EPiC Series in Computing

Volume 69, 2020, Pages 69–79

Proceedings of 35th International Conference on Computers and Their Applications

Convolutional Neural Networks with LSTM for Intrusion Detection

Mostofa Ahsan and Kendall E. Nygard

Department of Computer Science

North Dakota State University

mostofa.ahsan@ndsu.edu, kendall.nygard@ndsu.edu

Abstract

A variety of attacks are regularly attempted at network infrastructure. With the increasing development of artificial intelligence algorithms, it has become effective to prevent network intrusion for more than two decades. Deep learning methods can achieve high accuracy with a low false alarm rate to detect network intrusions. A novel approach using a hybrid algorithm of Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) is introduced in this paper to provide improved intrusion detection. This bidirectional algorithm showed the highest known accuracy of 99.70% on a standard dataset known as NSL KDD. The performance of this algorithm is measured using precision, false positive, F1 score, and recall which found promising for deployment on live network infrastructure.

1 Introduction

Competing with the recent growth of computer networks and internet usage, network intrusion has also become a crucial issue. A question frequently arising from security advocates concerns why we bother trying to detect intrusions if we have already installed firewalls, operating system patches and encrypted passwords? However, intrusions still often occur. Just as people sometimes forget to lock their personal computer, they also forget to update their firewall, or verify carefully before opening a malicious email. Even with very advanced protection utilized on a network or personal computer, it is not safe [27, 29]. As described by Heady et al [1], "An intrusion is a set of actions that attempt to compromise the confidentiality, integrity or availability of information resources." The unauthorized malicious users often try to breach a vulnerable system by such actions as directly attempting a break-in, launching a penetration attack, or seeking authentication in a devious way. Increases in intrusions is increasing rapidly, commensurate with increasing digital lifestyles. People depend upon internet services and tend to steadily increase their usage. Security breaches greatly affect people today. Thus it

is of high importance to develop effective precautionary measures to safeguard users from attacks to which they are susceptible [2, 35].
System designs which are used to detect malicious actions in a network are called Network Intrusion Detection Systems or NIDS [3, 2]. The two main types are the Signature-based Network Intrusion Detection Systems or SNIDS and Anomaly Detection based Network Intrusion Detection Systems or ADNIDS. SNIDS are aimed at detecting an unauthorized access or intrusion by matching patterns on the features for which it is trained. An ADNIDS type of system detects an anomaly when there is a deviation in the normal traffic pattern [2]. Since an ADNIDS is highly prone to false alarms, SNIDS are regarded as a preferred approach for Detecting Network Intrusions. Various artificial intelligence approaches are the key techniques used in SNIDS. Since machine learning techniques can be efficient and effective in detecting patterns from historical data, they have been employed to develop NIDS for anomaly detection. But, the drawback of machine learning approaches is that they rely on training and testing on datasets that may have features that differ from those encountered in new datasets to which they are applied. This motivates investigation into deep learning methods, which are promising because they are robust and efficient for problems with a large number of features [2].

Deep learning is a branch of machine learning that has achieved outstanding performances in various application domains, particularly with large datasets [4]. Deep learning methods learn automatically from raw data, then output results by operating in an end-to-end fashion which is very practical. Many deep learning approaches have been shown to proven well for NIDS. In short, state of the art Neural Networks are promising, particularly because they tend to be agent based, can reduce false alarm rates, and work well for large datasets.

2 Related Work

Analytical methods for network monitoring for Intrusion Detection (ID) gained recognition around 1980 with the work reported by Anderson [5]. Various methods of Artificial Intelligence were also developed fairly early, but were limited in their success due to difficulties in scaling to large and complex data streams. Machine learning techniques have evolved and now play a major and increasing role in distinguishing authentic and valid use from malicious intent. A modified support vector machine (SVM) combined with kernel principal component analysis (KPCA) and Genetic algorithms (GAs) showed effective results by around 2014 [6]. A GA is an approach that uses a problem-solving metaphor that is based on concepts from biological genetics. Basically, a GA follows the biological understanding that highly fit organisms are more likely than others to reproduce and pass on positive traits to their offspring. This metaphor has been used in many problem-solving domains to successfully develop high performance solutions through simulated evolution. However, in the training process of GA, fixed rules are implemented from the data, resulting tables with large numbers of rules that carry out the monitoring functions of NIDS [9, 36, 37]. A novel scheme using Principal Component Analysis (PCA) identifying anomalies as outliers were reported by Shyu et al. [8] in 2003. Since the anomalous data is highly susceptible to outliers, it is possible to train the datasets using a random oversampling or Synthetic Minority Oversampling Technique (SMOTE) method. This approach is effective in classifying malicious phishing emails [18, 21]. Clustering is a process of creating a partition of data in a way that each group contains similar characteristics. By finding a matching pattern, the data can be segregated. Since clustering can learn from the record and audit the data itself, it has a significant benefit for Intrusion Detection [9]. Mini batch K-means clustering produced very good accuracy by using K-means principal idea of allocating different random groups of distinct memory sizes, which facilitates the easiest process to store [10]. Since it takes different batches, it is somewhat time consuming, which impedes usage in practice.

For continuous streaming data, classification techniques play a major role in anomaly detection. To enhance the user experience and accommodate fast network streams, Li et al. proposed a K-Nearest Neighbor (KNN) classification in the setting of wireless sensor networks [7, 38, 21]. Machine-learning algorithms such as decision tree, rule-based induction, Bayesian network, and genetic algorithm have significantly enhanced network security. More recently, ensemble learning is being used for classification techniques in the quest of avoid false alarms. Classifier of Ensemble Accuracy (AUE) is a modified version of the Accuracy Weighted Ensemble (AWE) method, which uses the concept of updating a classifier according to the distribution [11].

In practice, traditional machine learning models, like the support vector machine (SVM) and k-nearest neighbor (KNN), contain either no hidden layer or just one. These traditional machine learning models are also referred to as shallow models [13]. Deep learning methods integrate high-level feature extraction and classification tasks, which overcome most of the limitations of shallow learning and further promote the progress of intrusions detection systems [12]. Deep learning methods can automatically extract features and perform classification on the dataset. Examples methods include Auto Encoder, deep belief network (DBN), deep neural network, and recurrent neural network (RNN) [14, 15]. Previously many deep learning approaches are shown to be effective for NSL KDD datasets [2, 4, 5, 6, 9, 11, 12, 13, 14, 15, 17, 18, 19, 22]. Stacked autoencoders were used in IEEE 802.11 network platforms to detect intrusion, which had an accuracy of 98.60% [16]. Ma et al. [17] designed a hybrid method that combined spectral clustering and deep neural network for intrusion detection on a NSL KDD dataset and achieved an accuracy of 72.64%. The gated recurrent unit (GRU) recurrent neural network (RNN) combination used as (GRU-RNN) was developed to detect intrusion over a software-defined network (SDN) and achieved an overall accuracy of 89% [19]. A hybrid of the stacked non-symmetric auto encoder and random forest was used for NIDS by Shone et al. [20]. Muna et al. [22] used a deep autoencoder for feature extraction and feedforward neural networks for classification for intrusion detection. A Restricted Boltzmann machine (RBM) is also effective to classify normal and anomalous network traffic [23].

These deep learning approaches are promising and effective, but still there are detection errors, such as a low detection rate for unprecedented attacks and high false-positive rate for minority attacks. To overcome these classification issues, the work of this paper concerns a novel technique that uses a hybrid of Convolutional Neural Network (CNN) and a Long Short Term Memory neural network (LSTM) to improve the detection rate of unknown attacks provide low false-positive rates for minority attacks.

3 Dataset

Considerable machine learning work has been done using the KDD Cup 1999 dataset. But this dataset had disadvantages including redundant records. The training dataset has 78% redundancy, and the testing data has 75% duplicate records. As a result, most of the prediction was biased [9]. Since the availability of the public data set of network intrusion systems is limited, a new version of this dataset also known as NSL KDD is now used by many researchers. The newer version combines some original data from the previous version and the redundancy of records is eliminated. The datasets are made of basically four types of attack classes [24,25]. The categorical attack classes are described in Table 1 below.

Table 1: Attack categories and their description.

Name of the attack	Description
Denial of Service (DoS)	Denial of Service is an attack category that depletes the victim's resources and reduces ability to handle legitimate requests – e.g. syn flooding. Relevant features: "Percentage of packets with error", "source bytes" [18,24,25]. Frequency in training dataset: 45,927 & in testing dataset: 7,458.
Probe attack (probe)	Surveillance and other probing attack whose objectives are to collect information about victim, remotely – e.g. port scanning. Relevant features: "source bytes", "duration of connection" [9, 24, 25]. Frequency in training dataset: 11,656 & in testing dataset: 2,421.
User to Root (U2R)	Unauthorized access to local root user privileges an attack type, that is used by an attacker to log into the system as a local user and get administrator access by exploiting some vulnerability in the victim's system – e.g. buffer autoflow attacks. Relevant features: "number of shell prompts invoked", "number of file creation" [9, 18, 24, 25]. Frequency in training dataset: 52 & in testing dataset: 200.
Root to Local (R2L)	Unauthorized access from a remote system as an administrator. Then the attacker intrudes into remote machine and get access to the victim's local machine – e.g. password guessing. Relevant features: Host level features: "number of failed login attempts" and network level features: "service requested", "duration of connections" [9, 24, 25]. Frequency in training dataset: 995 & in testing dataset: 2,754.

In each of the rows there are 41 features to characterize attributes of the flow and produce labels assigned as normal or attack type. These features can be primarily classified into four categories [26] as shown below.

Basic features. These are the attributes of the individual TCP connection

Redundant features. These are the duration, protocol_type, service, src_bytes, dst_bytes, flag, land, wrong_fragment, urgent

Content features. These are the attributes that suggests the domain knowledge within a connection. Redundant ones include hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file_creation, num_shells, num_access_files, num_outbound_cmds, is_hot_login, is_guest_login

Traffic features. These are the attributes that are calculated using only two-seconds of window time. Redundant ones include count, serror_rate, rerror_rate, same_srv_rate, diff_srv_rate, srv_count, srv_serror_rate, srv_rerror_rate, srv_diff_host_rate

Host features. These are the attributes that are designed to attack and access in more than two seconds: Redundant ones dst_host count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate,

dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate,dst_host_rerror_rate, dst_host_srv_rerror_rate.

4 Algorithms Used

Alternative machine learning algorithms have been shown promising to predict intrusion on NSL KDD datasets in previous work. But since shallow learning has a high false-positive rate, the work of this paper addresses deep learning methods that are a sub-field of machine learning that improves and advances shallow learning. Deep learning facilitates the modelling of complex relationships and concepts using multilevel representations [20]. In the work described here, we compare five well established deep learning algorithms that can be instantiated with our approach. We have worked Modular Neural Network (MNN), Artificial Neural Network (ANN), Feed Forward Neural Network, Auto Encoder (AE) and Recurrent Neural Network (RNN).

4.1 DenseNet (Densely Connected Networks)

Residual Network (ResNet) provides important new knowledge for parametrization of the functions in deep learning. DenseNet is a logical extension of ResNet. Recent work has addressed problems with vanishing gradients within ResNet as the method combines features through summation in passes to a next layer. DenseNet connect does not use summation, but employs a feed-forward technique. In DenseNet, each layer has direct access to the gradient from the loss function and the original input signal, which provides an improved flow of information and gradient accuracy throughout the network. Moreover, it has a regularization effect that reduces overfitting on tasks with similar training set sizes. The most important difference in comparison to other deep learning methods is that DenseNet have very narrow layers, for example, k=8, which refers to the hyperparameter k, the growth rate of the neural network. We have used Rectified Linear Unit (Relu) for the first three layers and the Softmax function for the activation layer for our experiment. We can write DenseNet as the following:

$$f(x)=f(0)+f'(x)x+12f''(x)x2+16f'''(x)x3+o(x3)$$

4.2 CNN (Convolutional Neural Network)

CNN, also known as ConvNet, is a deep learning algorithm that is mostly used for image classification by assigning various aspects or objects in the image and enable differentiation of one from another. The architecture of CNN resembles the connectivity pattern of neurons in the human brain, and was inspired by the Visual Cortex. It consists of several steps for classification of the dataset, such as Convolution, max-pooling, full connection, and fully connected-Relu. The convolution plays a major role for feature extraction and resizing data after multiple steps.

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)g(t-\tau)_{d\tau}$$

4.3 GRU (Gated Recurrent Units)

To address the vanishing gradient problem of standard RNN, GRU uses an update gate function to update and reset the gate for nonlinear output. Since in this paper we experimented multiclass

prediction, we choose to use a sigmoid activation function along with GRU to enhance accuracy. The gate function plays a vital role in updating how much of the past information must to be passed to the next layer. The update and reset equation of the gate is described below. Here W(z) is the network's own weight, which is multiplied by x_t when it is plugged into the network unit. The same process applied for h_{t-1} , which holds the information for the previous t-1 and multiplied by its own weights U(z).

The update function is: $z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1})$

The reset function is: $r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1})$

4.4 Bi-LSTM (Bidirectional Long Short Term Memory)

In a traditional Neural Network, all the input and outputs are independent of each other. But, in the case of predicting the next elements in the series or word in a sentence, the previous features or elements needed to remember for predicting the future element. RNN creates a loop, which helps in the persistence of these types of information. Bidirectional RNN usually brings together two independent RNNs, which enable running input in two directions, such as past to future and future to past. A Bidirectional LSTM also acts as Bidirectional RNN by preserving both historical and prediction results [28].

4.5 AE (Autoencoder)

AEs are a specific type of feedforward neural network where the size of the input is the same as the size of the output. AE compresses the input into a lower-dimensional code and then again reconstructs the output back from the representation. It consists of three major components: encoder, code and decoder. AE is used for mostly unsupervised learning because it does not need explicit labels for training. More specifically, we call them self-supervised, since they generate their own labels from the training data set. The encoder and decoder functions are described with the encoder function denoted by ϕ, which maps the original data X to a latent space F, which is situated at the bottleneck. The decoder function is denoted by ψ, which maps the latent F at the bottleneck to the output.

$$\emptyset : x \to \mathcal{F}$$

$$\psi : \mathcal{F} \to x$$

$$\phi, \psi = \text{argmin } \|X - (\psi \, o \, \phi) \, X\|^2$$

4.6 Proposed Hybrid of CNN and LSTM

Unlike traditional Convolutional Neural Network (CNN), RNN help to create an interaction between the input sequence, and hence provides a new approach to feature hybrid [28, 30]. As illustrated in Figure 1, investigators have devised methods for hybridizing the features using LSTM (a variant of RNN), which can extract the long-term dependencies of the data features in the sequence to improve the recognition accuracy [28, 30, 31, 32, 33, 34, 39]. In the work of this paper, we have developed a new but related strategy by using multiple convolutional kernels to extract features from the dataset. Moreover, this method establishes an end-to-end mapping of the relationship between the features and the attack types. Our approach consists of two stages, the first part is feature extraction based on CNN, and a feature fusion part based on LSTM in the later part. In the first stage, the forward propagation process is applied by assuming that an l layer is a convolution layer and the l-1 layer are a pooling layer or another input layer for the next extraction process. The equation behind the first layer is:

$$x_{jl} = f(\sum_{i \in M_j} x_{il-1} \times k_{ijl} + b_{jl})$$

The variable x_{jl} in the above equation denotes the jth feature image of the l layer. The latter part shows the convolution operation and summation for all feature maps of the l −1 layer and the jth convolutional kernel of layer l, and then add an offset parameter and then passes the activation function f(*). Among them, l is the index for the layers, f is the activation function, Mj is an input feature map of the upper layer, b is an offset, and k is the convolutional kernel. For downsampling, assume the l layer as the pooling layer and l -1 is the convolutional layer. The formula is described below:

$$x_{jl} = f(\beta_{jl}\, down\, (x_{jl-1}) + b_{jl})$$

The feature extraction stages uses Relu functions for both convolutions and pooling. For the first layers we used 48 convolutional kernels with 3x3 kernel sizes. The max-pooling was followed by 16 convolutional kernels with 3x3 kernel sizes and pooling length 2, and output size for the LSTM is set at 70. The Softmax function is applied for the attack types. The Adam optimizer is applied for gradient descent and a dropout value of .01.

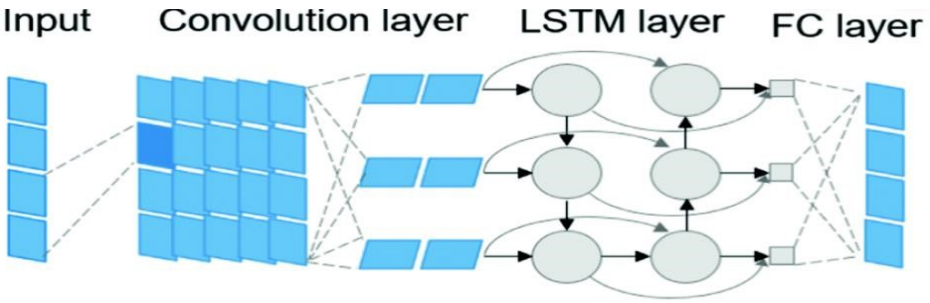

Figure 1: Illustration of the Hybrid CNN and LSTM architecture [39].

5 Experiments and Results

For deep learning methods, preprocessing of data always plays a major role. The first challenge was to convert the class labels into different four attack types. We segmented all the raw data into five categories including normal. Then we randomly selected ten percent of the training dataset and five thousand testing samples. Then, the data was normalized and preprocessed in scalar format to feed the neural networks as input.

The Autoencoder produced a very low accuracy of 37.65% without any hyperparameter tuning. For experimental standards, we set the epoch size of every method to 100. We have observed that DenseNet was able to produce 94.98% accuracy with only 20 epochs. Bidirectional LSTM was very close to DenseNet. It achieved the highest accuracy of 97.32%. For CNN we used a filter size of 16x16 and 50% dropout, which achieved an accuracy of 95.72% accuracy. The Gated Recurrent Unit (GRU) with

a Softmax activation function achieved 97.36% accuracy. The hybrid of CNN and LSTM, considered bidirectional approach, is able to outperform all other algorithms by achieving the highest 99.70% accuracy, as shown in Figure 2. For the convolutional layer, we used the Relu function and for the activation function, we used Softmax for 100 epochs. The Receiver Operating Characteristics (ROC) curve below have plotted the data of the algorithms those have achieved the highest accuracy. We have run several combinations of kernel sizes and pooling length to obtain the best result.

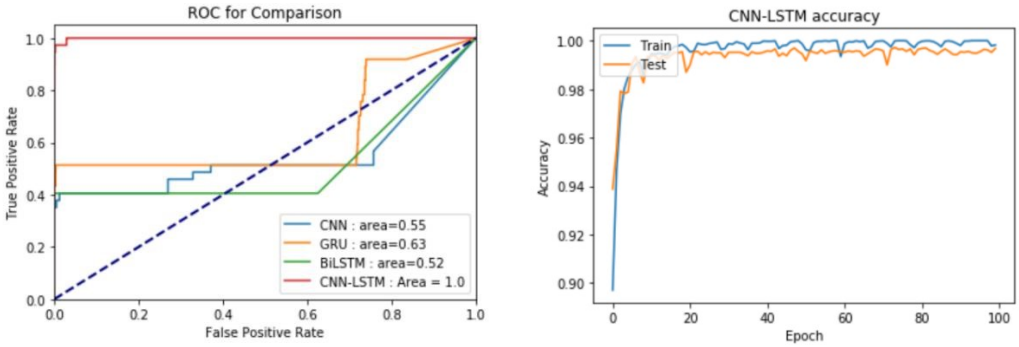

Figure 2: Top four Algorithms ROC curve (left) and hybrid algorithm accuracy (right).

We observe that the false positive for a Probe attack is high. But, every class label is predicted nearly perfectly. The overall f1 score is promising for all the attack types as plotted. on Figure 3.

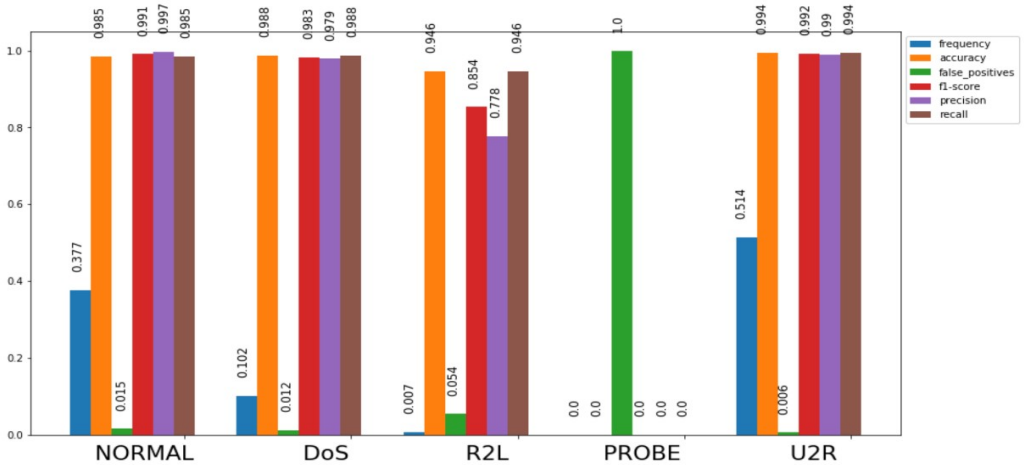

Figure 3: Individual class label result analysis.

There was a consistent increment in the accuracy of the training dataset from 93.38% in epoch 1 to 99.70 % in epoch 78. However, we observe that there is a slight decrease in testing accuracy after epoch 80. The ROC curve also shows a higher accuracy of 99 % as in Figure 2.

6 Conclusion and Future Work

In this work we have established that a hybrid of Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) is a very effective approach tor network intrusion detection.

Unprecedented high accuracy on a standard NSL KDD dataset was achieved without applying any hyperparameter tuning. This paper establishes that deep learning methods are very promising and effective for anomaly detection and intrusion prevention. We plan future experiment aimed at tuning this hybrid method for even better accuracy. We also plan to experiment with the algorithm on a live network.

References

1. Heady, Richard, et al. The Architecture of a Network Level Intrusion Detection System. No. LASUB-93-219. Los Alamos National Lab., NM (United States); New Mexico Univ., Albuquerque, NM (United States). Dept. of Computer Science, 1990.
2. Gurung, S., Ghose, M.K. and Subedi, A., 2019. Deep Learning Approach on Network Intrusion Detection System using NSL-KDD Dataset. International Journal of Computer Network and Information Security (IJCNIS), 11(3), pp.8-14.
3. Forouzan, Behrouz A., and Debdeep Mukhopadhyay. Cryptography and network security (Sie). McGraw-Hill Education, 2011.
4. Yin, Chuanlong, et al. "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks." IEEE Access 5 (2017): 21954-21961.
5. Javaid A, Niyaz Q, Sun W, et al. A Deep Learning Approach for Network Intrusion Detection System[C]// EAI International Conference on Bio-Inspired Information and Communications Technologies. ICST, 2016:21-26.
6. Kuang F, Xu W, Zhang S. A Novel Hybrid KPCA and SVM with GA Model for Intrusion Detection[J]. Applied Soft Computing Journal, 2014, 18(C):178-184.
7. Li W, Yi P, Wu Y, et al. A New Intrusion Detection System Based on KNN Classification Algorithm in Wireless Sensor Network[J]. Journal of Electrical and Computer Engineering, 2014, 2014(5):1-8.
8. Shyu, M-L., S-C. Chen, K. Sarinnapakorn, and LW. Chang, ―A Novel Anomaly Detection Scheme Based on Principal Component Classifier‖, ICDM Foundation and New Direction of Data Mining workshop, 03-1221.1-2312, 2003.
9. Seraphim, B. Ida, et al. "A Survey on Machine Learning Techniques in Network Intrusion Detection System." 2018 4th International Conference on Computing Communication and Automation (ICCCA). IEEE, 2018.
10. Peng, Kai, Victor CM Leung, and Qingjia Huang. "Clustering approach based on mini batch k-means for intrusion detection system over big data." IEEE Access 6 (2018): 11897-11906.
11. Ahmad, Iftikhar, et al. "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Ddetection." IEEE Access 6 (2018): 33789-33795.
12. Yang, Yanqing, et al. "Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural

Network." Sensors 19.11 (2019): 2528.

13. Ding, Yalei, and Yuqing Zhai. "Intrusion Detection System for NSL-KDD Dataset Using Convolutional Neural Networks." Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence. ACM, 2018.
14. Denton, A. M., Ahsan, M., Franzen, D., and Nowatzki, J. (2016, December). Multi-scalar analysis of geospatial agricultural data for sustainability. In 2016 IEEE International Conference on Big Data (Big Data) (pp. 2139-2146). IEEE.
15. Ahsan, Mostofa, Rahul Gomes, and Anne Denton. "Application of a Convolutional Neural Network using transfer learning for tuberculosis detection." 2019 IEEE International Conference on Electro Information Technology (EIT). IEEE, 2019.

16. Thing, Vrizlynn LL. "IEEE 802.11 Network Anomaly Detection and Attack Classification: A Deep Learning Approach." 2017 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2017.
17. Ma, Tao, et al. "A Hybrid Spectral Clustering and Deep Neural Network Ensemble Algorithm for Intrusion Detection in Sensor Networks." Sensors 16.10 (2016): 1701.
18. Ahsan, Mostofa, Rahul Gomes, and Anne Denton. "SMOTE Implementation on Phishing Data to Enhance Cybersecurity." 2018 IEEE International Conference on Electro/Information Technology (EIT). IEEE, 2018.
19. Tang, Tuan A., et al. "Deep Recurrent Neural Network for Intrusion Detection in Sdn-based Networks." 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft). IEEE, 2018.
20. Shone, Nathan, et al. "A Deep Learning Approach to Network Intrusion Detection." IEEE Transactions on Emerging Topics in Computational Intelligence 2.1 (2018): 41-50.
21. Gomes, Rahul, Mostofa Ahsan, and Anne Denton. "Fusion of SMOTE and Outlier Detection Techniques for Land-Cover Classification Using Support Vector Machines."
22. Muna, AL-Hawawreh, Nour Moustafa, and Elena Sitnikova. "Identification of Malicious Activities in Industrial Internet of Things Based on Deep Learning Models." Journal of Information Security and Applications 41 (2018): 1-11.
23. Aldwairi, Tamer, Dilina Perera, and Mark A. Novotny. "An Evaluation of the Performance of Restricted Boltzmann Machines as a Model for Anomaly Network Intrusion Detection." Computer Networks 144 (2018): 111-119.
24. Tavallaee, Mahbod, et al. "A Detailed Analysis of the KDD CUP 99 Data Set." 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. IEEE, 2009.
25. Dhanabal, L., and S. P. Shantharajah. "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms." International Journal of Advanced Research in Computer and Communication Engineering 4.6 (2015): 446-452.
26. Aggarwal, Preeti, and Sudhir Kumar Sharma. "Analysis of KDD Dataset Attributes - Class Wise for Intrusion Detection." Procedia Computer Science 57 (2015): 842-851.
27. Md Minhaz Chowdhury, Kendall E. Nygard, Deception in Cyberspace: an Empirical Study on a Con Man Attack, The 16th Annual IEEE international conference on electro information technology, May 14-17, 2017, Lincoln, Nebraska, U.S.A
28. Wang, Junliang, Jie Zhang, and Xiaoxi Wang. "Bilateral LSTM: A two-dimensional long short-term memory model with multiply memory units for short-term cycle time forecasting in re-entrant manufacturing systems." IEEE Transactions on Industrial Informatics 14.2 (2017): 748-758.
29. Md Minhaz Chowdhury, Jingpeng Tang, Kendall E. Nygard, An Artificial Immune System Heuristic in a Smart Grid, The 28th International Conference on Computers and Their Applications, 2013, Waikiki, Honolulu, Hawaii, USA.
30. Greff, Klaus, et al. "LSTM: A Search Space Odyssey." IEEE transactions on neural networks and learning systems 28.10 (2016): 2222-2232.
31. Tsironi, Eleni, et al. "An Analysis of Convolutional Long Short-term Memory Recurrent Neural Networks for Gesture Recognition." Neurocomputing 268 (2017): 76-86.
32. Zhou, Xiaoqiang, et al. "Recurrent Convolutional Neural Network for Answer Selection in Community Question Answering." Neurocomputing 274 (2018): 8-18.
33. Zhao, Rui, et al. "Learning to Monitor Machine Health with Convolutional Bi-directional LSTM Networks." Sensors 17.2 (2017): 273.
34. Nunez, Juan C., et al. "Convolutional Neural Networks and Long Short-term Memory for Skeleton Based Human Activity and Hand Gesture Recognition." Pattern Recognition 76 (2018): 80-94.
35. Md Minhaz Chowdhury, Kendall Nygard, "Machine Learning within a Con Resistant Trust Model", The 33rd International Conference on Computers and Their Applications (CATA 2018), March 19-21, 2018, Flamingo Hotel, Las Vegas, Nevada, USA.

36. Md Minhaz Chowdhury, Kendall E. Nygard, Krishna Kambhampaty, Maryam Alruwaythi, “Deception in Cyberspace: Performance Focused Con Resistant Trust Algorithm”, The 4th Annual Conference on Computational Science & Computational Intelligence, 2017.
37. Md Minhaz Chowdhury, Kendall E. Nygard, An Empirical Study on Con Resistant Trust Algorithm for Cyberspace, The 2017 World Congress in Computer Science, Computer Engineering, & Applied Computing, July 17-20, 2017.
38. Gomes, Rahul, Mostofa Ahsan, and Anne Denton. "Random Forest Classifier in SDN Framework for User-based Indoor Localization." 2018 IEEE International Conference on Electro/Information Technology (EIT). IEEE, 2018
39. Kim, Tae-Young, Cho, Sung-Bae, "Predicting Residential Energy Consumption using CNN-LSTM Neural Networks," Energy, Elsevier, vol. 182.C (2019), 72-81