be an effective method to the complex attack patterns that are happening nowadays. One prominent solution to attack the novel and present day intrusions is self-learning systems. The Self-learning system is one of powerful and proactive method that use machine learning concepts such as supervised and unsupervised algorithms to detect and classify the known and unknown security intrusions that can help in establishing the necessary actions on malicious activities in a

978-1-5090-6367-3/17/$31.00 ©2017 IEEE

timely manner.

While, the development of efficient machine learning solutions to network intrusions and adoption towards real-time IDS are in the initial stage. Though a lot of solutions had found, the applicability in real-time environment is completely ineffective. Most of the solutions resulted in achieving the high false positive rate with high computational cost [3]. This is due to the fact that most of the solutions have limited the learning patterns of attack locally with small-scale, low-level features patterns of normal and attack connections records. Most notably, machine learning has given birth to a new area called deep learning that can be defined as a complex model of machine learning algorithms. These can learn the abstract representation of complex hierarchical features globally with sequence information of TCP/IP packets.

Recently, Deep learning algorithms have witnessed as a powerful algorithms due to the remarkable results in speech processing, natural language processing and other domains [4]. The primary reason to that are the deep learning algorithms have associated with the two most important characteristics; hierarchical feature representations, learning long-term dependencies of temporal patterns in large scale sequence data. The taxonomies including the previous works of shallow and deep learning algorithms to ID is briefly outlined by [5]. In [6] used deep belief network (DBN) as classifier with the combination of restricted Boltzmann machine (RBM) for training and backpropogation for fine-tuning process of KDDCup 99. In [7] used multi-layer perceptron (MLP) to generalize the capability of architecture with corresponding to layers in attacking the intrusion that are existing in KDDCup 99. In [8] used the combination of classical machine learning classifier such as support vector machines (SVM) and neural network in classifying the connection records of KDDCup 99. In [9] used modified Jordon architecture to represent rule for patterns of attacks and showed the improvement in detection rates through various experiments. In [10] proposed the hybrid of SVM and RBM where RBM used as a mechanism for feature engineering and SVM was trained on the extracted data from RBM with following the gradient descent approach to increase the training speed.

The recent improvement in mechanisms of optimization methods and Graphical processing unit (GPU) support over parallel and distributed computing platforms have made easy to train deep learning algorithms. In [11] discussed the deep learning method with sparse auto encoder for feature learning in unlabeled data in first stage and NB-Tree, Random Tree, or J48 used as classifier in second stage. In [12] represented the TCP/IP packets of network traffic as sequence and used RNN with Hessian-Free Optimization method as classifier. Following they proposed LSTM based ID [13]. In [14] proposed deep learning approaches for traffic identification. In [15], [16], [17] has conducted the detailed research on KDDCup 99 data sets features and its classification methods by using the LSTM approach.

In [13] used LSTM for ID with the modified version of KDDCup '99' data set mainly due to the records for 'u2r' and 'r2l' are not more compared to 'DOS' and 'Normal' connection records. In [18] divided IDS into IDS-A (IDS-application layer), IDS-T (IDS-transport layer), IDS-N (IDS-network layer) and IDS-L (IDS-data link layer) and used artificial neural network for selection of right features in each IDS and used stacked auto encoder as classifier for each IDS. In [19] evaluated the comparative analysis of traditional classifier and deep learning methods for ID with KDDCup 99 challenge data set. They reported that the combination of SVM-RBMs attained higher precision in comparison to SVM and C4.5 in classifying the 'Normal' traffic connection records. In the case of 'u2r' and 'r2l' the performance was not acceptable. They improved the precision with using SMOTE package and concluded SMOTE was not a necessary requited step for SVM-RBMs. In [20] discussed CNN architecture on classifying malware to a specific family in two cases, one was based on representing the malware as gray scale images and learning the hierarchical features from those image representations. Second case classified malware to a specific family based on x86 instructions. In [21] proposed deep learning based methods for classifying malware based on system call traces. To extract the hierarchical feature representation from system call traces, authors have used CNN with n-grams and reported that the deep learning methods achieved good performance for malware classification in comparison to previously published results. To fully utilize the functionalities of deep learning approaches, they combined the hybrid classifier in such a way that the first layer was used as convolution and second layer remained as LSTM layer. Overall they claimed deep learning approaches were more effective in classifying malware based on system call traces in comparison to the traditional classifiers. A very first time, the concept of CNN was applied for ID [22]. Following, this paper aims at assessing the effectiveness of deep learning algorithms particularly CNN and the combination of convolution and sequential data modeling approaches specifically RNN, LSTM, and GRU in analysis and classification of normal and bad network connections in network ID with the publicly available and most well-known KDDCup 99 intrusion data set.

Towards this end, the remainder of this paper is arranged as follows Section II explore through CNN and its variants. Section III includes the information of NIDS data sets and the hyper parameter tuning approach for parameters of CNN and hybrid network. Section IV presents the summary of experimental results. Section IV concludes the paper with the future work direction.

II. Deep Learning

Convolutional network or convolutional neural network or CNN is an extension to tradional feed forward networks (FFN) in the context of inspiring the biological factors [23]. These were initially studied for image processing using Convolution 2D layers, pooling 2D layers and fully connected layer. Followed this, applied on natural language processing with Convolution 1D layer, pooling 1D layers and fully connected layer [24]. This infers that the CNN takes image data in the

form of 2D mesh and time series data in the form of 1D mesh in which the data are arranged in systematic time interval. Based on this, we model network traffic events as a time series data with a million of benign and malware connections and apply CNN and hybrid of CNN, and recurrent approaches on the same. The CNN is composed of Convolution 1D layer, pooling 1D layer, fully connected layer and non-linear activation function as $ReLU$.

Convolution is a primary building block of CNN. Given a 1D data of network traffic time series events as input vector $x = (x_1, x_2, ..., x_{n-1}, x_{41}, cl)$,(where $x_n \in R^d$ denotes features and $cl \in R$ denotes a class label) Convolution1D constructs a feature map fm by applying the convolution operation on the input data with a filter $w \in R^{fd}$ where f denotes the features in TCP/IP packets that results in a new set of features. A new feature map fm from a set of features f is obtained as

$$hl_i^{fm} = \tanh(w^{fm} x_{i:i+f-1} + b) \tag{1}$$

where $b \in R$ denotes a bias term. The filter hl is employed to each set of features f in a TCP/IP connection records $\{x_{1:f}, x_{2:f+1}, \ldots, x_{n-f+1}\}$ to generate a feature map as

$$hl = [hl_1, hl_2, \ldots, hl_{n-f+1}] \tag{2}$$

where $hl \in R^{n-f+1}$ and next we apply the max-pooling operation on each feature map as $\overrightarrow{hl} = \max\{hl\}$. This obtains the most significant features in which a feature with highest value is selected. However, multiple features obtain more than one features and those new features are fed to fully connected layer. A fully connected layer contains the $softmax$ function that gives the probability distribution over each class. A fully connected layer is defined mathematically as

$$o_t = soft\max(w_{ho} hl + b_o) \tag{3}$$

A. Hybrid network (CNN-RNN, CNN-LSTM, CNN-GRU)

The following methods offer eclectic mix of convolutional and recurrent neural networks. As network traffic event follows the time series patterns and the current network connection record can be classified based on the past traffic connection records. To capture the time series patterns across time-steps of newly formed features from max-pooling operation in CNN, we feed them to RNN, LSTM and GRU as defined below

$$FM = CNN(x_t) \tag{4}$$

Where, CNN is composed of convolution1D and Maxpooling1D layers, x_t denotes the input feature vector with a class label. . The newly constructed feature map vector FM is passed to RNN, LSTM and GRU to learn the long-range temporal dependencies.

CNN-RNN: Recurrent neural networks are similar to MLP with an additional cyclic loop [25]. This loop carries out past information across time steps.

$$h_t = f(w_{FMh} FM_t + w_{hh} h_{t-1} + b_h) \tag{5}$$

$$o_t = w_{ho} h_t + b_o \tag{6}$$

where f denotes the non-linear activation function, w denotes the weights, b denotes the biases and FM denotes the new feature vectors that are computed from CNN.

CNN-LSTM: Long short term memory is an improved method of recurrent neural networks (RNNs) developed to alleviate the vanishing and exploding gradient issue [26]. In contrast to the conventional simple RNN units LSTM introduces memory blocks. A memory block contains a memory cell and a set of gates. This appears as more effective in capturing long-range dependencies. A recurrent hidden layer function at time step can be generally defined as follows.

$$i_t = \sigma(w_{FMi} FM_t + w_{hi} h_{t-1} + w_{ci} c_{t-1} + b_i) \tag{7}$$

$$f_t = \sigma(w_{FMf} FM_t + w_{hf} h_{t-1} + w_{cf} c_{t-1} + b_f) \tag{8}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(w_{FMc} FM_t + w_{hc} h_{t-1} + b_c) \tag{9}$$

$$o_t = \sigma(w_{FMo} FM_t + w_{ho} h_{t-1} + w_{co} c_t + b_o) \tag{10}$$

$$h_t = o_t \odot \tanh(c_t) \tag{11}$$

where i, f, o denotes the input, forget and output gate respectively, b terms denotes bias, w denotes weight matrices, c denotes a memory cell, σ is element-wise sigmoid non-linear activation function, values in the range [0,1], $tanh$ is element-wise non-linear activation function, values in the range [-1,1] respectively and FM denotes the new feature vectors that are computed from CNN.

CNN-GRU: Gated recurrent network is an improved version of LSTM with lesser parameters [27]. This consumes less memory and computational cost in comparison to LSTM. GRU is defined as,

$$i\_f_t = \sigma(w_{FMi\_f} FM_t + w_{h\_f} h_{t-1} + b_{i\_f}) \tag{12}$$

$$f_t = \sigma(w_{FMf} FM_t + w_{hf} h_{t-1} + b_f) \tag{13}$$

$$c_t = \tanh(w_{FMc} FM_t + w_{hc}(f \odot h_{t-1}) + b_c) \tag{14}$$

$$h_t = f \odot h_{t-1} + (1 - f) \odot cl \tag{15}$$

where FM denotes the new feature vectors that are computed from CNN.

TABLE I: DESCRIPTION OF 10% OF KDDCup '99' AND NSL-KDD DATA SET

Attack category	Full data set	10 % data set			
	KDDCup 99	KDDCup 99		NSL-KDD	
	Train	Train	Test	Train	Test
Normal	972780	97278	60593	67343	9710
DOS	3883370	391458	229853	45927	7458
Probe	41102	4107	4166	11656	2422
r2l	1126	1126	16189	995	2887
u2r	52	52	228	52	67
Total		494021	311029	125973	22544

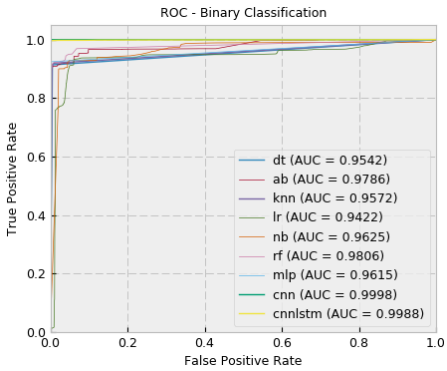

Fig. 1: ROC (TPR VS FPR)

III. Experiments

We used Google open source data flow engine, TensorFlow (r0.12.0) [28]. TensorFlow support an easy to use an environment where the mathematical operations are constructed in a data flow graph. This has support to run on various platforms such as CPU, GPU or android devices. To speed up the gradient descent computations, all deep learning architectures are trained on GPU enabled TensorFlow in single NVidia GK110BGL Tesla k40.

A. Description of Network intrusion data set (NIDS)

The network IDS data set was collected in air force base local area network (LAN), MIT Lincoln laboratory in 1998 by DARPA ID Evaluation Group (presently the Cyber Systems and Technology Group). The training data set consist of 24 attacks that were grouped into 4 attack categories and test data includes additional 14 attacks with 24 attacks and these attacks were grouped into 4 attack categories. The features include information of TCP/IP information that was extracted from both the flows of sender and receiver with a specific protocol type in a specific time window. The each traffic flows consist of 100 bytes of information. 41 features were composed of 34 continuous and 7 discrete valued and grouped into basic features in the range [1-9], content features in the range [10-22], traffic features with in a time window of in the range [23-31] and features based on host in the range [32-41]. The full data set contains more number of connection records in comparison to the attacks. In order to prevent it, we ideally formed a new training data set by adding 400 connection records to r2l and 100 connection records to u2r [29]. Moreover, we randomly removed 250 connection records from both the normal and DOS connection records. NSL-KDD is the refined version of KDDCup 99 challenge data set that is contributed by [30]. The detailed description of KDDCup 99 and NSL-KDD data is placed in Table I.

B. Hyperparameter identification

CNN is a parameterized function, so the performance solely depends on the optimal parameters. Initially, we started the experiment with a moderately sized CNN network including an input layer, hidden layer and an output layer to find right parameters and network structure. Two trails of experiments are run for filters 16, 32 and 64 with filter length 3, 5 and 5. 64 filters with filter length 5 have attained highest accuracy. To select the optimal learning rate, we performed the experiments on varied learning rate in the range [0.01-0.5]. All trails of experiments on varied learning rate are run till 500 epochs. Intuitively, lower learning rate in CNN networks have showed superior performance in distinguishing the connection records. Moreover, lower-learning rate anticipated more number of epochs to attain acceptable detection rate for low frequency attacks. The detection rate of 'r2l' and 'u2r' are low with respect to all learning rates till 400 epochs. This shows that both attacks require more than 400 epochs in training to learn the unique patterns of them. However, other attacks showed acceptable detection rate till 400 epochs in learning rate in the range [0.01-0.5]. By considering the training time, cost and the detection performance, we decided to set 0.1 as fixed learning rate for the rest of the experiments.

To select the CNN network structure for training an IDS model on KDDCup '99' to distinguish attacks from normal connection record and categorize to their category, two trails of experiments are run for the following network topologies till 100 epochs.

1) CNN 1 layer
2) CNN 2 layer
3) CNN 3 layer
4) CNN 1 layer with RNN,LSTM and GRU
5) CNN 2 layer with RNN,LSTM and GRU
6) CNN 3 layer with RNN,LSTM and GRU

The simple CNN network has showed best detection performance of high frequency attacks such as 'normal', 'dos' and 'probe'. Though, at last the complex networks have attained highest accuracy too. Most notably, the complex network topologies have took more number of epochs, possibly more than 1000 to attain the considerable detection rate for low frequency attacks. The number of optimal epochs required for learning the various attacks for each network topologies is of different. Moreover, network topologies have started to over fitting for high frequency attacks, once it has learned the complete patterns. It specifically means that the network has started to memorize the train data samples and results in decreasing the generalization performance for that specific attack. 3 layers CNN and its hybrid network has attained

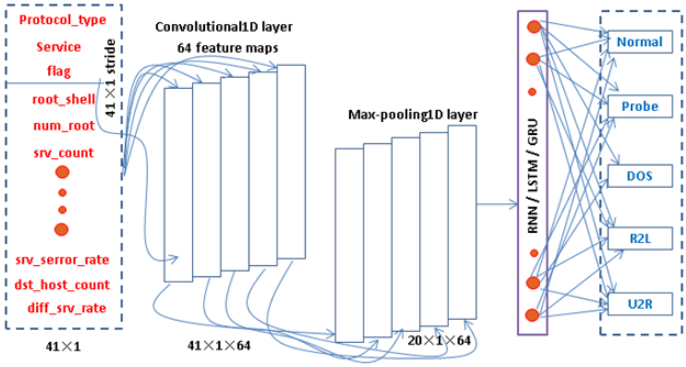

Fig. 2: Architecture of RNN unit and LSTM memory block

TABLE II: SUMMARY OF TEST RESULTS FOR KDDCup '99' IN CLASSIFYING THE CONNECTION RECORDS AS EITHER NORMAL OR ATTACK

Algorithm	Accuracy	Precision	Recall	F-score
CNN 1 layer	0.999	0.999	0.999	0.999
CNN 2 layer	0.998	0.999	0.998	0.999
CNN 3 layer	0.801	0.804	0.994	0.889
CNN 1 layer-LSTM	0.94	0.998	0.928	0.961
CNN 2 layer-LSTM	0.997	0.999	0.996	0.998
CNN 3 layer-LSTM	0.964	0.999	0.956	0.977
CNN 1 layer-GRU	0.922	0.995	0.907	0.949
CNN 2 layer-GRU	0.981	0.999	0.976	0.988
CNN 3 layer-GRU	0.936	0.999	0.921	0.958
CNN 1 layer-RNN	0.821	0.999	0.778	0.875
CNN 2 layer-RNN	0.973	1.0	0.967	0.983
CNN 3 layer-RNN	0.938	0.997	0.926	0.960

highest detection rate.

To identify the CNN network structure for distnguishing the connection record as either 'normal' or an 'attack', 2 trails of experiments are done for the afermentioned network toplogies. All experiments are run till 500 epochs. Due to overfitting the complex network performance trails the simple network structures. CNN 1 layer has performed well in comparison to other CNN network toplogies. When the max-pooling layer outputs are passed to recurrent layer to obtain temporal information, CNN 2 layer with RNN, LSTM and GRU has performed well.

C. Proposed architecture for network intrusion detection system (NIDS)

The proposed architecture for NIDS is displayed in Fig 2. It contains an input layer, hidden layer and an output layer. A hidden layer contains one or more CNN layer followed by FFN or RNN/LSTM/GRU. An input layer takes data of shape 41*1 and passes to CNN. A CNN constructs a tensor of shape 41*1*64 (64 denoted the number of filters) passes to max-pooling layer. It reduces the tensor shape into 20*1*64. This tensor can be passed to FFN for classification or RNN/LSTM/GRU to capture the temporal patterns.

D. Minimal feature sets

To know the effectiveness of minimal feature sets, CNN-LSTM is trained on 3 different minimal feature sets [31], [15]. All experiment on each minimal feature sets are run till 1000 epochs. The contribution of 11-feature sets is more towards classifying the connection record to their attack categories in comparison to the other minimal feature sets, However, the performance of 8 minimal feature sets is comparable to the 11 minimal feature sets. Moreover, the 8 feature sets performance is good, when we run experiments till 500 epochs. This infers that the CNN-LSTM has follows difficulty in learning the attack patterns. The detailed results is displayed in Table III. In all minimal feature sets, the CNN-LSTM has performed well in comparision to the other networks. Moreover, the performance of MLP is good in comparision to the CNN and CNN-LSTM architectures.

IV. Evaluation Results

Two trails of experiments are done on each CNN and its hybrid network. All the experiments are run for 1000 epochs. The performance of each trails of experiments are evaluated on the KDDCup 99 test data set. The details statistics is reported in Table IV. CNN-LSTM has performed well in comparision to the other network structures. Moreover, CNN-LSTM has learnt an aceptable detection rate relaed to all except 'u2r'.

The CNN 1 layer network and CNN-LSTM is evaluated on the KDDCup 99 test data set to identify the connection record either as normal or an attack. The performance of CNN 1 layer, CNN1 layer with LSTM and other classical machine learning classifiers are shown in Fig 1.

Generally, the inputs are passed to more than one layer in deep network and the activation in each layer facilitates to distinguish the connection records to their categories. The activation values of last layer should maximally separable for various classes of attacks. In order to visualize, we randomly selected 100 samples from the class normal, dos, probe, r2l and 50 samples from u2r and their high dimensional feature vectors of last layer are passed to t-SNE [32]. t-SNE transforms into two-dimensional representation. These vectors are plotted and shown in Fig 3. The connection records of 'normal', 'dos' and 'probe' are appeared in a separate cluster. The CNN-LSTM layer hasn't completely learnt the attack patterns of 'u2r' and 'r2l'. That's why the connection records of 'probe', 'u2r' and 'r2l' have wrongly clustered.

To understand the contribution of each features in classifying and categorizing the attack to their corresponding categories, we use backpropogation approach proposed by [33] in computer vision. The testing sample belongs to the u2r category is passed to the CNN-LSTM network. The first order Taylor expansion is applied on last layer neuron activation values. The resultant vectors are displayed in Fig 4. This helps to understand the contribution of each feature in classifying them to the corresponding class.

TABLE III: SUMMARY OF TEST RESULTS FOR MINMAL FEATURE SETS OF KDDCup '99' IN MULTI CLASS CLASSIFICATION SETTING

Algorithm	Normal		Dos		Probe		u2r		r2l		Accuracy
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	
MLP 8	0.760	0.168	0.901	0.318	0.470	0.012	0.0	0.0	0.0	0.0	0.684
CNN-LSTM 8	0.998	0.092	0.766	0.022	0.873	0.052	0.238	0.0	0.784	0.01	0.878
CNN 8	0.998	0.117	0.801	0.026	0.825	0.052	0.104	0.0	0.556	0.013	0.857
MLP 4	0.999	0.530	0.55	0.044	0.464	0.002	0.0	0.0	0.0	0.0	0.667
CNN-LSTM 4	0.999	0.082	0.862	0.062	0.863	0.067	0.209	0.005	0.248	0.002	0.846
CNN 4	0.997	0.104	0.831	0.079	0.712	0.063	0.03	0.0	0.326	0.007	0.827
MLP 4	0.871	0.079	0.927	0.17	0.553	0.011	0.0	0.0	0.0	0.0	0.885
CNN 4	1.0	0.015	0.943	0.145	0.607	0.040	0.0	0.0	0.306	0.009	0.852
CNN-LSTM 4	1.0	0.031	0.916	0.166	0.591	0.032	0.328	0.002	0.283	0.003	0.839

TABLE IV: SUMMARY OF TEST RESULTS FOR KDDCup '99' IN CATEGORIZING ATTACKS TO THEIR CORRESPONDING CATEGORIES

Algorithm	Normal		Dos		Probe		u2r		r2l		Accuracy
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	
MLP 6 layer	0.996	0.081	0.941	0.040	0.794	0.002	0.0	0.0	0.001	0.0	0.923
CNN 1 layer	0.999	0.053	0.941	0.003	0.887	0.011	0.0	0.0	0.654	0.003	0.944
CNN 2 layer	0.999	0.020	0.974	0.002	0.969	0.013	0.286	0.0	0.636	0.0	0.970
CNN 3 layer	0.999	0.022	0.975	0.003	0.925	0.012	0.343	0.0	0.633	0.0	0.970
CNN 1 layer-LSTM	0.998	0.054	0.942	0.007	0.882	0.012	0.0	0.0	0.530	0.002	0.941
CNN 2 layer-LSTM	0.999	0.028	0.974	0.008	0.781	0.005	0.0	0.0	0.712	0.002	0.970
CNN 3 layer-LSTM	0.997	0.006	0.995	0.014	0.868	0.004	0.0	0.0	0.745	0.001	0.987
CNN 1 layer-GRU	0.998	0.073	0.941	0.009	0.857	0.003	0.171	0.001	0.347	0.001	0.934
CNN 2 layer-GRU	0.999	0.031	0.972	0.005	0.873	0.004	0.0	0.0	0.724	0.001	0.969
CNN 3 layer-GRU	0.999	0.013	0.991	0.002	0.873	0.011	0.0	0.0	0.484	0.001	0.977
CNN 1 layer-RNN	0.987	0.073	0.941	0.017	0.861	0.005	0.243	0.0	0.312	0.001	0.931
CNN 2 layer-RNN	0.995	0.031	0.974	0.014	0.760	0.005	0.0	0.0	0.693	0.0	0.967
CNN 3 layer -RNN	0.999	0.027	0.974	0.004	0.912	0.007	0.029	0.0	0.674	0.001	0.969

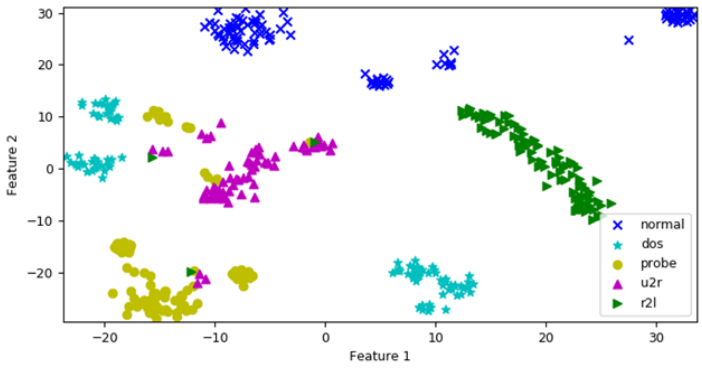

Fig. 3: Visualization of activation values of last hidden layer in CNN-LSTM

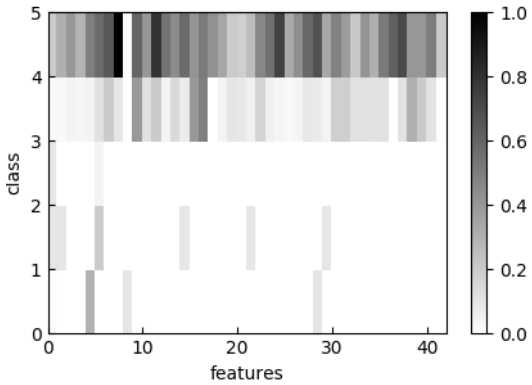

Fig. 4: saliency map for the selected connection record that belongs to r2l category

V. Conclusion

This paper analyzed the effectiveness of convolution neural network for ID by modeling the network traffic events as time-series of TCP/IP packets. The CNN algorithms outperformed the results of KDDCup 99 challenge and other published results of ID. In order to analyze the effectiveness of hybrid of recurrent neural network and convolution neural network, we used CNN as first layer with a recurrent neural network and its variants. As a result, the models dont improve the performance and showed results are comparable to CNN in most of the cases. Overall, from comprehensive evaluation measures of experiments this paper claims deep learning based approaches

such as CNN and RNN, LSTM, GRU are suitable at modeling network traffic as a sequence of TCP/IP packets in comparison to other conventional machine learning classifiers.

The attacks to ICT systems and networks are diverse and continuously evolving gradually. The attacks happened today are completely different than the attacks of one year ago. Due to this nature of attacks the model trained on DARAPA IDS data don't suits well in identifying the novel intrusions. And the attacks in DARPA IDS are well-known with including the inherent issues. Though NSL-KDD exists, it is outdated. Recently, UNSW-NB15 is introduced and we assume that this solves the issue of KDDCup 99 and NSL-KDD. However, we lack behind to evaluate the detailed analysis of them using various deep learning approaches. This remains as one of our future work.

Another direction towards future work will be to apply the discussed deep learning algorithms on real time network traffic data. The data will be generated and while transforming to connection records, the richness of them can be increased by adding additional characteristics from different logs, firewalls, alarms of each system, syslog servers, routers, and switches. To substantiate this, we consider the real time data generation as future work and apply the discussed learning algorithms on the same.

One of the major important issues experienced during training is that complex architecture required more computational cost. Due to this we are not able to train more complex architecture. This can be done by using advance hardware and distributing the jobs in training. Thus this can further enhance the reported results of attack detection rate.

REFERENCES

[1] J. P. Anderson *et al.*, "Computer security threat monitoring and surveillance," Technical report, James P. Anderson Company, Fort Washington, Pennsylvania, Tech. Rep., 1980.

[2] T. Vaidya, "2001-2013: Survey and analysis of major cyberattacks," *arXiv preprint arXiv:1507.06673*, 2015.

[3] W. Lee, W. Fan, M. Miller, S. J. Stolfo, and E. Zadok, "Toward cost-sensitive modeling for intrusion detection and response," *Journal of computer security*, vol. 10, no. 1-2, pp. 5–22, 2002.

[4] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[5] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," *arXiv preprint arXiv:1701.02145*, 2017.

[6] N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in *Advanced Cloud and Big Data (CBD), 2014 Second International Conference on*. IEEE, 2014, pp. 247–252.

[7] M. Moradi and M. Zulkernine, "A neural network based system for intrusion detection and classification of attacks," in *Proceedings of the IEEE International Conference on Advances in Intelligent Systems-Theory and Applications*, 2004, pp. 15–18.

[8] S. Mukkamala, A. H. Sung, and A. Abraham, "Intrusion detection using an ensemble of intelligent paradigms," *Journal of network and computer applications*, vol. 28, no. 2, pp. 167–182, 2005.

[9] J.-S. Xue, J.-Z. Sun, and X. Zhang, "Recurrent network in network intrusion detection system," in *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, vol. 5. IEEE, 2004, pp. 2676–2679.

[10] J. Yang, J. Deng, S. Li, and Y. Hao, "Improved traffic detection with support vector machine based on restricted boltzmann machine," *Soft Computing*, vol. 21, no. 11, pp. 3101–3112, 2017.

[11] Q. Niyaz, W. Sun, A. Y. Javaid, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (Formerly BIONETICS), BICT-15*, vol. 15, 2015, pp. 21–26.

[12] J. Kim and H. Kim, "Applying recurrent neural network to intrusion detection with hessian free optimization," in *International Workshop on Information Security Applications*. Springer, 2015, pp. 357–369.

[13] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[14] Z. Wang, "The applications of deep learning on traffic identification," *BlackHat USA*, 2015.

[15] R. C. Staudemeyer and C. W. Omlin, "Extracting salient features for network intrusion detection using machine learning methods," *South African computer journal*, vol. 52, no. 1, pp. 82–96, 2014.

[16] ——, "Evaluating performance of long short-term memory recurrent neural networks on intrusion detection data," in *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*. ACM, 2013, pp. 218–224.

[17] R. C. Staudemeyer, "Applying long short-term memory recurrent neural networks to intrusion detection," *South African Computer Journal*, vol. 56, no. 1, pp. 136–154, 2015.

[18] M. E. Aminanto and K. Kim, "Deep learning-based feature selection for intrusion detection system in transport layer."

[19] B. Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection," in *Communication Software and Networks (ICCSN), 2016 8th IEEE International Conference on*. IEEE, 2016, pp. 581–585.

[20] D. Gibert Llauradó, "Convolutional neural networks for malware classification," Master's thesis, Universitat Politècnica de Catalunya, 2016.

[21] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2016, pp. 137–149.

[22] R. Upadhyay and D. Pantiukhin, "Application of convolutional neural network to intrusion type recognition," Ben-Gurion University of the Negev, Tech. Rep., 2017.

[23] Y. LeCun *et al.*, "Generalization and network design strategies," *Connectionism in perspective*, pp. 143–155, 1989.

[24] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[25] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[28] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[29] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[30] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*. IEEE, 2009, pp. 1–6.

[31] R. Staudemeyer and C. Omlin, "Feature set reduction for automatic network intrusion detection with machine learning algorithms," p. 105, 2009.

[32] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[33] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.