Network Traffic classification and Abnormal Behavior Detection using Deep Learning

Submitted By

Dimpal Shah

16MCEI20

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2018

Network Traffic classification and Abnormal Behavior Detection using Deep Learning

Major Project

Submitted in fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering (Information and Network Security)

Submitted By

Dimpal Shah

(16MCEI20)

Guided By

Dr.Ankit Thakkar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2018

Certificate

This is to certify that the major project entitled "**Network Traffic classification and Abnormal Behavior Detection using Deep Learning**"submitted by **Dimpal Shah (16MCEI20)**, towards the fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering (Information and Network Security) of Nirma University, Ahmedabad, is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-I and part-II, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr.Ankit Thakkar
Guide & Associate Professor,
Department of Information Technology,
Institute of Technology,
Nirma University, Ahmedabad.

Dr.Sharada Veliveti
Associate Professor,
Coordinator M.Tech - CSE(INS)
Institute of Technology,
Nirma University, Ahmedabad

Dr.Sanjay Garg
Professor and Head,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr.Alka Mahajan
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Dimpal Shah**, **16MCEI20**, give undertaking that the Major Project entitled "**Network Traffic classification and Abnormal Behavior Detection using Deep Learning**" submitted by me, towards the fulfillment of the requirements for the degree of Master of Technology in **Computer Science and Engineering (Information and Network Security)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student

Date:

Place:

Endorsed by

Dr. Ankit Thakkar

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Dr. Ankit Thakkar**, Associate Professor, Department of Information Technology, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Dimpal Shah

16MCEI20

Abstract

Today every developing country is trying to become digital. This digitization makes an increase in usage of the Internet. All business also turned out to be online. Every industry is more over-dependent on the Internet because their business is running on mobile apps, Web application, etc. So business's privacy policy is at risk because of the threat of cyber crimes like identity theft, Denial Of Service (DoS), Phishing Attack, etc. It makes us be attentive to our presence on the Internet. One of the solutions of saving ourselves from being a victim of any cybercrime is Network traffic analysis. Network traffic analysis is the process of classification network packets into two categories, normal and attack. Here We are making a survey of different techniques for network traffic classification and discussed different learning approaches based on normal network traffic behavior of users. We also discuss procedures to detect abnormal behavior of traffic data by Machine Learning (ML) techniques. We are proposing a solution for attack detection using deep learning method. In our proposed solution, we are using a genetic algorithm for feature selection and training a neural network using Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) for sequential time-based data classification. We are using the KDDCUP99 dataset which has 41 features and one target column labeled with a name of the attack for our experiments. KDD99CUP has Four attack categories: DoS, Prob, User to Local (U2L), Remote to User (R2U). We are using 10% data of KDD99CUP dataset for the experiment.

Abbreviations

ML	Machine Learning
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IDS	Intrusion Detection System
SVM	Support Vector Machine
ACL	Access Control List
QoS	Quality of Service
IP	Internet Protocol
P2P	Peer to Peer
VoIP	Voice over IP
ISP	Internet Service Provider
PAYL	Payload Based Intrusion Detection System
IANA	Internet Assigned Numbers Authority
DNS	Domain Name Server
ANN	Artificial Neural Network
RBM	Restricted Boltzmann Machine
DBN	Deep Belief Network
LBNL	Lawrence Berkeley national laboratory
CNN	convolutional neural network

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As in the day of growing Internet usage, every business has grown on to the Internet via the web application, web services, and apps. These all things make the network more unreliable. As the number of user increases, usage of network bandwidth will also increase and the Internet crimes are also increases. For making network reliable and trustful, there is a need for a constant eye on the network traffic. As usage of Internet increase, network traffic is also increasing and also analysis become hard. For to handle large data, we need to classify network in such way that can make easy to analysis the current real-time attack.

As business taking full advantage of Internet, attacker also using Internet to perform a malicious activity using user information. There are many computers related crime reported in past decade. For example, botnet egg-drop attack started form 1993 and increased as per the time. A botnet is the group of people who all perform the malicious activities as per the instruction of the botmaster. The FBI Internet Crime Complaint Center reported 269,422 incidents of cyber attacks in 2014, with a total estimated loss of $800 million (FBI, 2015). The Verizons 2015 Data Breach Investigations Report shows that almost 80,000 security incidents were discovered by 70 organizations around the world in 2014, causing them an estimated financial loss of $400 million (Verizon,2015).

Intrusion Detection System helps to analyze network traffic and also helps in detection of attack. For handling network security risk and provides QoS to users on the Internet, network traffic classification techniques are used. Machine learning is a more powerful technique for collecting individual user's behavior and analyzed it . There are three types of network traffic: sensitive, best-efforts, and undesired. For traffic profil-

ing, classification algorithms like k-means, support vector machine, fuzzy base k-means clustering algorithm, and classifiers are used. As network classified based on the normal traffic, different intrusion detection systems are proposed for detecting abnormal behavior on to the real network traffic. Intrusion detection system is nothing but a measure of the deviation of the behavior into the traffic. As there are many type of attacks and they are changing constantly, there is a need for a system that can detect attacks whose signature is not known. For to identify unknown attack, here we are proposing intrusion detection system with good accuracy and has precision in detection of attack.

A brief discussion on different method for classification and intrusion detection system are in chapter 2.

Chapter 2

Classification of Network Traffic

2.1 Importance of Network Traffic Classification

Raising usage of Internet, also raise security threat and cybercrime. To protect users from not to become victim of the any attack from the attacker, we need solution for it. Classification is one of the essential technique to deal with huge data which is a mixture of traffic flow network. There are three major categories of network traffic including sensitive, best-efforts and undesired. Sensitive traffic delivers data on time like an online game, video chat, etc. Best efforts don't have any data loss like an email, peer-peer. Undesired data is spam mail, malicious attack, etc. There are two traditional methods that are mainly used for packet classification in [4]. Port-based classification is using ports for classification. The disadvantage of port-based technique is the use of a dynamic port in application. The payload-based method is using payload information for classification. It checks for signature to be up to date. so it overcomes port based limitation but it increasing processing time. So one new method found by researchers is statistically based that taking packet arrival time for classification. Network traffic can classify as normal and malicious (i.e abnormal). For classification, there are different parameters to be considered like port, IP address, payload size, packet arrival time, etc. Based on this parameter they are assigned to the relevant class.

2.2 Traffic classification and the dawn of Machine Learning

This section explains machine learning concepts and its application into the traffic classification. Before machine learning, Traffic classification was done using TCP and UDP port numbers and payload data inspection, but this two methods also have their limitation and to overcome this limitation machine learning technique becomes useful.

2.2.1 Deep Packet Inspection (DPI) Based traffic classification and its Limitation

There are mainly two methods, port-based and packet-based are used traditionally for classification. But they have their own limitation that is discussed in this section.

Port Based Classification

TCP and UDP are two most used ports in the application. Classifier sets in the middle of the application and observes TCP SYN packets to getting information about new client-side TCP connection. Then it checks target port number with Internet Assigned Numbers Authority (IANA) for to check port is registered.

The first limitation is many application uses client ports that are into the range of registered port numbers. Some ports are not registered with IANA. Example of it is peer to peer application like Napster and Kazaa. The application can also run on other ports rather than well-known ports. Sometimes server ports are dynamically assigned. Some application uses HTTP port 80 but firewall does not filter 80 port traffic [5]. IP over HTTP allow tunneling of application. The single port number assigned to the single application for different QoS requirement.

Payload based Classification

To overcome the limitation of port-based classification, researcher introduced payload-based classification method. Many application uses state reconstruction of the session using packet content. A Port and payload both method is used in [6]. The process starts with the identification of flow's port number if it does not match then it match protocol of the packet. If it fails to identify then it matches with the first KByte of the packet examine. Remaining flow can be classified using the entire payload. But the limitation

of this method is its complexity and load of processing on to the classification device. Classification of encrypted data is difficult by payload based method. A direct analysis of application may violate privacy policy of organization.

2.2.2 Statistical characteristics used for traffic classification

Port and payload based techniques are deep packet inspection techniques and their advantages and disadvantages discussed in the above section. To overcome the limitation of these two methods, a statistical characteristic like idle flow time, distribution of flow duration, inter-arrival time and packet length are used for identification of application. Traffic sampling and profiling using statistics properties using the flow of traffic in [4]. A flow based statistical method need information about flow. There are three types of flow uni-direction, bidirectional, and full directional. All Packets have the same source and destination IP address and ports. Bidirectional flow is a pair of unidirectional flow which direction is opposite to each other. Full directional flow is bi-directional flow that is collected during the whole lifetime. A large number of datasets, multidimensional space flows, and more number of packet attributes in traffic data is the reason for to use Machine Learning techniques into the classification.

2.3 Traffic Classification using Machine Learning Technique

ML is the useful technique to find out the pattern from the data. Applications of ML are search engine, Data mining, medical diagnosis, load forecasting and so on. History of Machine Learning Techniques for the network traffic classification is given in below table 2.1.

2.3.1 Concepts of Machine Learning

Packet classification traditionally is using port and payload based methods. The port-based classification unable to handle dynamic port. Machine learning techniques overcome the limitation of these two methods. Research is going on for traffic classification using ML. Some of machine learning techniques are discussed in this section. Most of the machine learning technique for traffic classification is based on supervised learning and unsupervised learning. Supervised learning is known as classification ML Technique and

Paper Title	Year	Brief Description
Netman: a learning network traffic controller [7]	1990	It is implemented for to maximize call completion in circuit switched telecommunication network
Artificial Intelligence and Intrusion Detection Current and Future Directions [8]	1994	Intrusion detection using AI machine learning technique first startup into the area of network intrusion detection
A Statistical Method for Profiling Network Traffic [9]	1999	Two clustering methods used to make a group of similar activity for the user and helps to detect abnormal behaviour
Profiling Internet backbone traffic: Behavior Models and Applications [10]	2005	Prevent from the cyber attacks data mining and information theoretical technique use for discovering the pattern and abnormal behaviour
Automated Traffic Classification and Application Identification using Machine Learning[11]	2005	Auto class machine learning method used as a limitation of port and payload based classification. The unsupervised Bayesian classifier is also used for the first classification of data
Network Traffic Classification Using K-means Clustering[12]	2007	unsupervised k-means clustering algorithm used with accuracy 80 % overall
Machine learning based encrypted traffic classification: Identifying SSH and Skype[13]	2009	for to make classification ssh and skype application compare five machine learning algorithms, AdaBoost, naive Bayes, SVM, Ripper, and c4.5 make a conclusion as c4.5 is better among all five
Real-Time Traffic Classification Based on Statistical and Payload Content Features[14]	2010	As disadvantage of port and payload based method in this author uses statistical and payload based features for to classify data trace MAWI and this works with HTTP and FTP more accurate than with DNS

Table 2.1: History: Machine Learning Technique as a solution for Network Traffic Classification

unsupervised learning is known as clustering ML Technique. There is one more approach that is hybrid in which both supervised and unsupervised learning approaches are used. Machine learning traffic classification has two main categories that are probabilistic and deterministic. Deterministic method classify classes based on the distance calculation. Probabilistic classification uses the probability value to categories all respective classes. Assignment of the class is done based on the largest probability value [**?**].

2.3.2 Supervised Learning Approaches

Labeled Data is used in supervised learning approach makes classification easy. There are many supervised learning techniques available that are discussed in this section. Supervised learning defined classes for all predefined traffic flow. When new instance comes, it maps with predefined class.

- **Naive Bayes**

  Naive Bayes is classification technique that is based on Bayes' theorem and it is a probabilistic classifier. This is one of the scalable classification method. Bayesian Probability can be described by following equation 2.1 [15].

$$posterior = \frac{prior \times liklihood}{evidence} \tag{2.1}$$

  In most of the example, the denominator is constant. In this method, one assumption is made that all features are independent of each other. A Prior for the given class can be defined as a number of samples for the given class from the total number of sample into data. In [16] authors used naive Bayes classification technique for classification of application on-site institution full duplex researcher data. Data contains both directions traffic during 24-hour on full duplex gigabytes Ethernet link. Authors focused on to the features like Flow duration, TCP port, payload size, Effective bandwidth and packet inter-arrival time. Two methods, Fast correlation-based filter and kernel density based estimator is used for classification. A kernel density based method gives around 93 percentage of accuracy. Dataset first pre-filtered using FCBF. That gave around 94 % accuracy. So they used method Naive Bayes with FCBF and kernel density estimator that gives the highest accuracy. In [17] authors Proposed new schema for traffic classification using ISP dataset that gives the best result with a small amount of training unidirectional

flow data. In this method, authors used flow base information correlation data for the prediction of the class as explain into the naive Bayes theory. Classification is done based on the maximum prior correlation value. Experiments perform on the two real-time datasets and it gives high performance in less time. Best first search method used for the feature selection.

In [18] authors used dataset from University of Queen Mary repository and from around 266 attribute applying feature selection methods, they got 8 attributes for the classification. Wrapper and filters methods are used for feature selection. In [19] authors used Gaussian naive Bayes technique and get 100% accuracy after so many iterations. Proposed technique have steps: Capture, Analyzing, Extraction, Preprocessing and classifier Gaussian Naive Bayes to predict DDoS Attack. An Experiment performed into the hmad Dahlan University Networking Laboratory for the 60 minutes slots.

- **Decision Tree**

  A decision tree is a graphed structure in which node represents an attribute of the dataset, Branch represents the condition of the result: true or false and each leaf node represents the class. The decision tree is easy to understand, allowing to work with a different scenario, and gives results: worst, best and expected. It can be use with other techniques also. In [20] author used different supervised learning method for to detection of the botnet packet. But decision tree has more accuracy than other algorithms. Authors performed experiments on to ISOP dataset which has both malicious and non-malicious and measure accuracy with four parameters precision, recall, F-measure and correlation coefficient. Authors used flow-based detection method for detection of the botnet and got the best result for C4.5, RFTree, and RTree.

  In [21] authors used decision tree with practical swarm optimization (PSO) for to remove false positive and detection of spam. Classification of the attributes is done using normalization of information gain. Feature selection method MBPSO and the decision tree c4.5 algorithm are used to analyzed result. A measure of

result parameters is sensitivity, specificity, and accuracy. Random forest is one of the tree-based machine learning algorithms which is a group of individual decision trees. In [22] authors used different machine learning approaches for capturing c&c (Command and Control) session for detection of DDOS and botnet attack. 55 feature vectors are used for experiments. Random forest algorithm is best for high dimensional data. In this paper, the author compares the result with naive Bayes, K- nearest and SVM algorithm with random forest. Accuracy of the experiment is calculated using 10 fold cross method.

C4.5 Decision tree has the ability to classify the discrete or continuous type of data. It's starts from the top and goes up to the class label [23].

- **Genetic Algorithm**

  A genetic algorithm is a biological structure which has cells to build block. The Genetic algorithm also has chromosomes in binary form 0's and 1's. Chromosomes are nothing but DNAs. A genetic algorithm has several phases like initialization, fitness test, selection, crossover, and mutation. So finally we can say that genetic algorithm is one of the optimization technique which find out input and find optimized output. The main application of genetic algorithm is feature selection. A Genetic algorithm defined rules for identifying malicious behavior into the network in [24]. Here authors concentrate only TCP/IP connection and used penalty fitness formula for accuracy measure.

- **SVM: Support Vector Machine**

  SVM is one of the machine learning algorithms which we can use in both classification and regression. Hyper-plane is used to classify features and select hyper plane which correctly segregate data in two classes. Selection of hyper plane is based on the margin between nearest point. SVM is of two types: linear and non-linear. In non-Linear SVM, first function is apply to convert point into the linear and then the process is similar as linear SVM. The bad hyperplane selection gives noisy classification. In [25] authors used SVM for TCP network classification to solve the multi-class problem of SVM. Classification process follows steps. First, collect traffic and represent it into the flow, take TCP flow with bidirectional that observed port number and three-way handshaking of packets that are not in sequence. Train-

ing phase takes application payload and apply payload pattern matching method for classification according to coordinate surface matching if matched than assign to that class else assigned to the unknown class.

Parallel SVM for network traffic classification to handle big data problem is used in [26]. MapReduce open source programming model used for big data storage and processing. Training phase divides data set into multiple subsets of data and then load it with corresponding mapper nodes. Hadoop is used for implementation and labeled flow as unique ID.

2.3.3 Unsupervised Learning Approaches

Unsupervised learning is also known as clustering in which similar types of attributes are collected in one group called cluster. An attributes in one cluster that are dissimilar to other. There are many clustering algorithms like K-means, Fuzzy c-means, etc. The main component for clustering algorithms is distance measure. Euclidean distance is distance function used by many researchers for clustering. Distance for high dimensional data is calculated using minkwoski distance method. Unsupervised methods for network traffic classification are explained in this section.

- **K-means Clustering**

  K-means clustering algorithm used statistical information to build classifier. Clusters are build using similarity measures, distance calculation like a Euclidean algorithm. Euclidean distance calculate using equation 2.2 [12]. K-means clustering algorithm creates spherical clusters in shape.

$$D(a,b) = (\sum_{i=1}^{n}(a_i - b_i)^2)^{1/2} \tag{2.2}$$

  If distance nearest to two clusters than optimal solution can be derive using error minimization that is defined in equation 2.3 [12].

$$Error = \sum_{i=1}^{n}\sum_{j=1}^{m}|dis(a_i, b_i)|^2 \tag{2.3}$$

  Mean square error can calculate using distance error and cluster center. K-means clustering followed by a two-steps procedure. first, calculate mean center for all data

and than reassignment of the cluster by new cluster. K-means clustering method is partition based.

k-means for network traffic classification used in [27]. The experiments for classification using log transformation data and original data monitor of 1000 users in research lab and using k=20 cluster to 200 clusters, gets 80% accuracy and after applying log transformation get 10% more that is 90% accuracy. Online and offline data classification is done in [28] on Auckland IV and Calgary dataset. Classes considered for the Auckland IV datasets are DNS, FTP (control), FTP (data), HTTP, IRC, LIMEWIRE, NNTP, POP3, and SOCKS. In [29] k-means clustering and expectation maximization algorithm performances are compared using accuracy as a measurement parameter for differentiating both algorithms. For feature extraction, authors used correlation-based feature selection. It selects attribute: Flow duration, Packet length, Inter-arrival time, and Total number of the packet in the flow for the classification.

- **DBSCAN**

  A DBSCAN is a density-based algorithm which groups points that are very nearby to their neighbors and also mark outliers which are alone in the low-density region. DBSCAN algorithm works in steps. First, calculates neighbor points and finds core point from the min points neighbors. The second step is collect all connected components to the core point and ignore all noncore points, and final step is to assign all noncore point to neighbor point or assign them as noise.

  In [30] DBSCAN is used for abnormality detection using two parameters epsilon and min points. A K-nearest neighbor algorithm used for calculation of two parameters. Epsilon gives radius between two points and min points gives the minimum number of points required to define the group as a cluster. Experiment dataset includes remote and local computers. In [31] authors establish netstream that produce one direction network flow which generates a network with same properties i.e source address, destination IP, source port, destination port, and protocol. DBSCAN is used for clustering normal and abnormal behavior. In the lab, by using netstream dataset collected and extracted features using feature extraction. DBSCAN algorithm used for clustering and performed DoS attack for abnormal behavior de-

tection. The result is compared with k-means clustering according to false alarm rate. k-means method gives more false alarm rate than DBSCAN.

- **Autoclass**

  Autoclass is clustering algorithm which based on the Bayesian method and useful for finding the optimal solution for a large dataset. Theory of Bayesian classification is to find the best class that predict data in given modal space. This is a probabilistic algorithm. Autoclass calculates likelihood of instances and according to that calculates weights of all instances. Autoclass can handle missing data. This method allows selecting clustering automatically. Autoclass uses Expectation Maximization(EM) algorithm. In that, there are two steps. First, finds parameters for the cluster and maximization step, find mean and variance to re-estimate parameters till it converts into the local maximum.

2.3.4 Hybrid Approaches

Hybrid machine learning approach is most useful in network traffic classification in which classification and clustering both techniques are used for detection of abnormality. Fuzzy logic and genetic algorithm are used in [1] for network abnormal behavior detection which has one flow exporter that collects flow and processed packets. Classification is done as per the features like source and destination port and IP, packets per second, and bits per second. Processed packet becomes input for the fuzzy logic that uses the Gaussian function. After fuzzification, it defuzzifiy packets and if needed alarm raise based on the given thresold value. Figure **??** shows the structure of fuzzy logic proposed system. The digital signature of network segments flow analysis (DSNSF) helps for creating a network profile in [1].

To overcome supervised approach of SVM, Researcher in [32] proposed enhanced SVM method that uses a Genetic algorithm and TCP/IP fingerprinting along with self-organized feature map (SOFM). SOFM is used for user profiling for normal traffic Packet TCP/IP fingerprinting for filtering unknown packets and the Genetic algorithm is used for the choice of the more appropriate packets. Sliding window concept is used for identification packet for a particular connection. The result of the proposed system is compared with supervised and unsupervised SVM method. In [33] SVM-CART algorithm used for abnormal behavior detection. First linear discriminant applied for reduction of di-

mensionality of the dataset and also make a selection of the best features. Classification algorithm assigned more weight to nearer point and CART gives more weight to a distant point in proposed SVM method. Accuracy is compared with KNN algorithm.

In [34] new method is defined for detection of intrusion that is cluster center and nearest neighbor. First, K-means clustering algorithm used for extracting center of each predefined class and find the nearest neighbor for all cluster. It creates new data set with one dimension that is calculating a distance. The result is compared with SVM, CANN, and KNN. But CANN gives good result compared to KNN. The advantage of CANN is it taking less computational time and it fails to recognize U2L and R2L attack.

In [35] Decision tree and one-class SVM used to build a misuse based anomaly detection model. C4.5 used as decision tree algorithm which decomposed data into regions and labeled that data as a class. One class SVM is one of the best technique for detection of outliers. In [35], proposed system used decision tree algorithm for extraction of feature and based on the extraction of features if attack is known then raise alarm else again go to the one class SVM step that contains leaf node subset. One class SVM verifies attack is there or not and if it is there then it raises a notification to admin. The decision tree has 99% accuracy for detection of known attack but, in author's proposed system one-class SVM added to identify the unknown attack and reduce false alarm rate.

2.4 Deep learning: A Machine learning Technique for network traffic classification

Deep learning is the more powerful and flexible machine learning algorithm which creates a nested hierarchy of concepts with concepts defined in relation to a simple concept. To understand the concept of deep learning lets take one example, identification square shape. First step is to check if the shape has four lines or not and then check for other properties of a square shape that is all lines are in same length, perpendicular, etc. To identify square, create a simple task from the complex large task. This is the way deep learning works. In this section, we are explaining how deep learning is useful than other machine learning techniques and how its used as network traffic classification.

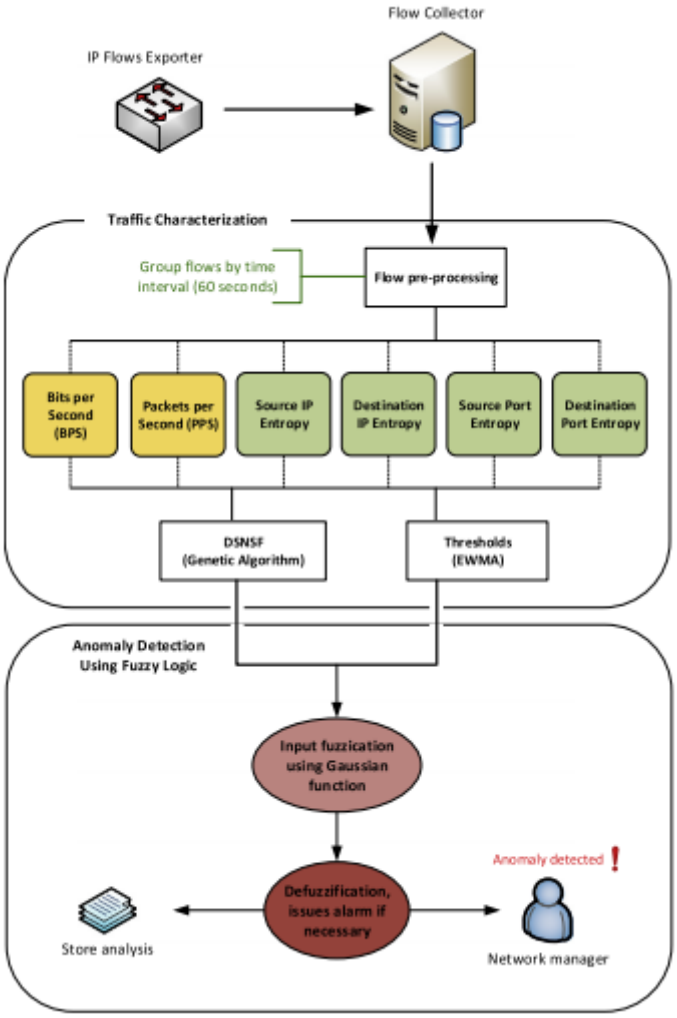


Figure 2.1: System Architecture for anomaly detection using Fuzzy Logic and Genetic algorithm [1]

2.4.1 Benefits of Deep learning Over Other Machine Learning Techniques

The following points proves that deep learning is better than other machine learning Techniques.

- Data Dependencies

  Most of the machine learning algorithm performance measure on the scale of data. Depp learning algorithm is not well with small data. So deep learning can easily handle large-scale data than other machine learning algorithm.

- Hardware Dependencies

  Deep learning algorithm works on the high-end machine where others can work on the low-end machine. For example, GPU which can perform a large amount of matrix manipulation.

- Feature engineering

  Deep learning has in build mechanism to learn high-level features from the data. In other machine learning algorithm, this is time-consuming and complex work that is done by manual and coded feature extraction algorithm.

- Problem Solving Approach

  Traditional Machine learning Techniques breaks the problem into several parts and solve them and combine them again. Deep learning solves problem end to end.

- Execution Time

  Deep learning takes a long time to train but less time to run testing where other machine learning takes more time to validate.

- Interpretability

  Deep learning cannot be interpreted easily but other machine learning technique can interpret easily.

2.4.2 Deep learning for network traffic classification

Deep Learning creates a nested hierarchy of concepts with concepts that are defined in relation to the simple concepts. Deep learning architecture has three types of layers.

The first layer is Input layer, second hidden layer and at last output layer. A number of hidden layers depend on the number of features (Attributes in given Dataset). There is a number of bias that are used in the calculation of weights in the hidden layer. Deep learning has many advantages with respect to ML techniques. Deep learning works with large data. It has an inbuilt mechanism to learn high-level features from the data. It provides end to end solution. It takes time for learning but it gives the fast result at the time of validation. Deep Learning is also named as the neural network because it is one kind of network that is made of many layers.

There are many deep learning based machine learning algorithms like Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Adaboost, autoencoder, etc. But the main concept of deep learning is its self-learning. A brief discussion on different deep learning techniques in network traffic classification is done in further part of this section.

- **Convolutional Neural Network (CNN)**

  A convolutional neural network is the network of neurons with respect to bias and weights. It has one 3D layer that takes input as 3D and gives output in 3D visualization. It has layers like input layer, convolutional layer, an activation layer, pull layer and fully connected layer. In [36] authors proposed one-dimensional convolution network for an end to end encrypted traffic. This solution is applied to the ISCX-VPN dataset. 1D-CNN makes tarffic classification easy. The malware traffic classification method is implemented using a convolutional neural network that taking traffic input as an image in [37]. This is the first trial of traffic classification using raw data. Total 8 experiment was done on data and concluded that session is better with respect to flow. In [37], the author has used 3 types of CNN classifiers in two scenarios. The average accuracy of classifiers is around 100%.

- **Recurrent Neural Network (RNN)**

  It is the type of ANN which allows execution of time sequence with dynamic temporal behavior. It has its own memory for processing inputs. Mainly two types of RNN are present Long Short-Term Memory (LSTM) and gated recurrent unit. In [38] authors used LSTM RNN model to train intrusion detection system. To detect a multichannel attack, in [39] authors have used LSTM-RNN method. The multi-feature extraction method was used for feature extraction. RNN is also used

as cyber-physical system's attack detection [40]. In [38] authors used LSTM-RNN method as IDS on KDD99CUP dataset. LSTM-RNN is the best algorithm for sequential data training. In [41], researchers worked on different optimizer and found that LSTM-RNN performed best using Nadam optimizer and gives 97.54% accuracy. In [42], authors identified parameters for LSTM-RNN that are best performed on the KDD99CUP dataset. They gave learning rate 0.1 and 1000 epochs to LSTM-RNN model and got 93.82% accuracy. In [43], authors used Deep Recurrent neural network (DRNN) for identifying user behavior into the tor network and proposed a system with tor server and client and sniff traffic using Wireshark network analyzer tool. It has good user prediction ability. In [44], authors used Deep Neural Network (DNN) based method for identifying flow based issue for Software Defined Networking (SDN) network which is more popular and good solution for future Internet communication.

In [2], researcher proposed forensics evidence creating a system which using RNN that is useful for identifying attack into the computer system. It reduces cost and time of forensics process. In [45], authors proposed anomaly-based IDS using the autoencoder and Restricted Boltzmann Machine (RBM) deep learning method on the KDD99CUP dataset. In RBM, one hidden layer perform feature reduction clustering and their weights are passing to other RBM hidden layer for to create deep belief network (DBN).

In [39], authors proposed multi-channel attack detection system using LSTM-RNN on KDD99CUP dataset. They used voted algorithm for identify an attack. There are three main categories for features: basic, Content and flow based. Three channels with three different kinds of the dataset are given to LSTM-RNN and then it passed to Logistic regression for the final result of attack detection. The outcome is measured by accuracy and detection rate that is compared with different neural network based algorithm result. It gives 98.94% accuracy after training testing using the proposed system.

- **Restricted Boltzmann (RBM) and Autoencoder**

  RBM is the stochastic approach which has a stochastic unit with distribution. Numbers of layers are present between hidden layer and input/output layers. For

minimizing reconstruction error, adjusting of weights are done by RBM. Autoencoder is an architecture which has two parts, first encoder, and a second decoder. These two are adjusted between the input to hidden layer and hidden to output layer respectively. In [46], authors used these two methods for network intrusion detection.

2.5 Datasets: Available for Network traffic classification and Abnormal Behavior detection

Many researchers are doing work on finding a deviation in the behavior of the users from its normal behavior. It is hard for a researcher to work on dynamically changing dataset and learning any machine learning model with unlabeled data. Availability of well processed and labeled datasets are so rare. This section gives information about publicly available dataset for network traffic classification and abnormal behavior detection.

- **CAIDA Dataset**

  CAIDA is an Internet trace dataset which contains active and passive both measurement. Active measurement of the Internet is done the macroscopic project by performing active probing on IPv4 and IPv6 Internet. Passive measurement is done on a network like academic, nonprofit and commercial infrastructure by monitoring specific link, IP address, etc [47].

- **Lawrence Berkeley National Laboratory (LBNL) and ICSI Dataset** It is a web repository of packets which are collected from the activity of more than 100 hours and using 1000 of several internal hosts. It has many dimensions [48].

- **DARPA Dataset**

  Defense advanced research project agency is named as DARPA. This dataset has probability measurement of false alarm rate and detection rate. It has data of years 1998 and 1999. The main advantage of this dataset is that it has different data types that are used for many intrusion detection systems with eliminating privacy concern. 2000 DARPA dataset is available at [49].

- **KDD cup 1999 Dataset**

  KDD dataset name comes from knowledge discovery and data mining. It has data

from both attack "BAD" and normal" GOOD" connections. Data audited in this dataset mostly simulated on military network environment [50].

- **Internet Traffic Archive Dataset** This data set is given by ACM Sigcomm. This dataset is used in the study of network dynamics, patterns of growth and characteristic of usage. On this archive, users can't perform traffic analysis because of it containing privacy details [51].

- **ISCX Dataset**

  ISCX named from Information security center of excellence. Dataset has full packet payload in pcap file format. ISCX contains a number of datasets named as ISCX UNB Dataset, ISCX NSL-KDD dataset, ISCX VPN-VPN traffic dataset, ISCX Botnet dataset, ISCX Tor-monitor dataset and ISCXFlowmeter for the researcher. Dataset repository present at Canadian Institute of cybersecurity. This dataset contains 7 days normal and malicious activities [52].

- **Kyoto University's Honeypots Dataset**

  Kyoto dataset has traffic data that are generated with sanitized IP address using Bro IDS tool on honeypot [53].

- **MAWI Working Group Traffic Archive**

  This dataset is storage of traffic data from the WIDE project. It is created on the daily traces using transit link of WIDE (upstream of ISP) [54]. This dataset provides daily updated traffic as per the new application and anomalies. Dataset has labeled traffic data.

- **Queen Mary Research Online Repository**

  This Repository provides meta storage of dataset for a researcher with partial access [55].

- **UNIBS Dataset**

  UNIBS has two datasets UNIBS 2009 and ssh tunnel dataset. These dataset traces are taken from edge routers of University of Brescia in the period of continuous three working days. This has 27 GB data consist of both TCP and UDP traffic. Data includes web, mail, BitTorrent, Skype (TCP & UDP) and other. It contains around 79000 flows. The whole data trace is done on GT client daemon [56].

- **Auckland IV Dataset**

  The dataset contains Synchronized GPS IP header taking from DAG3 card at Auckland University. This dataset contains TCP, UDP, and ICMP traffic excluding non-IP traffic. For the processing of the trace, it suggests Libtrace tool [57].

- **CALGARY Dataset**

  Data is traced by monitoring Calgary university Internet link. This data packets are with the payload. It has 60 GB data in duration of 1 hours [22].

2.6 Traffic Classification Evaluation matrices

For to make difference between traffic classification technique, evaluation matrix is a method for predict accuracy. Calculation of accuracy needs number of matrices that explained in below section.

Let's take traffic class C in which we are interested. It is collected from the set of the huge traffic class. Instances is part of Class C or not is identified by classification. Its accuracy can be calculated using the following characteristic.

- **False Negatives(FN)**

  Classification of class C as not part of a class recognized as incorrectly. Meaning of this condition the situation is not present really but by classification, it is showing the presence of it [34].

$$FalseNegativeRate = \frac{FalseNegative}{TruePositive + FalseNegative} \tag{2.4}$$

- **False Positive(FP)**

  Classification of class C as part of traffic class recognized incorrectly. It means that the given condition is there but actually it does not present [34].

$$FalsePositiveRate = \frac{FalsePositive}{FalsePositive + TrueNegative} \tag{2.5}$$

- **True Positive(TP)**

  Classification of class C as Part of class recognized correctly. The data that are

correctly classify by classifier [34].

$$TruePositiveRate = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2.6}$$

- **True Negative(TN)**

  Classification of class C as not part of class recognized correctly. The data that are not correctly classy by classifier [34].

$$TrueNegativeRate = \frac{TrueNegative}{FalsePositive + TrueNegative} \tag{2.7}$$

Traffic classification also has two most recent matrices for evaluation. Flow accuracy and Byte accuracy. Flow Accuracy is the percentage of flow classified correctly from given dataset and Byte accuracy measured numbers of bytes from packets, classified flow correctly. Mean square error is also traffic classification measure which is square of the difference between actual and predicted value. It is defined by following equation 2.8 [18].

$$MeanSquareError = \frac{1}{N}\sum_{i=1}^{N}(ActualValue - PredictedValue)^2 \tag{2.8}$$

The major parameter that is considered for the neural network is model training and testing accuracy. Accuracy is defined by the following equation 2.9 [18].

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative} * 100 \tag{2.9}$$

Table 2.2: Survey Table

Paper Title	**Method used**	**Dataset used**	**Total Number of Features in Dataset**	**Number of Features used**	**K- fold cross Validation (Y/N)**	**Accuracy**
Internet Traffic Classification Using Bayesian Analysis Techniques [**?**]	FCBF+NB kernal Estimator	High performance Network Monitor Dataset [58]	6	6	N	94 %
Internet Traffic Classification by Aggregating Correlated Naive Bayes Predictions [17]	Bag of Flow(BoF)-NB	ISP [59] , Wide [54]	20	12	N	Not Available
A Comparative Performance Analysis on Network Traffic classification using Supervised learning algorithms [18]	Naive Bayes	University of Queen Mary repository DataSet [55]	266	8	N	94.81 %
Continued on next pages						

Paper Title	Method used	Dataset Used	Total Number of Features used	Number of Features Used	K- fold cross Validation (Y/N)	Accuracy
DDoS Attacks Classification using Numeric Attribute-based Gaussian Naive Bayes [19]	Gaussian Naive Bayes	Ahmad Dahlan University Networking Laboratory (ADUNL) [19]	2	2	N	100%
An efficient Flow-based Botnet Detection using Supervised ML [20]	supervised learning methods, c4.5 decision tree,RFTree,RTree	ISOP data [60]	Not Available	39 for each flow	N	95%
Binary particle swarm optimization (PSO) with mutation Operator for feature Selection using decision tree applied to Spam Detection [21]	DT (C 4.5) with MBPSO feature selection method	6000 users email data [61]	57	3	Y	94.27%

Continued on next pages

Paper Title	Method used	Dataset Used	Total Number of Features used	Number of Features Used	K- fold cross Validation (Y/N)	Accuracy
Command & Control (C&C) Session Detection Using Random Forest(RF) [22]	Random Forest	CCC Dataset, Practice Dataset [62]	Not Available	55	Y	99%
Network Intrusion Detection System Using J48 Decision Tree [63]	Decision Tree J48	Kyoto 2006 Data [53]	24	7	Y	97.23%
Network Traffic Classification - A Comparative study of two common decision tree (DT) Methods: C4.5 & Random Forest (RF) [23]	C4.5 and Random Forest	network traffic Dataset [16]	248	Not Available	N	99.67 % (C4.5) and 98.64 % (Random forest)
Using Genetic Algorithm for Network Intrusion Detection [24]	Genetic-Rule based algorithm	DARPA Dataset [64]	9	9	N	Not Available

Continued on next pages

Paper Title	**Method used**	**Dataset Used**	**Total Number of Features used**	**Number of Features Used**	**K- fold cross Validation (Y/N)**	**Accuracy**
Support Vector Machinces for TCP traffic classification [25]	SVM-multi stage	UNIBS [56], LBNL [**?**], CIADA [47]	Not Available	Not Available	N	90%
Scalable Network Traffic Classification Using Distributed Support Vector Machines [26]	Distributed SVM	CBA Lab dataset [65]	Not Available	5	N	93.133%
Network Traffic Classification Using K-means Clustering [27]	K-means Clustering	High performing Network Monitoring Dataset [58]	Not Available	11	N	90%
Traffic Classification Using Clustering Algorithms [28]	K-means Clustering	Auckland IV and Calagary [57]	Not Available	Not Available	N	80% for k=500

Continued on next pages

Paper Title	Method used	Dataset Used	Total Number of Features used	Number of Features Used	K- fold cross Validation (Y/N)	Accuracy
performance analysis of unsupervised machine learning techniques for network traffic classification [29]	K-means clustering and Expectation Maximization	Network Trace Data [29]	Not Available	5	N	65%
Intrusion Detection Using Clustering of Network Traffic Flows [30]	DBSCAN	Stanford University Dataset [66]	4	4	N	Not Available
Traffic Features Extraction and Clustering Analysis for Abnormal Behavior Detection [31]	DBSCAN compare with K-means	CAIDA [47]	Not Available	5	N	80.9%
Network Anomaly Detection System using Genetic Algorithm and Fuzzy Logic [1]	Fuzzy logic and Genetic algorithm	Data of State University of Londrina using the sFlow protocol [1]	Not Available	4	N	96.53%

Continued on next pages

Paper Title	Method used	Dataset Used	Total Number of Features used	Number of Features Used	K- fold cross Validation (Y/N)	Accuracy
A hybrid machine learning approach to network anomaly detection [32]	Enhanced SVM method with self organized feature map, TCP/IP packet fingerprinting and Genetic algorithm (GA)	MIT Lincoln Lab dataset [49]	41	11	N	87.74%
A novel technique for intrusion detection sytem for network security using hybrid SVM-CART [33]	SVM-CART	KDD99CUP [50]	41	41	N	95%
CANN:An intrusion detection system based on combining cluster centers and nearest neighbors [34]	KNN and center cluster SVM	DARPA 1999 and KDD Dataset [50] [49]	41	6 and 19	N	99.56%

Continued on next pages

Paper Title	Method used	Dataset Used	Total Number of Features used	Number of Features Used	K- fold cross Validation (Y/N)	Accuracy
A novel hybrid intrusion detection method integrating anomaly detection with misuse detection [35]	Decision Tree (DT) and 1-class SVM	NSL-KDD [50]	41	Not Available	N	87%
Deep Learning based Multi-channel intelligent attack detection for Data Security [39]	LSTM-RNN (Long Short Term Memory Recurrent Neural Network)	NSL-KDD [50]	41	3	N	98.94%
Malware Traffic Classification Using Convolutional Neural Network for Representation Learning [37]	Convolutional neural network (CNN)	USTC-TFC2016 [37]	20	4	N	99.41%

Continued on next pages

Paper Title	**Method used**	**Dataset Used**	**Total Number of Features used**	**Number of Features Used**	**K- fold cross Validation (Y/N)**	**Accuracy**
End-to-end Encrypted Traffic Classification with One-dimensional Convolution Neural Networks [36]	1-class CNN	ISCX VPN-non VPN Dataset [52]	12	2	N	85.8% and 92% respectively
An Evolutionary General Regression Neural Network Classifier for Intrusion Detection [67]	E-GRNN (Evolutionary general regression neural network)	UNB ISCX [52]	41	41	N	95.97%
An anomaly-based Network Intrusion Detection System using Deep learning [46]	Restricted Boltzman machine and Autoencoder	KDD99 CUP [50]	41	41	N	Not Available

Chapter 3

Abnormal Behavior Detection

The most important problem in recent days is the use of Internet and increases ratio of cyber crime. For to prevent our system, there must be some method that handles irregular event in system for example, a firewall. System detection and prevention are the techniques to avoid cybercrime. Abnormal behavior is also known as Anomaly Detection. There are two types of intrusion detection system, anomaly based and misused based. Anomaly-based intrusion detection helps to detect new attack that is exactly opposite of signature-based intrusion detection system which detects known attack.

In any kind of intrusion detection or abnormal behavior method, first step is network traffic classification. For network traffic classification, first port based and payload based methods came. But as payload becomes encrypted and application using dynamic ports, these two methods failed to detect attack. So machine learning techniques identified as traffic classification technique. In previous chapter we discussed different machine learning approaches for traffic classification. In machine learning technique, supervised learning approach fails to identify application correctly and also it requires more human efforts to make data labeled before learning. So unsupervised learning approach proposed which does not required labeled data. There is a chance of false positive attack detection using unsupervised ML technique. So researchers have used hybrid approach. Hybrid approach using both supervised and unsupervised techniques. First, apply unsupervised technique, labeled cluster as a class and use the supervised technique for identification of new instances. These have issues with the size of data and performance. Then other and more effective machine learning technique comes that is deep learning approach that has the capability to handle large data with less false positive rate.

Deep learning also have so many methods that are explained in chapter 2. But best of them is a recurrent neural network (RNN). RNN is best algorithm based on its capacity to deal with time-based network changes that are not available in CNN. CNN take raw data as input for the process. But sometimes it fails to detect the attack. CNN do analyze the component of the data, group them, and then recognize the structure or abnormality in the network. RNN is helpful in recognizing pattern timely based. So based on to the survey into chapter 2, we proposed one system architecture that helps to detects an attack from network traffic with good performance rate and accuracy.

3.1 Proposed System

This section is about the architecture of proposed attack detection system. It has a detailed overview of steps that are followed by architecture for attack detection.

3.1.1 Architecture of proposed system

Figure 3.1 shows the proposed system for attack detection. The proposed system has three phases: Data Preprocessing, Training Phase and Attack Detection. Data Preprocessing is done by using one hot encoding and Random Forest regressor methods. Training phase includes two steps: Features extraction and LSTM-RNN learning. Attack detection phase has prerequisite of loading training model that helps to predict traffic type as normal or attack. For feature extraction, we are using a genetic algorithm that is best-performed machine learning algorithm for feature selection.

- **Data preprocessing**

  Network traffic has integer and string both data. So for the further data analysis process, we are applying one hot encoding method that converts string data to binary so that we can use it directly for the neural network training. Further, it is required to transform data from its original format to the LSTM-RNN input format. LSTM-RNN takes an array of 3 dimensions in which first is a number of the data sample, second is a number of features, and third is a time stamp.

- **Feature Extraction**

  Network traffic data has numbers of features but all are not useful for identifying attack or normal traffic. So there is a need for finding a correlation between features and identifying vectors subset. This step takes all traffic data attributes and first

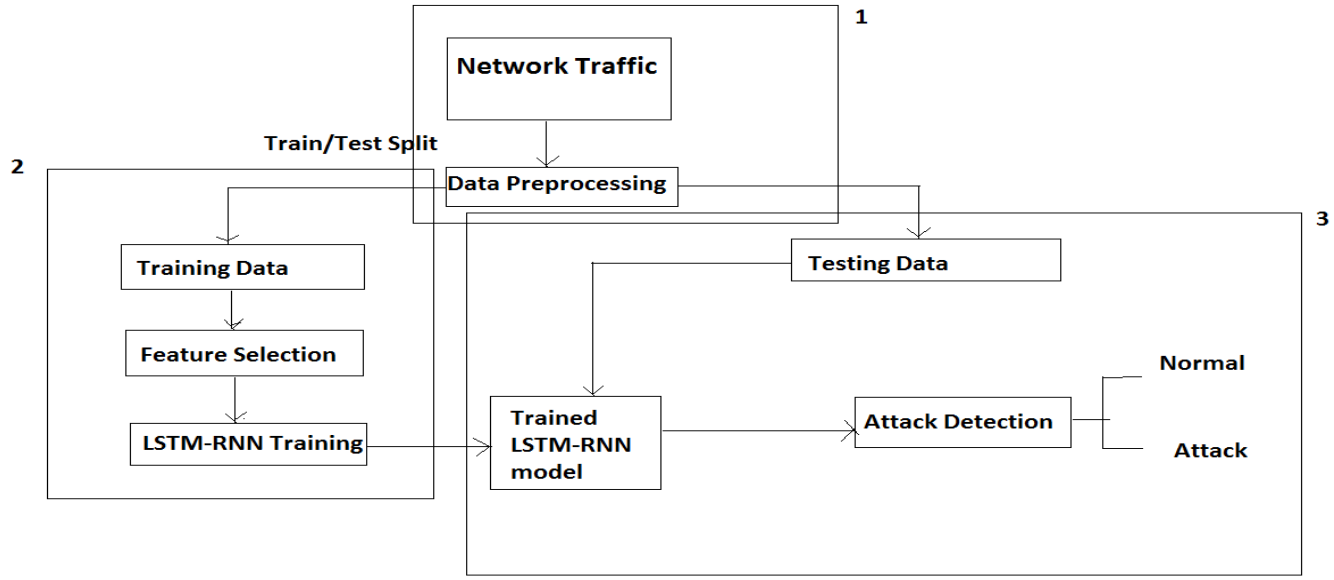


Figure 3.1: Overview of attack detection proposed system

apply correlation algorithm that assigns weights to the features between -1 to 1. So here we apply Random forest regressor [68] which improves the accuracy and protect model from the over-fitting. Our KDD99CUP dataset has 41 features. We want to find some accurate features subset that give good accuracy. After applying this machine learning technique, the remaining feature set is given to Genetic Algorithm (GA) for further dimensionality reduction. GA is best for the feature selection [1].

- **Training model using LSTM-RNN neural network method**

  After Feature extraction, Data is given to LSTM-RNN model shows in figure 3.4 for attack detection. LSTM-RNN model has two LSTM layers, two dense layers, and two dropout layers. Drop out layer dropped or ignored neurons that are responsible for over-fitting. Dense layer performs the linear operation on the data. Finally, activation layer applies a sigmoid function to the data for to predict category of incoming traffic.

- **Attack Detection**

  After Data preprocessing, first apply the random train test split method for splitting data into training and testing. Use training data to fit the LSTM-RNN model. For detection of an attack, load the trained model and give testing data. It produces the output which identify data is normal or attack.

3.1.2 Genetic Algorithm (GA) for Feature selection

In our proposed system for attack detection, there is a need for better accuracy of training model and one way is the identify feature set that is helpful in identifying an attack. GA is used for feature selection in [1]. GA has four steps process, Initialization of population, calculation of fitness, selection, crossover, and mutation. Algorithm 1 shows steps for GA feature selection [69].

Algorithm 1 A standard Genetic Algorithm

```
Population set initialization
while stop condition do
    for chromosomes in population do
        Fitness calculation for chromosomes
    end for
    chromosomes selection for crossover
    crossover
    mutation
    population replacement including chromosomes
    return chromosome with best fitness value
end while
```

It has four component. A population which includes individuals where an individual gives expected solution. The solution given by individuals is good or bad, is decided using fitness function. For new generation creation, selection function takes a good individual from the present population. Crossover and mutation identify new regions for searching. This keeps some current information as it is.

3.1.3 Recurrent Neural Network (RNN) and Long short-term memory (LSTM)

RNN is useful for sequential modeling because, it has cyclic connection like feed forward neural network. Let's assume, input layer as I, hidden layer as H, and output layer as O. Sequence of input is $I = (i_1, i_2...., i_n)$. RNN calculates hidden layer sequence $H = (H_1, H_2...., H_n)$ and Output layer sequence $O = (o_1, o_2...., o_n)$ using equation 3.1 [70] and 3.2 [70] respectively.

$$H_n = \sigma(W_{ih} I_n + W_{hh} h_{n-1} + b_h) \tag{3.1}$$

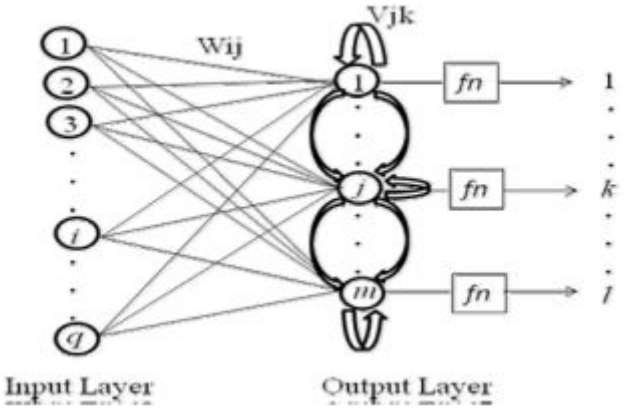


Figure 3.2: Recurrent Neural Network structure [2]

$$o_n = W_{ho} h_n + b_o \tag{3.2}$$

In 3.1 and 3.2 σ indicates activation function, I,h, and o are input, hidden and output layer value, and W,b are weights and bias value respectively.

Figure 3.2 shows the design structure of RNN. The first step is to initialize weights value of the q number of input neurons and also initialize weights from m output neurons to k neurons with its actual values. Then initialize all output neurons with value zero. After than calculation of output is done by using activation function. This process will continue until learning error becomes zero. RNN is using Back Propagation Training Time (BPTT) for variable length input sequence. RNN has problem while training with BPTT [71]. BPTT model takes training data for learning and also save output gradient error value with a time stamp. Sometimes RNN become hard to train because gradient can explode while applying BPTT algorithm to it.

Long Short-term memory (LSTM)

LSTM is the type of RNN which has the capability of long-term dependencies learning [3]. Figure 3.3 shows the architecture of LSTM. Cell state is the key of LSTM. It runs straight for the entire cycle with some changes into the linear interaction. Gates into LSTM has the ability to change or remove data from the cell state. Gates have two component sigmoid activation function and multiplication pairwise operation. Forget layer (sigmoid layer) into the LSTM, takes h_{t-1} and x_t and generates output in the form of 0 and 1. O means can't take and 1 means can keep. It is generated using equation 3.3

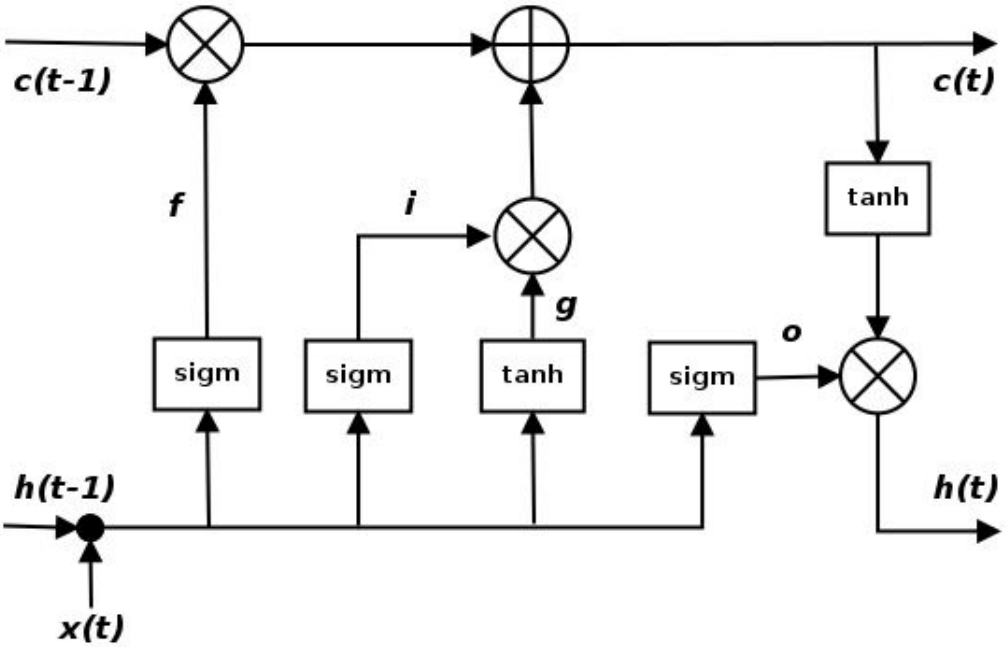


Figure 3.3: Long Short Term Memory [3]

[3].

$$f = \sigma(W_f\,[h_{t-1}, x_t + b_f] \tag{3.3}$$

The next step of LSTM layer is the calculation of which data will next store into the cell state. Input gate layer (Sigmoid Layer) decides which value should update. New candidate value is calculated by tanh layer in LSTM architecture using equation 3.4 [3]. After adding new candidate value into the cell state, it forgets old value using equation 3.5 [3].

$$C = tanh(W_c\,[h_{t-1}, x_t + b_c] \tag{3.4}$$

$$C_t = f * C_{t-1} + i * g \tag{3.5}$$

Now it is time to generate an output which is done by the sigmoid gate. Output of first iteration of LSTM is calculated by multiplying tanh layer output with output of sigmoid using equations 3.6 and 3.7 [3].

$$o = \sigma(W_o\,[h_{t-1}, x_t + b_o] \tag{3.6}$$

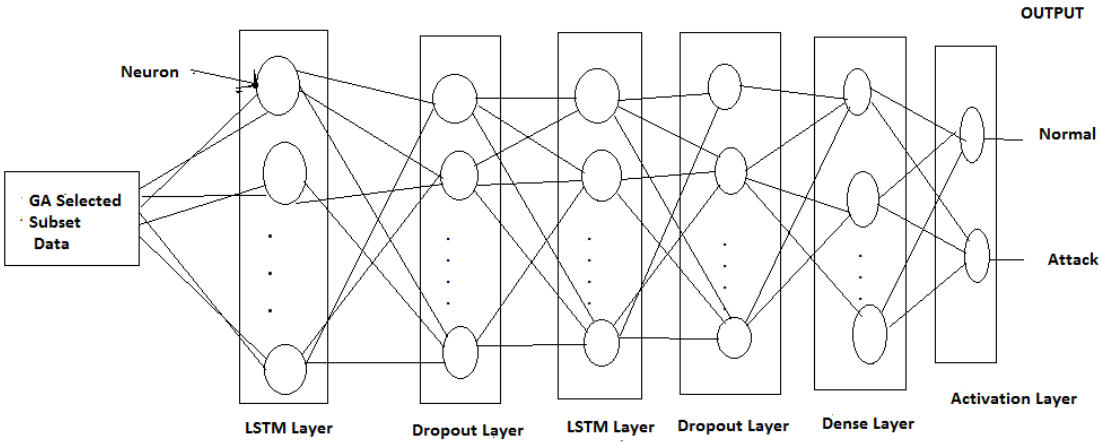


Figure 3.4: LSTM-RNN proposed model structure

$$h_t = o * tanh(C_t) \tag{3.7}$$

Here σ represents sigmoid activation function. W_o, W_C, and W_f are weights for output gate o, cell state, and forget gate f respectively.

Figure 3.4 shows our proposed LSTM-RNN model which has two LSTM-layers, two Dropout layers, one Dense layer and one activation layer. For to stop over fitting into the neural network training dropout layer is useful. So here we used it after every LSTM layer. Dense layer is nothing but fully connected neural network which is mainly for high-level reasoning in the neural network. It uses all previous layer activation function. Activation layer applying activation function for the final output layer. The activation function is almost the part of every layer. Here activation function can also be known as logistic regression layer. In our model tanh activation function which also known as the tangent logistic sigmoid function is used in every layer except final dense activation layer. For the final layer, we want to identify either it attack or normal, we just need the answer either 0 or 1.

Chapter 4

Experiment and Result

Our proposed system is trained using the LSTM-RNN model shows in figure 3.3. We compared our experiment result with the attack detection proposed system based on the multi-channel intelligent method in [39] on NSL-KDD dataset. In our experiment, genetic algorithm is used for feature selection and LSTM-RNN model for attack detection.

We performed our experiment on system with Intel Core i5-4301 CPU with 2.0 GHz. It has 8.0 GB RAM and all experiment run on tansorflow 1.2 version environment.

4.1 Dataset and Training/Testing Split

We are using KDD99CUP Dataset [50] for training our proposed model. KDD99CUP is the intrusion detection system (IDS) dataset which is used by many researchers. KDD99CUP dataset name comes from knowledge discovery and data mining. It has data of both types: attack and normal connections. Data audited in this dataset mostly simulated on military network environment. KDD99CUP is the large-scale dataset. For our experiment, we used 10% of data from the whole KDD99CUP dataset. It has mainly four types of attacks and normal traffic data with labeled target. Our goal is to identify an attack. In our experiment as we want to identify the attack, hence we take normal data and other attack data. We named attack data as abnormal data. So for us, now target becomes of two categories, normal and abnormal.

We split our dataset into training and testing. Our training/testing ratio is 80-20 %. To avoid over fitting data, we applied random split of dataset using sklearn method that

is very quick and accurate.

4.2 Initialization of parameters for the Genetic algorithm and LSTM-RNN neural network

Our proposed system is using a genetic algorithm for features selection. Before applying the Genetic algorithm, we applied random forest regressor that assigns weights to data between -1 to 1 for all individual features. We discarded features data that have weights between -1 to 0 as we want a positive maximum optimal solution. Now remaining data features are given to genetic algorithm. For the genetic algorithm, we need population and generation parameters. We initialized GA algorithm with 100 population and 10 generation. [72].

Our training model consists two LSTM-RNN layers, two dropout layers, and two dense layers. We trained LSTM-RNN model with using batch size 128, epochs 10, and learning rate 0.001. We used Adam optimizer which has the ability to update weights of training data in an iterative manner. Adam gives good and fast result for deep learning techniques. Adam has four parameters for a configuration that are, alpha, beta1, beta2, and epsilon. Alpha is known as learning rate which shows the weight update proportion. Beta1 and Beta2 are the decay rate. Epsilon saves the model from the divisible by zero error while implementation.

4.3 Experiments and Results

As discussed above, KDD99CUP dataset has total 41 features with target label. It has four attack categories (DoS, Prob, R2L, and U2R) data and normal data. Use of all features for training model is time consuming. So there is a need of reducing time complexity. For that we did some experiments using Genetic algorithm. Table 4.1 shows the experiment result with accuracy and time taken to train GA. Accuracy can be calculated using equation 2.9.

GA gives 16 features subset for the training neural network model. 4.2 shows the list of features after using correlation and GA. We used this features subset for training our

Exp. No.	Method Used	Total Number of Feature	Number of Features selected	Time taken for Feature Selection	Validation Accuracy using Logistics Regression
1	GA (100 population+10 Generation [72])	41	27	5 days	98.76 %
2	GA (100 population+10 Generation [72])	41	19	3 days	99.47 %
3	Random Forest Regressor(Correlation method)	41	24	60 min	99 %
4	Correlation+GA (100 population+10 Generation [72])	24	16	2 hours	98.79 %

Table 4.1: Features Selection Experiments results for training and testing model

Method used	Selected Features
Random Forest Regressor(RF) and GA	protocol_type, Service, flag, src_bytes, urgent, logged_in, is_guest_log, count, serror_rate, srv_rerror_rate, same_srv_rate, dst_ host_count, ds_host_diff_srv_rate, dst_host_serror_rate, dst_host_srv_serror_ rate, dst_host_srv_rerror_rate

Table 4.2: List of Selected Features

LSTM-RNN model. Table 4.3 shows the LSTM-RNN training model time and accuracy with applied feature selection methods.

Table 4.4 shows the final attack detection performance metrics value. Here in attack types, '0' means normal and '1' means attack. We compared our work with [39]. They proposed LSTM-RNN and voting algorithm for attack detection. They have used multichannel intelligent method for attack detection where as we used GA and LSTM-RNN method for attack detection. We got accuracy 99.80%. In [39] authors compared their result with Generalized Regression Neural Network (GRNN), Probablastic Neural

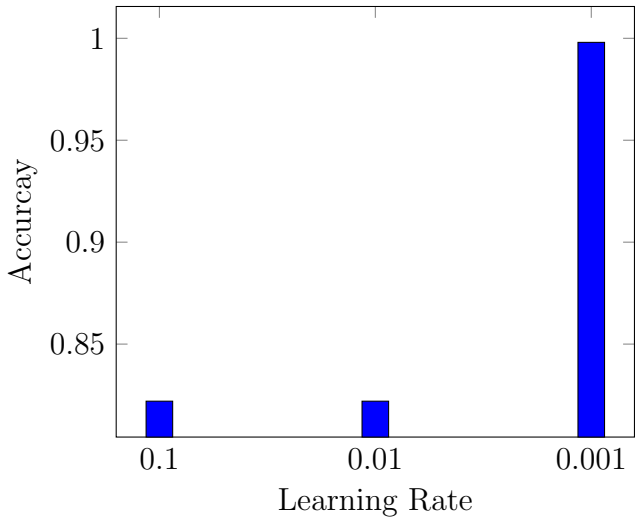
Exp. No.	Method Used	Number of Features Used	Time Taken for Training	Accuracy
1	GA+LSTM-RNN	27	1 day	86 %
2	Peorson Correlation+GA+LSTM-RNN	19	2 hours	99 %
3	Random Forest Regressor+GA+LSTM-RNN	16	10 min	99.80 %

Table 4.3: Proposed system Training experiments and results

Attack Type	**precision**	**recall**	**f1-score**	**support**
0	1.0	1.0	1.0	8815
1	0.99	0.99	0.99	2017
avg/total	1.0	1.0	1.0	10832

Table 4.4: Final Result of Trained model with performance measure metrics value

Network (PNN), Radial Basis Neural Network (RBNN), k-nearest neighbours Network (KNN), Support Vector Machine (SVM), and Bayesian methods. Compared to these metods, their proposed method has accuracy 98.94%. Compared to this result, our proposed system's performance is improved. In graph, X represents the learning rate and y represents the accuracy for comparison of function accuracy based on the learning rates.



Chapter 5

Conclusion

The threat of cybercrimes increased as Internet usage increased. Hence traffic analysis becomes necessary to handle fraud on the Internet or attacks. We discussed approaches for intrusion detection using machine learning techniques. From them, neural network based machine learning is best as per the survey. So we proposed LSTM-RNN based machine learning approach for detection of an attack. In our approach, we first applied the genetic algorithm for feature selection and after then we trained the neural network. We used test data for prediction of attack. It gives accuracy 99.80%. We compared our result with multichannel intelligent attack detection system in [39]. In [39], authors proposed LSTM-RNN model for attack detection for data security which have three channels of feature subset and voting algorithm. Training of individual three channel is done by applying different three subsets of the dataset. Voting algorithm compares all three channels output and then gives average of three channel's result as an output. Their model accuracy is 98.94%. For to improvement into the attack detection, we choose one different way using a LSTM-RNN model with GA and we got improvement into the attack detection performance.

Bibliography

[1] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abro, and M. L. Proena, "Network Anomaly Detection System using Genetic Algorithm and Fuzzy Logic," *Expert Systems with Applications*, vol. 92, pp. 390 – 402, 2018.

[2] A. Barradas-Acosta, E. Aguirre-Anaya, M. N.-M. H. Perez-Meana, and S.-E. Culhuacan, "Attacks recognition using recurrent neural network," *Recent Advances in Applied Mathematics and Computationalñ and Information Science*, vol. 2, pp. 402–409, 2009.

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

[4] R. Deebalakshmi and V. Jyothi, "A survey of classification algorithms for network traffic," in *Science Technology Engineering and Management (ICONSTEM), Second International Conference on*, pp. 151–156, IEEE, 2016.

[5] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-service Mapping for QoS: A Statistical Signature-based Approach to IP Traffic Classification," in *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, IMC '04, (New York, NY, USA), pp. 135–148, ACM, 2004.

[6] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, pp. 50–60, ACM, 2005.

[7] B. Silver, "Netman: A Learning Network Traffic Controller," in *Proceedings of the 3rd International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems - Volume 2*, IEA/AIE '90, (New York, NY, USA), pp. 923–931, ACM, 1990.

[8] J. Frank, "Artificial intelligence and intrusion detection: Current and future directions," in *Proceedings of the 17th national computer security conference*, vol. 10, pp. 1–12, Baltimore, MD, 1994.

[9] D. J. Marchette, "A Statistical Method for Profiling Network Traffic.," in *Workshop on Intrusion Detection and Network Monitoring*, pp. 119–128, 1999.

[10] K. Xu, Z.-L. Zhang, and S. Bhattacharyya, "Profiling Internet Backbone Traffic: Behavior Models and Applications," *SIGCOMM Comput. Commun. Rev.*, vol. 35, pp. 169–180, Aug. 2005.

[11] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," in *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)l*, pp. 250–257, Nov 2005.

[12] Y. Liu, W. Li, and Y. C. Li, "Network Traffic Classification Using K-means Clustering," in *Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007)*, pp. 360–365, Aug 2007.

[13] R. Alshammari and A. N. Zincir-Heywood, "Machine learning based encrypted traffic classification: Identifying SSH and Skype," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–8, July 2009.

[14] F. Dehghani, N. Movahhedinia, M. R. Khayyambashi, and S. Kianian, "Real-Time Traffic Classification Based on Statistical and Payload Content Features," in *2010 2nd International Workshop on Intelligent Systems and Applications*, pp. 1–4, May 2010.

[15] T. R. Patil and S. Sherekar, "Performance analysis of naive bayes and j48 classification algorithm for data classification," *International Journal of Computer Science and Applications*, vol. 6, no. 2, pp. 256–261, 2013.

[16] A. W. Moore and K. Papagiannaki, "Toward the Accurate Identification of Network Applications.," in *PAM*, vol. 5, pp. 41–54, Springer, 2005.

[17] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys Tutorials*, vol. 10, pp. 56–76, Fourth 2008.

[18] R. Archanaa, V. Athulya, T. Rajasundari, and M. V. K. Kiran, "A comparative performance analysis on network traffic classification using supervised learning algorithms," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 1–5, Jan 2017.

[19] A. Fadlil, I. Riadi, and S. Aji, "DDoS Attacks Classification using Numeric Attribute-based Gaussian Naive Bayes," *International journal of advanced computer science and applications*, vol. 8, no. 8, pp. 42–50, 2017.

[20] M. Stevanovic and J. M. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," in *2014 International Conference on Computing, Networking and Communications (ICNC)*, pp. 797–801, Feb 2014.

[21] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection," *Knowledge-Based Systems*, vol. 64, pp. 22 – 31, 2014.

[22] L. Lu, Y. Feng, and K. Sakurai, "#38;C Session Detection Using Random Forest," in *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication*, IMCOM '17, (New York, NY, USA), pp. 34:1–34:6, ACM, 2017.

[23] A. Munther, A. Alalousi, S. Nizam, R. R. Othman, and M. Anbar, "Network traffic classification x2014; A comparative study of two common decision tree methods: C4.5 and Random forest," in *2014 2nd International Conference on Electronic Design (ICED)*, pp. 210–214, Aug 2014.

[24] W. Li, "Using genetic algorithm for network intrusion detection," *Proceedings of the United States Department of Energy Cyber Security Group*, vol. 1, pp. 1–8, 2004.

[25] A. Este, F. Gringoli, and L. Salgarelli, "Support vector machines for TCP traffic classification," *Computer Networks*, vol. 53, no. 14, pp. 2476–2490, 2009.

[26] V. D'Alessandro, B. Park, L. Romano, C. Fetzer, *et al.*, "Scalable network traffic classification using distributed support vector machines," in *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*, pp. 1008–1012, IEEE, 2015.

[27] Y. Liu, W. Li, and Y.-C. Li, "Network traffic classification using k-means clustering," in *Computer and Computational Sciences, 2007. IMSCCS 2007. Second International Multi-Symposiums on*, pp. 360–365, IEEE, 2007.

[28] J. Erman, M. Arlitt, and A. Mahanti, "Traffic Classification Using Clustering Algorithms," in *Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data*, MineNet '06, (New York, NY, USA), pp. 281–286, ACM, 2006.

[29] H. Singh, "Performance Analysis of Unsupervised Machine Learning Techniques for Network Traffic Classification," in *2015 Fifth International Conference on Advanced Computing Communication Technologies*, pp. 401–404, Feb 2015.

[30] M. Bailey, C. Collins, M. Sinda, and G. Hu, "Intrusion detection using clustering of network traffic flows," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2017 18th IEEE/ACIS International Conference on*, pp. 615–620, IEEE, 2017.

[31] J. Zhang, Y. Tong, and T. Qin, "Traffic Features Extraction and Clustering Analysis for Abnormal Behavior Detection," in *Proceedings of the 2016 International Conference on Intelligent Information Processing*, ICIIP '16, (New York, NY, USA), pp. 25:1–25:6, ACM, 2016.

[32] T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection," *Information Sciences*, vol. 177, no. 18, pp. 3799 – 3821, 2007.

[33] A. Puri and N. Sharma, "A novel technique for intrusion detection system for network security using hybrid SVM-CART," 2017.

[34] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-based systems*, vol. 78, pp. 13–21, 2015.

[35] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4, Part 2, pp. 1690 – 1700, 2014.

[36] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *Intelligence*

and Security Informatics (ISI), 2017 IEEE International Conference on, pp. 43–48, IEEE, 2017.

[37] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *2017 International Conference on Information Networking (ICOIN)*, pp. 712–717, Jan 2017.

[38] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *Platform Technology and Service (PlatCon), 2016 International Conference on*, pp. 1–5, IEEE, 2016.

[39] F. Jiang, Y. Fu, B. B. Gupta, F. Lou, S. Rho, F. Meng, and Z. Tian, "Deep Learning based Multi-channel intelligent attack detection for Data Security," *IEEE Transactions on Sustainable Computing*, vol. PP, no. 99, pp. 1–1, 2018.

[40] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly detection in cyber physical systems using recurrent neural networks," in *High Assurance Systems Engineering (HASE), 2017 IEEE 18th International Symposium on*, pp. 140–145, IEEE, 2017.

[41] J. Kim, H. Kim, *et al.*, "An Effective Intrusion Detection Classifier Using Long Short-Term Memory with Gradient Descent Optimization," in *Platform Technology and Service (PlatCon), 2017 International Conference on*, pp. 1–6, IEEE, 2017.

[42] 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.

[43] T. Ishitaki, R. Obukata, T. Oda, and L. Barolli, "Application of Deep Recurrent Neural Networks for Prediction of User Behavior in Tor Networks," in *Advanced Information Networking and Applications Workshops (WAINA), 2017 31st International Conference on*, pp. 238–243, IEEE, 2017.

[44] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *Wireless Networks and Mobile Communications (WINCOM), 2016 International Conference on*, pp. 258–263, IEEE, 2016.

[45] N. T. Van, T. N. Thinh, and L. T. Sach, "An anomaly-based network intrusion detection system using Deep learning," in *System Science and Engineering (ICSSE), 2017 International Conference on*, pp. 210–214, IEEE, 2017.

[46] N. T. Van, T. N. Thinh, and L. T. Sach, "An anomaly-based network intrusion detection system using Deep learning," in *2017 International Conference on System Science and Engineering (ICSSE)*, pp. 210–214, July 2017.

[47] R. CAIDA and Collaborators, "CAIDA DATASET," 1997. Available: https://www.caida.org/data/, Last accessed: 2018-05-07.

[48] I. LBNL and L. B. N. Laboratory, "LBNL/ICSI Enterprise Tracing Project," 2013. Available: http://www.icir.org/enterprise-tracing/Overview.html, Last accessed: 2018-05-07.

[49] L. L. M. I. of Technology, "DARPA Intrusion Detection Data Sets," 1998. Available: https://www.ll.mit.edu/ideval/data/1998data.html, Last accessed: 2018-05-07.

[50] I. The UCI KDD Archive and C. S. U. of California, "KDD cup 1999 Data," 1999. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Last accessed: 2018-05-07.

[51] L. Privacy and S. Notice, "Internet traffic archive," 2000. Available: http://ita.ee.lbl.gov/, Last accessed: 2018-05-07.

[52] U. o. B. UNB, "ISCX," 2012. Available: http://www.unb.ca/cic/datasets/index.html, Last accessed: 2018-05-07.

[53] K. U. H. Kyoto University, "Kyoto Dataset," 2016. Available: http://www.takakura.com/Kyoto$_data$/, $Lastaccessed : 2018 - 05 - 07$.

[54] W. P. Kenjiro Cho, "MAWI Working Group Traffic Archive," 1999. Available: http://mawi.wide.ad.jp/mawi/, Last accessed: 2018-05-07.

[55] L. Queen Mary University, "Queen Mary Research Online," 2000. Available: http://library.qmul.ac.uk/research/research-data-management/preserving-your-data/, Last accessed: 2018-05-07.

[56] U. Associated ground truth, "UNIBS:Data sharing," 2009. Available: http://netweb.ing.unibs.it/ ntw/tools/traces/, Last accessed: 2018-05-07.

[57] I. a. l. University of Auckland, "Aucland IV," 2001. Available: https://wand.net.nz/wits/auck/4/, Last accessed: 2018-05-07.

[58] A. Moore, J. Hall, C. Kreibich, E. Harris, and I. Pratt, "Architecture of a network monitor," in *Passive & Active Measurement Workshop*, vol. 2003, 2003.

[59] Y. Wang, Y. Xiang, J. Zhang, and S. Yu, "A novel semi-supervised approach for network traffic clustering," in *Network and System Security (NSS), 2011 5th International Conference on*, pp. 169–175, IEEE, 2011.

[60] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, W. Lu, J. Felix, and P. Hakimian, "Detecting P2P botnets through network behavior analysis and machine learning," in *Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on*, pp. 174–180, IEEE, 2011.

[61] G. F. J. S. Mark Hopkins, Erik Reeber, "UCI Machine Learning Dataset," 1999. Available: http://archive.ics.uci.edu/ml/datasets/Spambase, Last accessed: 2018/05/07.

[62] A. m. r. c. MWS, "CCC and Practice Dataset," 2014. Available: http://www.iwsec.org/mws/2014/about.html, Last accessed: 2018/05/07.

[63] S. Sahu and B. M. Mehtre, "Network intrusion detection system using J48 Decision Tree," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2023–2026, Aug 2015.

[64] D. I. Lincon Laboratory, "DARPA2009," 2009. Available: https://www.researchgate.net/publication/279850205$_{C}reating_{N}ovel_{F}eatures_{t}o_{A}nomaly_{N}etwork_{D}etect$ $2009_{D}ata_{s}et, Lastaccessed : 2018 - 05 - 07$.

[65] V. Carela-Español, T. Bujlow, and P. Barlet-Ros, "Is our ground-truth for traffic classification reliable?," in *International Conference on Passive and Active Network Measurement*, pp. 98–108, Springer, 2014.

[66] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection." http://snap.stanford.edu/data, June 2014.

[67] J. Brown, M. Anwar, and G. Dozier, "An evolutionary general regression neural network classifier for intrusion detection," in *Computer Communication and Networks (ICCCN), 2016 25th International Conference on*, pp. 1–5, IEEE, 2016.

[68] A. Saabas, "Random Forest Regression," 2016. Available: http://blog.datadive.net/selecting-good-features-part-iii-random-forests/, Last accessed: 2018/05/11.

[69] J. L. Ribeiro Filho, P. C. Treleaven, and C. Alippi, "Genetic-algorithm programming environments," *Computer*, vol. 27, no. 6, pp. 28–43, 1994.

[70] I. Wikimedia Foundation, "RNN hidden and output vector calculation," 2018. Available: https://en.wikipedia.org/wiki/Recurrent$_n eural_n etwork, Last Accessed = 2018/05/11.$

[71] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[72] G. Stein, B. Chen, A. S. Wu, and K. A. Hua, "Decision Tree Classifier for Network Intrusion Detection with GA-based Feature Selection," in *Proceedings of the 43rd Annual Southeast Regional Conference - Volume 2*, ACM-SE 43, (New York, NY, USA), pp. 136–141, ACM, 2005.