Russian Federation Ministry of Education
Radio technique and Cybernetics Department
Moscow Institute of Physics And Technology(State University)
Intelligent Information System and Technology Chair

Classification Encrypted Network Traffic Using Convolution Neural Networks combine with Long Short Term Memory

A thesis presented for the degree of Master
Direction of training: 03.04.01 Applied Mathematics and Physics

Performer:
Duong Viet Tung

Supervisor:
Prof A.N.Nazaroz

Consultant:
Ilia Voronkov

Abstract

With the current rapid development of the internet, it is undoubtedly an indispensable part of our lives. The internet can connect everything to each other, devices, objects or people with people. And the most attention recent example is The RIPE NCC (Réseaux IP Européens Network Coordination Centre) has run out of IPv4 Addresses, it is means that network in Europe, Middle Earth and part of Central Asia now are not able to receive new Ipv4 address. Along with that is the exchange of extremely large information flows between points over a given period of time. It also raises concerns about information security and confidentiality.

And that why, nowadays, internet traffic classification has become more important . It have important role in network management and cybersecurity. By keeping information about traffic flow and classifying them, we can identify destabilizing factors that can occur, thereby preventing and minimizing impacts, bad behavior may harmful the system. With the expansion of encrypted methods are becoming popular.

Virtual Private Networks (VPNs) or Tor are an example, extends a private network across a public network, and enables users to send and receive data across shared or public networks.

From there they can get information as they want, something like a movie or a copyright game from unofficial source. But that also causes certain bad factors to penetrate their computers and terminals, uncontrollably. And for those reason, we need an tool or method to detect and classify network traffic has been encrypted from other source.

In this paper, I proposed this method using Convolution Neural Network combining with Long Short Term Memory to examines and classification encrypted traffic through VPN. The method is validated with the public ISCX VPN-nonVPN traffic dataset. It is very famous dataset and has been used widely for research in the world.

Declaration

I declare that this thesis is an original report of my research, has been written by me and has not been submitted for any previous degree. The experimental work is almost entirely my own work, the collaborative contributions have been indicated clearly and acknowledged. Due references have been provided on all supporting literature and resources.

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification.

Signature:
Name: Duong Viet Tung
Group: M01-818a
Date: 15/06/2020

Acknowledgement

At first, I would like to thank my supervisor, Professor Alexey Nazarov and my moderator, Mr.Ilya Voronkov, for guiding me throughout the completion of this project.

The second, I would like to give my deep thanks to Professor Avedian and Mr.Pantiukhin for teaching me all duration of the course, and give me knowledge to complete this project.

I also give thank to my classmates and my friends Truong Dang Khoa, Tran Anh Duc for helping me when I need.

And the last but not least is my family, whose always stand besides me and a solid fulcrum for me to keep going and finish my work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Preamble

The traffic encryption has been widely used in the Internet due to the advanced encryption techniques. The encryption techniques not only protect Internet user's freedom, privacy and anonymous,but also make them evade the detection of firewall. Moreover, encryption is becoming widespread in today's Internet, serving as a base for secure communications. Nowadays, traffic classification has attracted a lot of interests in both academia and industrial activities related to network management.

According to the difference of ISO/OSI layer (will be introduced in more detail in the following chapters), traffic encryption techniques can be divided into application layer encryption, presentation layer encryption and network layer encryption. Some tunnel technology such as VPN is based on these techniques. This encryption type is also called protocol encapsulation. In some cases, encrypted traffic through regular encryption can be further encrypted through protocol encapsulation (e.g. Skype traffic through VPN).

1.2 Motivation

1.2.1 Business Context

The network traffic classification problem can be seen as a never ending race between application developers, on the one side, and the network operators and the research community, on the other. Today, the amount of web traffic is growing rapidly and Internet access prices are decreasing constantly. Therefore Internet service providers must find ways to cut expenses while not affecting their QoS (Quality of Service) in order to keep their customer base. In simple words, QoS is the ability of the network to provide better or special service to a set of users and application to the detriment of others.

- "Ability of the network" - it means that this will be done by network infrastructure i.e., switches and routers. There will be no traffic shaping server, the network infrastructure have to do this job.
- "To provide better or 'special' service" - it means some traffic flow will be given preferential treatment and others will be given what is left over.
- "To a set of users and applications" - it means which flow should get preference and which will get detriment treatment .
- "To the detriment of others" - it means that since there will be a finite amount of bandwidth, if some flow is given more bandwidth there will be less available for the remaining traffic.

The problem ISPs (Internet service provider) are facing is that the traffic is encrypted and therefore cannot be easily focus on. Another problem faced by companies is network enforcement. A sophisticated user can exploit existing web traffic as well as add encryption layers to your own traffic to circumvent company policies. Companies can address these issues by using pattern recognition to gain more knowledge about encrypted traffic and the ability to classify it. Zander et al [46] propose unsupervised machine learning methods for streaming traffic and enabling QoS. This method can also be used to enforce corporate network policies.

1.2.2 Security context

There are various types of threats, attacks and vulnerabilities present to corrupt and breach the system security. Security attacks are the computer attacks that compromise the security of the system. Conceptually, the security attacks can be classified into two types that are active and passive attacks where the attacker gains illegal access to the system resources. During a passive attack the attacker intercepts the transit information with the intention of reading and analyzing the information not for altering it. Specifically, in our context, a passive case,enemy could violate a user's privacy by analyze the network traffic generated by people devices on the network side. The passive attack is hard to detect because it does not involve any alteration in the data or system resources. Thus, the attacked entity does not get any clue about the attack. Although, it can be prevented using encryption methods in which the data is firstly encoded in the illegible language at the sender end and then at the receivers end it is again converted into human understandable language. The passive attacks are further classified into two types, first is the release of message content and second is traffic analysis.

- The release of message content can be expressed with an example, in which the sender wants to send a confidential message or email to the receiver. The sender does not want the contents of that message to be read by some interceptor.

- By using encryption a message could be masked in order to prevent the extraction of the information from the message, even if the message is captured. Though still attacker can analyse the traffic and observe the pattern to retrieve the information. This type of passive attack refers to as traffic analysis.

Active attacks are the attacks in which the attacker tries to modify the information or creates a false message. The attacker could use the email address and ICC-ID of the Mayor of New York to obtain his password (by further passive attacks) and login to his iPad or email account. Thereafter the attacker could actively perform malicious actions attributed to his target. The prevention of these attacks is quite difficult because of a broad range of potential physical, network and software vulnerabilities. Instead of prevention, it emphasizes on the detection of the attack and recovery from any disruption or delay caused by it. The active attacks are in the form of interruption, modification and fabrication.

- Interruption is known as masquerade attack in which unauthorized attacker tries to pose as another entity.

- Modification can be done using two ways replay attack and alteration. In the replay attack, a sequence of events or some data units is captured and resent by them. While alteration of the message involves some change to the original message, either one of them can cause alteration.

- Fabrication causes Denial Of Service (DOS) attacks in which attacker strive to prevent licit users from accessing some services, which they are permitted to or in simple words the attacker gain access to the network and then lock the authorized user out.

In this work, I consider passive attacks. If network traffic is not encrypted, the passive attacker's task is relatively simple as he can analyze the payload and examine the information of each packet. Tracking user activity on the web has been proposed as a preventive measure. This is done by analyzing un-encrypted HTTP requests and responses. Such a passive attacker could use the information he gathered to analyze the user's activity as well as reveal information around the user's interests and habits. However, as traffic encryption becomes more and more popular, traditional methods for classify network traffic are no longer useful. Many works have shown that encryption is not sufficient to protect confidentiality. So my motivation is apply new technology (Deep Learning) to make it possible and gain good results when apply it into this case.

1.3 Classical Approaches

This section describes advanced methods in the field of network classification. I call this classification identification the application layer classification (eg skype, facebook). And it belongs to the 7th layer of the OSI network model. The application identification problem has been constantly changing, depending on the needs of a given task. On the one hand, applications and especially those that do not want to be detected (for example, P2P applications) to use network resources without control. On the other hand, network operators, researchers and even ISPs need to know network traffic characteristics to manage resources. This optimizes the use of network traffic And so, different ways have been applied to achieve best results for classify traffic.

1.3.1 Port-Numbers based Approach

Many different approaches has been used for that task classification network traffic [27],[16]. Ports-based method is the most simple method. That is very useful only for the application and services, which used fixed port numbers, but we can easy cheating by changing the port numbers in the system. For example:now Peer-to-peer are rising and which can use port 80 to associated

Port Number	Assignment
20	File Transfer Protocol (FTP) Data Transfer
21	File Transfer Protocol (FTP) Command Control
22	Secure Shell (SSH) Secure Login
23	Telnet remote login service, unencrypted text messages
25	Simple Mail Transfer Protocol (SMTP) E-mail routing
53	Domain Name System (DNS) service
67,68	Dynamic Host Configuration Protocol (DHCP)
80	Hypertext Transfer Protocol (HTTP)
110	Post Office Protocol (POP3)
119	Network News Transfer Protocol (NNTP)
123	Network Time Protocol (NTP)
143	Internet Message Access Protocol (IMAP)
161	Simple Network Management Protocol (SNMP)
194	Internet Relay Chat (IRC)

Table 1.1: Common Port number
[44]

with HTTP traffic.So the Port-based method has no sense for this case. A limitation here is only looking for port number. This table 1.1 above shows an example of several common ports number and application assigned with it. For more example, DNS service use port 53 and Dynamic Host configuration protocol use port 67 and 68 and port 110 (POP3) to receive emails. Invalid because of the inaccuracy and incompleteness of its classification results. We may find it very difficult to set limits on the current accuracy of these methods because it largely depends on the characteristics of the monitored network and the system to establish the truth that the platform is used for. to authenticate them.

1.3.2 Deep Packet Inspection Approach

In other way, DPI(Deep Packet Inspection) has shown it is more effective than Port based when trying classify P2P network. Because DPI examine the content of the packets looking for characteristic signatures of the applications in the traffic [5] and [17]. For more deeper, Deep packet inspection is a form of packet filtering usually carried out as a function of your firewall. It

Website	**Domain**	**Category**
Wikipedia	wikipedia.org	Censorship-free encyclopedia
Google	google.com	Worldwide Internet search engine
Google Encrypted	google.com	Search
Facebook	facebook.com	Social network
Youtube	youtube.com	Video
OpenVPN	openvpn.net	Avoidance of political internet censorship
Strong VPN	strongvpn.com	Avoidance of political internet censorship
Falun Dafa	falundafa.org	Spiritual
VPN Counpons	vpncoupons.com	Avoidance of political internet censorship

Table 1.2: Website blocked in China using DPI method
[43]

is applied at the Open Systems Interconnection's application layer. Deep packet inspection evaluates the contents of a packet that is going through a checkpoint. Using rules that are assigned by anyone, your ISPs, or the network or systems administrator, Deep packet inspection determines what to do with these packets in real time. In addition, it can work with filters in order to find and redirect network traffic from an online service, such as Twitter or Facebook, or from a particular IP address. But it also has certain drawbacks like requires a lot of processing power and slow and in some case this method cannot de-crypt the encrypted traffic. So with the increasing amount of encrypted traffic (using VPN,TOR...), DPI show disadvantages and can not be use for encrypted traffic classification task. We can see this example in table 1.2 above about China using this method to monitor and censor network traffic and content that it claims is harmful to citizens or state interests.

1.3.3 Statistical based Approach

The next, statistical classification is machine learning approaches. Example [18],[35] is Support Vector Machine . Decision Tree is [14] and [32]. Other method, K-Nearest Neighbour (KNN) is[25].There are some popular Machine Learning method were proposed for traffic classification. The workflow for this method is explained as follows: Firstly, we process raw data and extract features which is useful. After that, we classify traffic with those features by some ML classifier. For more details, Machine Learning method uses data sets (usually labeled) from which the feature set is extracted. This information is an input to the ML technique to extract and export knowledge in different structures (e.g. decision trees, clusters) depending on the ML technique used. The resulting structure is then used to classify cases that are not labeled with the assumption that the features of the undefined cases will behave in the same manner as the identified cases. Generally, there are two kind of ML classifier are used: one is the supervised methods like decision tree and Naive Bayes, the other is unsupervised approaches like k-means .

- Supervised Learning: Known as classifications method, require a pre-classified data set called a training data set, the model will use the training data to learn a link between the input and the outputs. The classification accuracy of supervised ML algorithms are evaluated by applying them to test set.

- Unsupervised Learning Approach: Unsupervised ML algorithms cluster the data to be classified and associate these clusters to traffic classes. So it do not need a complete labeled training set. Basically, three main clustering methods have been used in the network traffic classification literature: classical cluster algorithms forming clusters in numerical domains, cases of partitioning into separate clusters; Incremental clustering creates a group of hierarchical cases and; Probability based method assigns instances to probability classes, not defined. They generally achieve slightly lower accuracy than supervised techniques, but with much less training time.

But for that traditional ML classifier, it contains some factors cause instability. As we shown before the progress of that method. The accuracy is depends on the experience of users whose choose features. Some features might not be analyzed or used, but it can represent traffic flow quite well.

1.3.4 Host-behavior-based Approach

Another method introduced for overcoming disadvantages of the payload-based methods is the host-behavior-based techniques. It utilize host behavior to solve classification problems. These algorithms apply heuristic theory to perform classification effectively, especially for covered traffic A number of researchers have studied host behavior in the field of traffic measurement [45]. The application runs at the host and creates Traffic is part of the server traffic profile. Traffic Profiles show which favorite applications are used on the host. Once the host's favorite information is relevant, it applied, it may help to categorize traffic. For example, the web browser server is more vulnerable to open HTTP connections consecutively. A very capable host to receive POP3 flow when it is running the POP3 mail server. For example, Karagiannis, et al [15] developed behavior of the host at three levels: Social level, Functional level and Application level. And it achieved results approximately 80-90% of total flows with 95% accuracy.

Conclusion With the Approach mentioned above, there are certain good points and limitations. However, with the rapid and strong development of Deep Learning in recent times, we can completely use them as a new approach to solve the problem of classification of encrypted network traffic. Deep Learning can alleviate the heavy work of selecting features and collecting information about private features as it automatically extracts and selects features through training. And in the next chapter we will see articles and achievements related to using Deep Learning to classify encrypted network traffic.

1.4 Thesis Organization

In this paper, I proposed an method classification encrypted traffic using Convolution neural network and Long Short Term Memory. This model based on Deep Learning which can minimize data pre-processing requirements. It means, characteristics of CNN can affect and reduce parameters with still keep accuracy. There are three main things: sparse connectivity, pooling and weight sharing. That why CNN can be deeper than other traditional neural networks without computing complex. And it is also promising in application for classification encrypted traffic. For more details, we will go deeper in the next few chapters

The main contributions of the paper are as follows:

- Proposed an Encrypted traffic classification method with CNN+LSTM.
- We perform experiments to demonstrate the impact of this method using CNN+LSTM for encrypted traffic classification.

- Increase accuracy of this method and compare with another method.

The rest of paper is organized as follows. Related works in the network traffic classification for Deep Learning approaches are discussed in **Chapter 2**. In **Chapter 3** is Networking Background. In **Chapter 4** is Neural Network Background (Includes Convolution Neural Network and Long Short Term Memory knowledge). In the next, **Chapter 5** describes the dataset and how can I executed on the dataset (methodology), while **Chapter 6** presents and discusses the experiments results. Finally, **Chapter 7** presents the conclusions and future work.

Chapter 2

Related Work

There are many researches on regular encrypted traffic classification based on Neural networks or Deep learning approach now.For example:

In 2010, Sun, Runyuan, et al [36] proposed Probabilistic Neural Network (PNN). For identify Web and P2P, the results they receive is better when compare with other ML method. Like Support Vector Machine (SVM) or Radial Basis Function Neural Network (RBFNN). Their Model based on Bayes decision rule, model is simple with three layers: Input layer, Pattern layer (Hidden layer), output layer. The figure 2.1 below show the pattern neurons by dot product between input vector X and weight vector W, and pattern neurons perform non linear operator. The nonlinear operator applied in the pattern neuron is usually a radial basis function, so the input pattern layer is usually called radial basis layer. IF X and W_i are both normalized to unit length, then the output of ith pattern unit is:

$$net(H_i) = e^{-\frac{(W_i^{xh}-X)^T(W_i^{xh}-X)}{2\sigma^2}} \tag{2.1}$$

and they have the output as follows:

$$y_i = \sum_{i=1}^{h} W_{ij}^{hy} net(H_i), j = 1, 2, ..., m \tag{2.2}$$

Where h is the number of pattern unit. W_{ij}^{hy} is the weight between ith pattern unit and jth output unit,

For Dataset, they designed a Distributed host based traffic collecting platform (DHTCP). They applied DHTCP in two laboratory environments of Harbin Institute of Technology and the University of Jinan consequently. Data are collected from 21 to 29 November 2009 And the raw data of DHTCP gathered only contains basic process, traffic and packet information. Total size of selected data is about 68 megabytes, not so much. It also contains 49 types of application in the dataset. And about 22 statistical features were used in this data. The accuracy received usually around 88%, is an acceptable result for a simple PNN network.

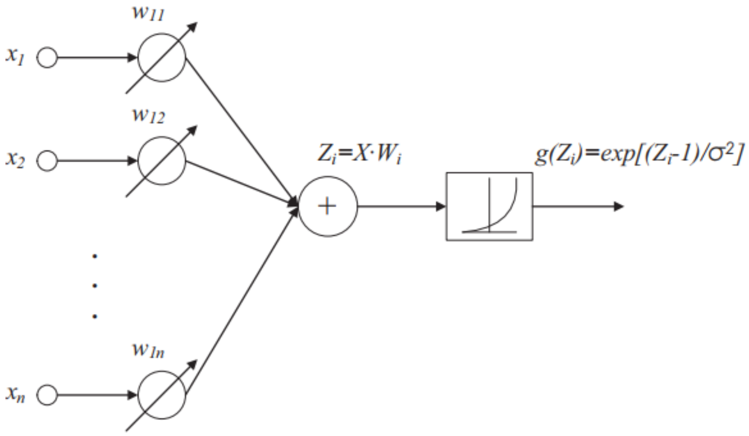


Figure 2.1: Pattern neuron
[36]

In the same year, Ting, Hu, Wang Yong and Tao Xiaoling [38], they used Kernel Self-Organizing-Map (KSOM) for traffic classification. We know SOM is able to automatically cluster according to internal relation between the inputs samples. They applied kernel method into SOM to make it can classification network flow. The kernel method can replace inner product of the maps values with kernel function. That why KSOM can change non-linearity of input space to linear problem more easily in high dimension feature space. We have distance measurements function here:

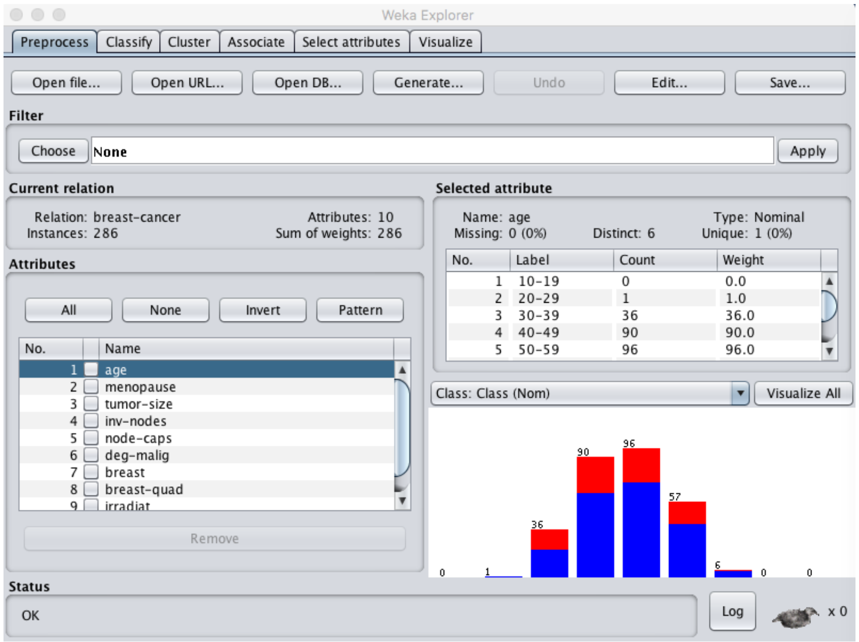
$$J(w_j) = K(X, X) + K(w_j, w_j) - 2K(X, w_j) \tag{2.3}$$

When K is kernel function, w is weight vector and X is the sample set, by substitute this equation for weight equation of traditional SOM, we have:

$$
\begin{aligned}
w_j(t+1) &= w_j(t) + \eta(t)J(w_j) \\
&= w_j(t) + \eta(t)(\frac{\partial K(w_j, w_j)}{\partial w_j} - 2\frac{\partial K(X, w_j)}{\partial w_j})
\end{aligned} \tag{2.4}
$$

We can apply different kernel function for new weight formulas. They used dataset named Moore Set, were created by Moore and Zuev [28] for purpose flow classification. In that dataset, flows are defined by tuple consisting of source and destination IP, source and destination port and protocol. Each flow has 248 features. The results compared with Naive-Bayes method is better with accuracy approximately 94%. Furthermore, this method also good for visualization of classification results.

Miller, Shane, Kevin Curran and Tom Lunney [26], proposed Multilayer Perceptron (MLP) Neural Network, a very familiar and basic neural network for us. In their paper, Netmate was mentioned as an analyser tool for convert capture files into flow record, seems useful for pre-processing phase. The overall size of the dataset captured and processed through NetMate was 9829 flows with 3569 flows representing VPN traffic and 6260 flows representing Non-VPN traffic. The training dataset contained 7863 instances. The final testing dataset contained 1257 instances and the validation dataset contained 253 instances. They also used another tool called Weka (see picture 2.2 below) for feature extraction and also for constructed neural network structure (choose parameters). Weka have a graphical user interface for easy access the functions and contains a set of intuitive tools and algorithms for data analysis and predictive models. And MLP model complete with six hidden layers. This model look quite simple but see the results is acceptable. The accuracy about 92% and 93% for VPN-nonVPN dataset.



Filter

None

Current relation

Relation: breast-cancer
Instances: 286
Attributes: 10
Sum of weights: 286

Selected attribute

Name: age
Missing: 0 (0%)
Distinct: 6
Type: Nominal
Unique: 1 (0%)

No.	Label	Count	Weight
1	10-19	0	0.0
2	20-29	1	1.0
3	30-39	36	36.0
4	40-49	90	90.0
5	50-59	96	96.0

Attributes

No.	Name
1	☐ age
2	☐ menopause
3	☐ tumor-size
4	☐ inv-nodes
5	☐ node-caps
6	☐ deg-malig
7	☐ breast
8	☐ breast-quad
9	☐ irradiat

Class: Class (Nom)

Status

OK

Figure 2.2: Weka tool Interface
[3]

Zhou, Huiyi, et al [48], proposed model complementary CNN, it is called Min-Max Normalization Convolutional Neural Networks(MMN-CNN). So they constructed the matrix dimension, which is suitable for the study of convolution neural network according to the number of features. Then, each element in the matrix is treated as a pixel, the value in the matrix is the gray scale of the pixel. Dataset they used is Moore Dataset proposed above. They designed six different types of MNN-CNN based on structure of LeNet-5 to choose which one is better. In basically only the size of the layers is changed, so the results we have will not change so much for each network. Experimental results show that with the increase in the number of feature maps, the accuracy of traffic classification has changed a little, but the time for testing traffic classification has increased clearly. The overall results are very impressed 99.3% with max-pooling. And compared with Principal Component Analysis(PCA), Gauss Random Projection(GRP) and Sparse Random Projection (SRP), MNN-CNN get a best results. For more details we can see the table 2.1.

Cui, Susu, et al [7], proposed CapsNet which is encrypted traffic classifi-

Method	PCA	SRP	MMN-CNN
Overall Accuracy(%)	96.09	97.23	99.30
Testing Time(s)	6.03	5.92	6.03

Table 2.1: Compare Different Algorithms
[48]

cation based on session packet. So in the pre-processing phase, they do some following step: Pcap-session segmentation,deleting MAC-IP, session-packets segmentation, unifying input size and converted to IDX format to fed into CapsNet model. CapsNet with the size of 28*28 and consists of convolution operation and dynamic routing. CapsNet use vector instead of scalar as input and output, and don not use pooling layers. The first and second layers of CapsNet are normal Convolution layers, third layer is Digit-Caps disseminate and update input capsule. The capsule processing is divided into two steps: linear combination and routing. And final are two Fully connected layer with Softmax classifier. The results of this model are really impressive when Precision, Recall and F1 score also get 99.3%.

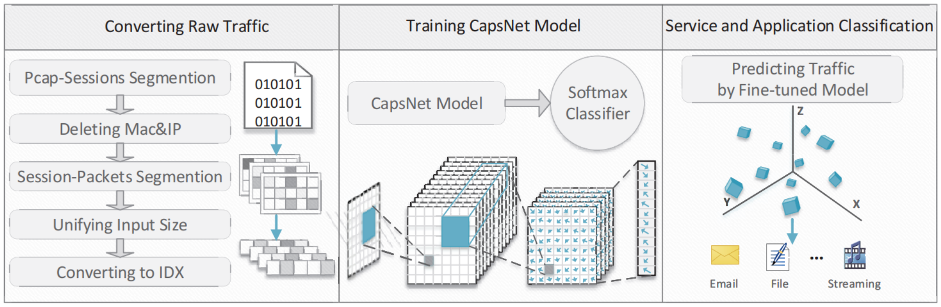


Figure 2.3: SPCaps Framework
[7]

Another researcher is about Convolution Long Short-Term Memory Neural Network, Zou, Zhuang, et al [49], this is complicated model. They combined both of the Convolution neural network and the recurrent network. And this model extracts features in both the packet and flow level. Dataset they used is ISCX VPN-not VPN, which is very famous dataset for traffic classification, it is contains 25GB raw traffic in Pcap format, and they relabeled into 12 classes. The Convolution neural network is used to examine the packet features hidden inner a single packet. It is contains two Convolution layers (Conv1 & Conv2) and one Fully connected Layer(FC). For more detais, we can look down the figure 2.4. ReLu is used as activation function. In the end of FC layer, they added dropout layer which is has 0.25 probability. The recurrent neural network is trained to extracting the flow features based on the inputs of the packet features of any three sequence packets in a flow. Recurrent layer is used LSTM cells, number of hidden units in LSTM is 256. Furthermore, to avoid overfitting, the output probability is set at 0.8. About the results, their model achieved average precision and recall 91%. For additional, Lopez-Martin, Manuel et al [23], also proposed model with CNN and RNN for classification. But in their paper, more models are used to show the obvious effect. They introduced seven combined model CNN+RNN for experiments in other case. And most used for comparing is model CNN+RNN 2a, which is received good results with accuracy always higher than 98%, and in many case higher than 99% for 15 frequent labels.

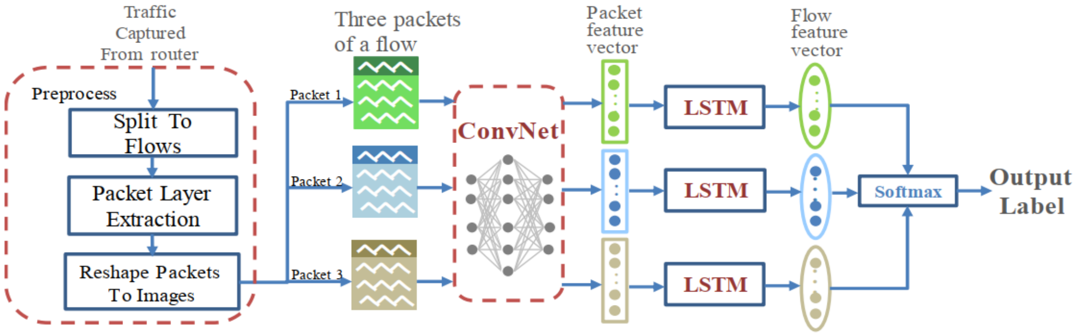


Figure 2.4: Architectures of model CONV+LSTM
[49]

And so interesting, Vu, Ly et al [40] used Time Series analysis for encrypted traffic. They combined the payload-based and flow-based methods, figure 2.5 show how they do feature extraction. In this figure, features one through three describe flow definition including source transport layer port, destination transport layer port and protocol. Package samples are arranged in a time series. In this work, they use this information to recognize the flow of continuous packets. The fourth feature showing the size of the application data is strongly represented Differences between network traffic applications. Moreover, in an encrypted Internet application, the IP header and transmission header of the IP packets are not encrypted. Therefore, they can represent these headings in byte values. Features from 5 to 44, totally 40 features, accurately representing the values of the 20-byte IP header and the 20-byte transmission header. For transport layer headers, there are two common protocols, i.e., User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). The size of the UDP header is 8 bytes, while the size of the TCP header is usually 20 bytes. Therefore, they padded 0 bytes at the end of the UDP header to achieve a header size of 20 bytes. The rest of the features are the first n bytes of the application presenting the data of the application layer.

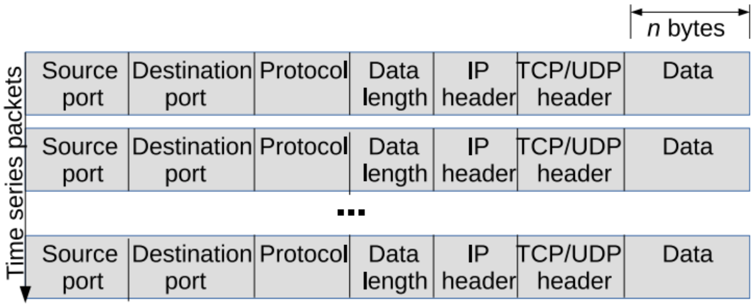


Figure 2.5: Features extraction
[40]

They developed a the deep learning method uses LSTM to maintain the time dependence of receiving network traffic through time series analysis of network application traffic. Their network model includes three hidden layers: LSTM (128 neurons), two Fully connected Layers with 128 and 12 neurons. And the last layer is Softmax layer with 12 neurons. The results received are impressive with accuracy 98% which less complicated than other network like CNN+LSTM (Feature size = 1000,accuracy = 96%) or with SAE (Feature size = 1500,accuracy = 97%). With features size very small when compare to each other network (Feature size = 55).

Otherwise, Wang, Pan, et al [41], proposed encrypted data classifiers (denoted as DataNets) based on three deep learning schemes, i.e. MLP, Stacked Auto Encoder (SAE) and Convolution neural networks, using an open dataset that has over 195000 encrypted data samples from 15 applications. MLP based Datanet consists of one Input layer, two Hidden Layers and one Output Layer. The Output layers computed by 15 Neurons with classifier Softmax. And cross entropy is used as loss function. SAE architecture have one input layers, three stacked encoders layers and one output layers (contains Softmax classifier inside). SAE can be used for reduction dimension or feature extraction. We will go in details SAE schemes with two figures (2.6 and 2.7) to understand SAE based Datanet deeper, how it can be trained. With input Packet Bytes Vector (PVB) with 1480 neurons in first hidden layer, encoder 1 with 740 neurons, after that they applied ReLu activation function, the results are the input for the next hidden layer. Encoder 2 with 92 neurons. ReLu are applies as follow and the third layer, encoder consist 32 neurons. Final is fully connected layer with Softmax Classifier give the final results. For training, it has two steps: training the encoders and training the classifier. For training Encoder, they do as follows, compute the reconstruction function errors between input and output using mean squared errors by these functions:

$$\epsilon(k) = \frac{1}{m}\sum_{i=1}^{m} c_j^2(k) \tag{2.5}$$

where $e_j(k) = \hat{y}_j(k) - y_j(k)$ is the error between the output and targeted value and m is the number of samples. They updated the weight with this equation:

$$\triangle w_j(i) = -\eta \frac{\partial \epsilon(k)}{\partial z_j(k)} z_i(k) \tag{2.6}$$

where z_i is the output of previous neuron and η is learning rate.

And the last will be CNN for encrypted packet classifiers. It consists of three convolution layers, 2 Max-Pooling layers and a fully-connected layer with Softmax as classifier. DataNet is the core to the SDN-HGW (Software Defined Networks-Home Gateway) can use for Smart-Home Network. The results are quite good, approximately more than 95% with precision, recall and F1-score. Experiments on full and balance Dataset.

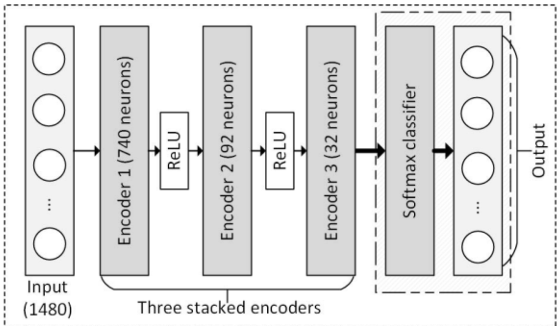


Figure 2.6: Overview SAE
[41]

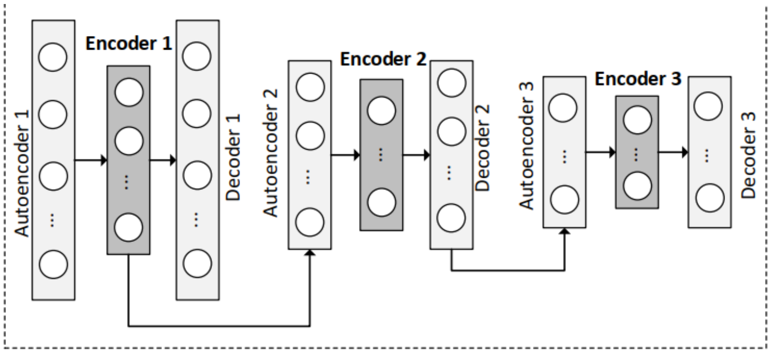


Figure 2.7: Encoder training
[41]

Lotfollahi, Mohammad, et al [24] also using CNN and SAE in Deep Packet Framework. It can handle both application identification and traffic characterization tasks. The special thing here is that Deep Packet can skip the feature extraction phase. There are six step as follows: Data-link layer removal, Transport header modification, Irrelevant Packet rejection, Byte Convention, Truncation, Normalization and IP masking. For the architecture of SAE, we can see it can be implemented with five fully-connected layers,which one of them have 400, 300, 200, 100 and 50 neurons in order, with drop out rate is 0.05 for preventing over-fitting. At a final layer, Softmax classifier is used with 17 and 12 neurons are added. Parallel to that is the CNN model with two Convolution layers and followed by a Max-Pooling. After that two dimensional tensor is compressed into a one-dimensional vector and fed into three layers of Fully connected, final as usual is Softmax Classifier. For train-

Class Name	**CNN**			SAE		
	Rc	Pr	F1	Rc	Pr	F1
Chat	0.71	0.84	0.77	0.68	0.82	0.74
Email	0.87	0.96	0.91	0.93	0.97	0.95
File Transfer	1.00	0.98	0.99	0.99	0.98	0.99
Streaming	0.87	0.92	0.90	0.84	0.82	0.83
Torrent	1.00	1.00	1.00	0.99	0.97	0.98
VoIP	0.88	0.63	0.74	0.90	0.64	0.75
VPN:Chat	0.98	0.98	0.98	0.94	0.95	0.94
VPN:File Transfer	0.99	0.99	0.99	0.95	0.98	0.97
VPN:Email	0.98	0.99	0.99	0.93	0.97	0.95
VPN:Streaming	1.00	1.00	1.00	0.99	0.99	0.99
VPN:Torrent	1.00	1.00	1.00	0.99	0.99	0.99
VPN:VoIP	1.00	1.00	1.00	1.00	0.99	0.99
Average	0.94	0.93	0.93	0.92	0.92	0.92

Table 2.2: Traffic characterization results
[24]

ing, Adam optimizer was used with Mean Square Error as loss function to train SAE. And for CNN, Adam and Categorical cross entropy was used. The results are good with traffic through VPN (more than 95% with application identification and 92% with traffic characterization). We can see this table 2.2 above for more details about traffic characterization results. But with Tor traffic,Deep Packets was unable to classify the inherent Tor's traffic accurately. Because Tor encrypts its traffic before transmit it. Therefore, we can see that this method still has certain limitations when classifying traffic through some other encrypted ways.

Ran, Jing, Yexin Chen and Shulan Li [31], also proposed method based on CNN, but not regularly, this is 3D-CNN. Pre-processing step here is quite important for this model as the input must be 3D for this network. First, traffic flows are identified and divided into individual files. After they extract the first n packet of each flow. After that, they trimmed all packet to fixed length, deleted the redundant part. If the size of packet is too small, add zeros for them. Each packet is converted into a 2D image using one-hot encoding. Finally, images of the same stream combine together a 3D input file. 3D CNN fits data with both space and time features. The kernel is 3D, and small kernel can reduce network complexity without changing performance. For more details, we can look at figure 2.8 below. The architecture of this 3D-CNN contains four Convolution layers with ReLu as activation function, two max pooling layers and two dense layers with 1024 and 10 neurons, final is Softmax layers as classifier. The Optimizer and learning rate they used is Adam Optimizer and 0.0001. The results received is very good with most of specific case get F1 score above 99%, despite two malware classes get 94% and 95%, but it is no problem and acceptable.

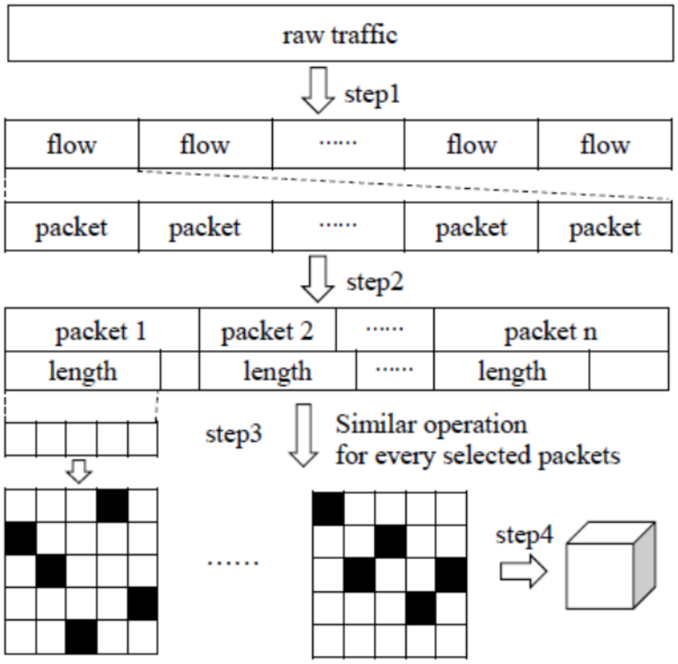


Figure 2.8: Pre-processing of raw data
[31]

And a very familiar and well-known model in image processing has also been used in this field of traffic classification, it is Generative adversarial network(GAN). The main idea of GAN is that two networks, the generator network and discriminator network, both networks are used simultaneously to optimize and find the most optimal convergence point. Vu, Ly, Cong Thanh Bui, and Quang Uy Nguyen [39], proposed auxiliary classifier GANs (AC-GAN) is used to generate composite templates for supervised network classification task. The main difference between AC-GAN and GAN is that

AC-GAN has both random noise and input class label to create sample of input class label. Another different is the output of Discriminator D including probability distribution over sources L_S and over class label L_C:

$$L_S = E[logP(S = real|X_{real})] + E[logP(S = fake|X_{fake})] \tag{2.7}$$

and

$$L_C = E[logP(C = c|X_{real})] + E[logP(C = c|X_{fake})] \tag{2.8}$$

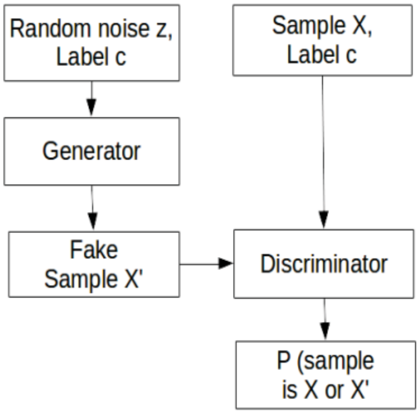


Figure 2.9: Training process AC-GAN
[39]

Discriminator D is trained to maximized $L_S + L_C$ and Generator G is trained to maximized $L_C - L_S$. In the training process described in Figure 2.9, the generator takes inputs as random noise z and label c and gives the output of fake samples while the discriminator has inputs as a real sample data or a fake sample data. The objective of training discriminator is making the probability of a sample data being a real sample or a fake sample closely to 0 or 1 to make discriminator able to recognize between real samples and fake samples. However, the purpose of network G training is to create samples that are close to the real samples in the data set or the output probability of D equals 0.5. In their experiment, G and D networks have two hidden layers and set the learning rate as 0.001. They use a public Dataset named Network Information Management and Security group(NIMS) with two classes, SSH and non-SSH are generated from the applications, and 22 statistical features for the classifier input. In that paper, for their purpose, they re-labeled NIMS

Method	**Acc Score**	**F1 Score**
SVM	0.9873	0.5909
DT	0.9976	0.9482
RF	0.9978	0.9485
SVM+AC-GAN	0.9878	0.6050
DT+AC-GAN	0.9978	0.9552
RF+AC-GAN	**0.9989**	**0.9543**

Table 2.3: Comparing results
[39]

dataset into two classes as SSH and non SSH. Totally in the dataset includes 35454 SSH flows and 678395 non-SSH flows. For the results, we can the table 2.3.

Look up the table, we can see the model applied RF(Random forest) and AC-GAN achieved best result when comparing with other model, 99% with accuracy score and 95% with F1 Score. In addition, GAN has been applied in IDS and malware detection to create detrimental attacks to cheat and evade detection systems.

And the most recent, in March 2019. Zeng, Yi, et al [47] proposed Deep-Full-Range(DFR). Which is a light-weight framework with the aid of deep learning for traffic application and intrusion detection. Three DL algorithm are employed CNN, LSTM and SAE (See figure 2.10). With the relevant full-range structure, DFR is able to categorize encrypted traffic and malware traffic in a frame without human assistance and private details.

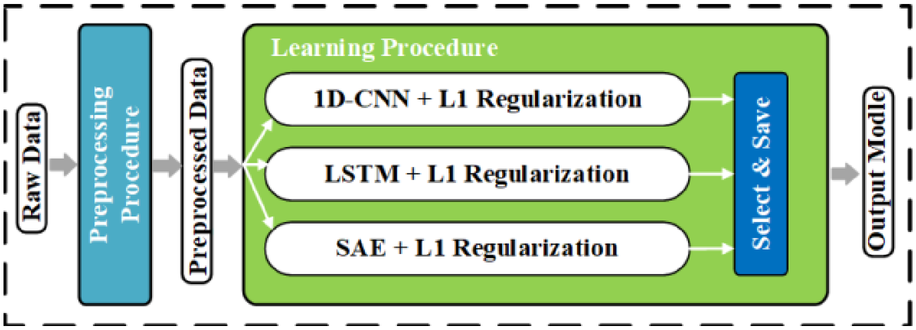


Figure 2.10: Deep-Full-Range Framework
[47]

DFR framework have two main procedure: Pre-processing and DFR. Pre-processing procedure is implemented by five steps as follow: Packet Generation, Traffic purification, Traffic refiner, Length Unificaiton. And last one is IDX Fill Generation will convert pcap files to IDX format files as input data for next learning procedure. In the next part of procedure, it contains CNN, LSTM and SAE. For CNN, it has two Local Response Normalization(LRN), different with other network, are added to punish abnormal reactions or to classes for better generalization. Responded-normalized activity equation:

$$b^i_{x,y} = a^i_{x,y}/(k + \alpha \sum_{j=max(0,i-n/2)}^{min(N-1,i+n/2)} (a^j_{x,y})^2)^\beta \tag{2.9}$$

When $a^i_{x,y}$ denoted as the activity of a neuron computed by applying kernel i at position (x, y). N is the number of total kernel in the layer. The constants k, n, α, and β are hyper-parameters whose values are determined using a validation set.

The second is LSTM constructed of three layers with 256 LSTM cells. And they also applied dropout for all layer of LSTM for better generalization After that, it will go through Softmax classifier. And the last one of DFR's core is two SAEs. The first encoder has 1000 neurons which are densely connected with 900 inputs and 900 outputs. The second encoder has 1500 neurons. They are trained separately, SAE2 trained by reducing the variance between input 2 and output 2. SAE2 will be trained after finish SAE1 and stored in DFR with sigmoid function. The results show that the DFR framework can accomplish a much accurate and robust classification result than ML-Based C4.5 method and DL-Based 1D-CNN method. The averaging F1 score of the DFR framework is 99.87%,an impressed number.

As I have mentioned above, a number of papers appeared and introduced giving us an overview of the current use of Neural Network (NN) and Deep Learning (DL) in the field of encrypted network traffic classification task. Undoubtedly, NN and DL are absolutely effective tools that can help us solve these problems. With new developments, each result is gradually improved. Each model shows its effectiveness MLP, CNN, 3D-CNN, RNN, LSTM, SAE,etc. But the most important thing is that each approach has its own advantages and disadvantages. Hence, it is difficult to choose a particular method to implement an traffic classification in some specific case. To the best of my knowledge, I am not the first use this Deep learning approach(CNN+LSTM) to classification encrypted traffic. However, with what

I have consulted and learned, I will try to optimize my model to further improve the accuracy.

Chapter 3

Networking Background

In this chapter, we will discuss computer networking theory. The main goal of this chapter is to provide a brief overview of key networking protocol components. The concepts introduced in this chapter are the premise for construct the methodology in the following chapters.

3.1 OSI

Open System Interconnection (OSI) is a basic model of communication processes, establishing network architecture standards at an international level, which is a common basis for different systems connect together [30], [20]. The OSI model organizes communication protocols into 7 layers Each of which addresses a narrow part of the communication process, divides the communication process into multiple layers, and in each layer can have many different protocols implement specific communication purpose.

3.1.1 Protocols in OSI

There are two types of protocols used in the OSI model: Connection - Oriented and Connection-less.

- Connection-Oriented: Before transmitting data, the cascading entities in two systems must establish a logical link. They negotiate with each other the set of parameters that will be used during the data transfer phase. The data is transmitted with error control, data flow control mechanisms to improve the reliability and efficiency of data transmission. After the exchange is complete, the link will be deleted.

- Connection-less: Data is transmitted independently on different routes. With Connection-less protocols there is only a single phase is data transmission.

3.1.2 OSI Layers

Layer 7 (Application Layer): Define the interface between the user and the OSI environment, including many application protocols that provide facilities for users to access the network environment and provide distributed services. When the application entity is set, it will call Application Service Elements. Application elements are coordinated in the application entity environment through links called Single Association Object (SAO). SAO will control communication and allow serialization of communication events

Layer 6 (Presentation Layer): Solve the problems related to the syntax and semantics of the transmitted information. Represent user information in accordance with the working information of the network and in opposite case. This layer is responsible for converting network data from one representation type to another.

Layer 5 (Session Layer): The session layer allows users on different machines to establish, maintain, cancel and synchronize communication sessions between themselves. In other words, this layer establishes transactions between the top and bottom entities.

Layer 4 (Transport Layer): The top layer is related to data exchange protocols between open systems, controlling end-to-end data transfer. The transport layer divides large packets into smaller ones before sending them and numbers the packets to ensure they will be transmitted in order. The last layer responsible for the level of security in data transmission, transport layer protocol is highly dependent on the nature of the network layer.

Layer 3 (Network Layer): Performs routing functions for packets in the same network or different networks. Another important purpose of this layer is the congestion control.

Layer 2 (Data Link Layer): The main function of this layer is to carry out links, maintain and remove data links. Error control and traffic control.

Layer 1 (Physical Layer): The lowest layer in the OSI model. Entities communicate with each other over a physical link. The physical layer identifies functions, procedures on electricity, mechanical, and optical to maintain, activate and release physical connections between network systems. Ensure switching requirements work to create real links for information bits of information.

3.2 TCP/IP model

TCP/IP (Transmission Control Protocol/Internet Protocol is a stack of protocols that work together to provide inter-network media. In 1981, TCP/IP version 4 (IPv4) was completed and commonly used on computers using UNIX operating systems, becoming into one of the basic protocols of the Windows operating system at the time. In 1994, a new version of IPv6 was created on the basis of improving the limitations of IPv4.

3.2.1 Layers of TCP/IP

Application Layer: Equal to session, presentation, and application layers in the OSI model. The application layer supports applications for transport layer protocols. Provides interface for users of TCP / IP models. Application protocols include TELNET (remote access), FTP (file transport), SMTP (email)...

Transport layer: According to the transport layer in the OSI model, this layer makes connections between two servers on two networks using two protocols: The Transmission Control protocol and the User Datagram protocol. Details of this protocol will be introduced in the next section.

Internet Layer: Similar to Network Layer in the OSI model, the network layer provides a logical address for the network physical interface. The network layer implementation protocol is the Connection-less protocol, which is the operating core of the Internet. Along with routing protocols such as RIP, OSPF, BGP, IP network layer allows flexible connection of different types of ”physical” networks such as Ethernet, Token Ring... In addition, this layer supports physical address mapping (MAC) provided by the Network Access Layer.

Network Access Layer: Corresponding to the physical layer and data link in the OSI model, the network access layer provides the physical means of connecting cables, adapters, network cards, and connectivity protocols. The Internet segments data into frames.

The following figure 3.1 is an overview of the OSI model and TCP/IP model.

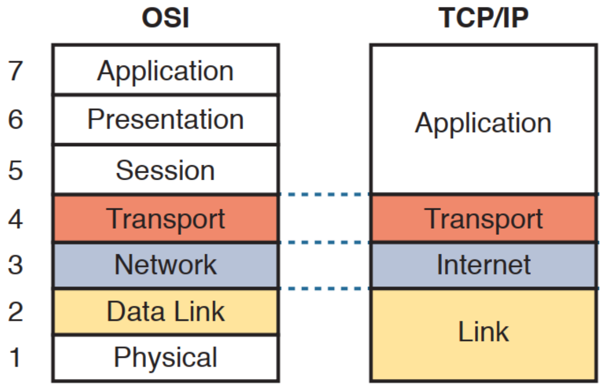


Figure 3.1: OSI and TCP/IP model
[29]

3.2.2 Basic Protocols in TCP/IP model

User Datagram Protocol(UDP)

UDP is a Connection-less protocol. UDP used for processes that do not require high reliability, no acknowledgement (ACK) authentication mechanism, does not guarantee delivery of data packets to the destination and in the correct order and does not perform, eliminate duplicate packets. It provides a mechanism for assigning and managing port numbers to uniquely identify applications that run on a network client and perform multiplexing. UDP is often used in conjunction with other protocols such as SMNP and VoIP.

- SNMP(Simple Network Management Protocol):is a popular, highly compatible network management protocol. SNMP provides administration information and supports Agent management.

- VoIP(Voice over IP):The system ensures real-time features, high transmission rates, voice packets without excessive delay and high reliability.

Transmission Control Protocol(TCP)

TCP is Connection-Oriented protocol, it is mean before transmitting data, the TCP entity transmits and the TCP entity negotiates to establish a temporary, logical connection that exists during the data transmission. TCP receives information from the upper layer, splits the data into multiple packets at specified lengths, and passes the packets down to network layer protocols for routing. The TCP processor verifies each packet, without acknowledgment, the packet will be re-transmitted. The receiving TCP entity will restore the original information based on the packet order and move data upstream. TCP provides the ability to transfer data securely between internetwork components. Provides functions for checking the accuracy of data on arrival and re-transmission of data when an error occurs. The format and content of a TCP segment is as figure follow:

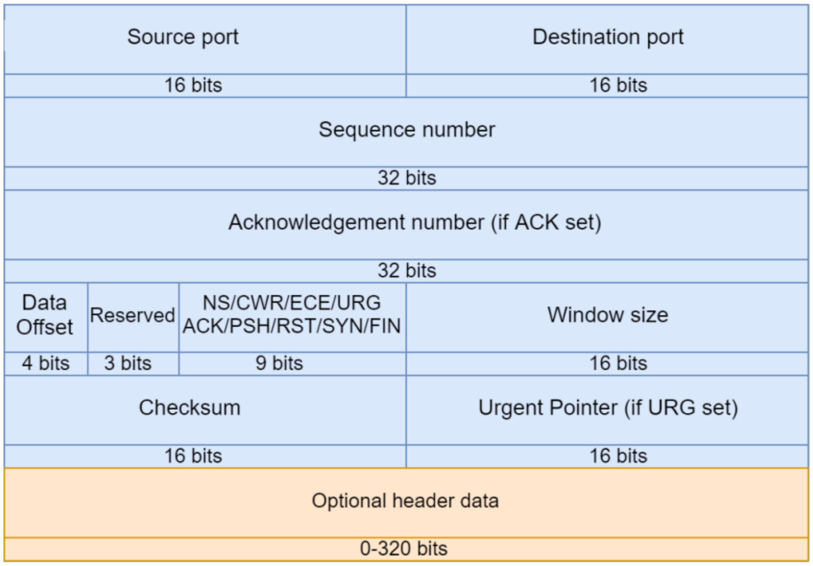
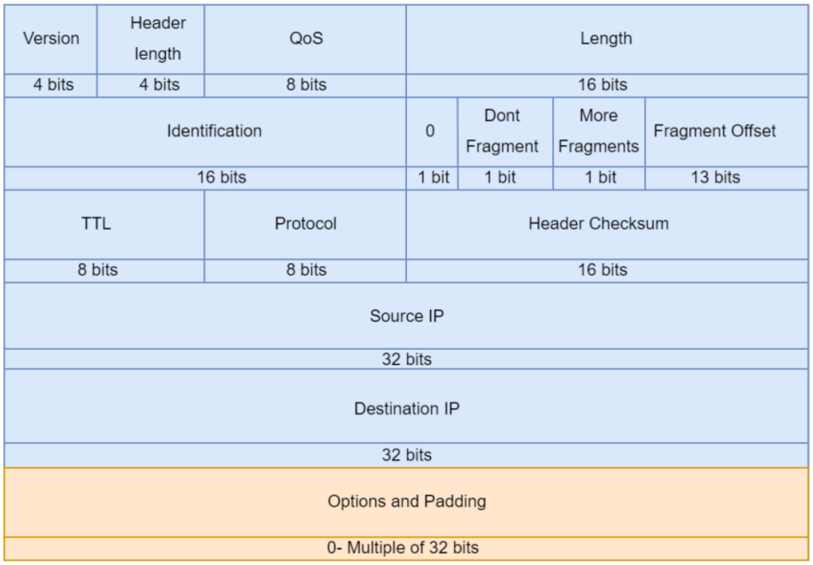


Figure 3.2: TCP Segment
[34]

From figure 3.2 we see the layout of the TCP header where some of the most important fields are the source and destination ports, as they are used to uniquely identify the flow that the traffic. Other fields of interest are the checksum field and the window size. The checksum field is used to check for corruption of the TCP segment. The window size indicates the maximum amount of traffic, the sender can send before having to wait for an acknowledgement (ACK).

Internet Protocol(IP)

Internet protocol is connection-less protocol. The primary function of IP is to provide Datagram service and the ability to connect sub-nets into inter-locations to transfer data with the Datagram IP packet switching method, perform the address determination and routing process. The IP Header is added to the top of the packets and transmitted by the lower layer protocol in the form of data frames. IP routes packets over the network using dynamic routing tables at each hop. IP performs the removal and recovery of packets according to the size requirements defined for the physical layer and data link implementation. IP checks for control information errors using the checksum value.



<table>
<tr><td>Version</td><td>Header length</td><td>QoS</td><td colspan="4">Length</td></tr>
<tr><td>4 bits</td><td>4 bits</td><td>8 bits</td><td colspan="4">16 bits</td></tr>
<tr><td colspan="3">Identification</td><td>0</td><td>Dont Fragment</td><td>More Fragments</td><td>Fragment Offset</td></tr>
<tr><td colspan="3">16 bits</td><td>1 bit</td><td>1 bit</td><td>1 bit</td><td>13 bits</td></tr>
<tr><td colspan="2">TTL</td><td>Protocol</td><td colspan="4">Header Checksum</td></tr>
<tr><td colspan="2">8 bits</td><td>8 bits</td><td colspan="4">16 bits</td></tr>
<tr><td colspan="7">Source IP</td></tr>
<tr><td colspan="7">32 bits</td></tr>
<tr><td colspan="7">Destination IP</td></tr>
<tr><td colspan="7">32 bits</td></tr>
<tr><td colspan="7">Options and Padding</td></tr>
<tr><td colspan="7">0- Multiple of 32 bits</td></tr>
</table>

Figure 3.3: IP Datagram
[34]

The figure 3.3 above shows the IP Datagram structure. Each datagram has a header section containing control information. If the IP address is in the same network as the source station, the packets will be directed to the destination,if the destination IP address is not in the same IP network as the source machine, the packets will be sent to an IP Gateway relay server for forwarding. The IP Gateway is an IP network device that handles the transfer of IP data packets between two different networks.

3.3 Session/Flow

When dealing with network traffic it is necessary to distinguish between a session, a flow and an individual packet [8]. A network flow is a unidirectional packet stream from one host to another. A flow is uniquely defined by a five tuple:

$Source_{IP}$,$Destination_{IP}$,$Sourceport$,$Destinationport$,$protocoltype$.

As the session terminology varies between sources, we define a session as a bidirectional packet stream between two hosts. As a session can contain both directions of flows it means that the source and destination ports are interchangeable.

Chapter 4

Neural Network theory

This chapter discusses the basic knowledge of neural networks as well as the two network architectures that I will apply to the method of classifying encrypted network traffic: CNN and LSTM.

4.1 Neural Network

4.1.1 Introduction

The artificial neural network (ANN) is an information processing model adapted from the information processing method of the biology neurons systems. It is made up of a large number of connected neurons together through direct link and passing along the connections and calculating new values at the nodes [12], [13]. An artificial neural network is configured for a specific application (pattern recognition, data classification, etc.) through a process of learning from a set of training patterns. Essentially, learning is the process of correcting the weights between neurons. They can be used to model complex relationships between input and results or to search for patterns in data. Neural networks help us cluster and classify. You can think of them as a clustering and classification layer on top of the data you store and manage. They help to group unlabeled data according to similarities among the example inputs, and they classify data when they have a labeled dataset to train on.

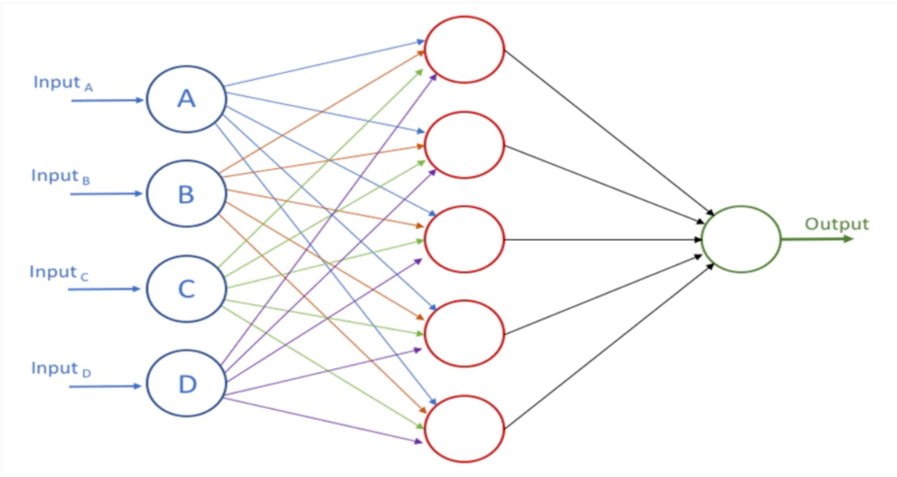


Figure 4.1: Simple Neural Network
[37]

4.1.2 Neurons

Artificial neurons are the processing units of a neural network. A neuron consists of inputs, weights, a bias, and activation functions all used to calculate an output. Each input multiplied by the weight represents the importance of that input to the output of the neuron.

4.1.3 Architecture

Neural Network (NN) is a combination of perceptron layers, also known as multilayer perceptrons. A basic NN structure will have 3 types of layers:

- Input layer: The leftmost layer of the network representing the network's inputs.
- Hidden layer: The layer located between the input layer and exit layer represents the logical reasoning of the network.
- Output layer: The rightmost layer of the network represents the outputs of the network.

Note that a NN has only one input and one output layer but may have many hidden layers. In NN, each node is a neuron, but their activation function may be different. However, in reality, people often put them in the same form to calculate for convenience . At each level, the number of nodes (neurons) can vary depending on the problem and the solution. But often when working people put hidden layers with an equal number of neurons. In addition, neurons in the layers are often linked together to form a full-connected network.

4.1.4 Loss Function

The lost function, also known as the cost function, is like a form of getting the model to pay a penalty every time it predicts a mistake, and the number of fines is proportional to the severity of the error. In every supervised learning problem, our goal is always to minimize the lost function near or equal to 0. When training, the loss function is usually calculated with the formula equal to the average of the total error of each prediction:

$$L = \frac{1}{N} \sum_{i} L_i \tag{4.1}$$

Working with the loss function of a multi-layer network usually involves a loss function with a non-quadratic, non-convex surface and a height with many local minimum and saddle points. This has led to the development of techniques such as batch learning and Stochastic Gradient Descent (SGD) to help ensure better and faster convergence.

4.1.5 Activation function

Neural network activation functions are an important component of deep learning. Activation functions determine the output of a deep learning model, its accuracy and the computational effectiveness of training a model that can create or break a large neural network [1].

Activation functions also have a major influence on the ability of the neural network to converge and the speed of convergence, or in some cases, the activation functions may prevent the neural networks from converging for first place. Activation functions are commonly used such as: Tanh, Sigmoid, ReLu,Leaky-ReLu.Table 4.1 is an overview for some activation function.

Function	**Equation**	**Range**	**Characteristic**
Sigmoid	$\frac{1}{1+e^{-x}}$	(0,1)	Smoth Gradient,Clear Predictions
TanH	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	(-1,1)	Like Sigmoid,Zero Centered
ReLU	$\begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}$	[0,∞)	Computationally Efficient
Gaussian	$f(x) = e^{-x^2}$	(0,1]	Differentiable, Non-Negative
SoftMax	$f(x) = \frac{e^{x_i}}{\sum_{j=1}^{J} e^{x_j}}$	(0,1)	Able to handle multi class Useful for output neuron

Table 4.1: Activation Function
[1]

4.1.6 Learning rate

Learning speed is a super parameter that controls how much a model changes in response to an estimated error every time the model weight is updated. Choosing learning rates is a challenge because values that are too small can lead to lengthy training that can be stuck, while values that are too large can lead to learning an optimal sub-weight set too quickly or the training process is not stable. The learning rate may be the most important hyper parameter when configuring your neural network. Therefore it is vital to know how to investigate the effects of the learning rate on model performance and to build an intuition about the dynamics of the learning rate on model behavior.

4.2 Convolution Neural Network

4.2.1 Introduction

Convolution Neural Networks(CNN), sounds like a weird combination of biology and math, but these networks have been some of the most influential innovations in the field of computer vision [10]. 2012 was the first year the neural network developed prominently when Alex Krizhevsky [19] used them to win the competition that year. Ever since then, a host of companies have been using deep learning at the core of their services. In recent years, we have witnessed many remarkable achievements in the field of Computer Vision. Large image processing systems such as Facebook, Google or Amazon have put in their products intelligent functions such as facial recognition, selfdriving car development, and automated delivery drones. The idea is simplified as you give the number array to computer and it will output features that describe the probability of the image being a definite object (for motorcycles, aircraft,ships etc.) The CNN architectures consist of three main Layers: Convolution Layer (ConV Layer), Pooling Layer and Fully Connected Layer (FC). Stack these layers to form a full CNN architecture. Note that some layers contain parameters and other do not have.

4.2.2 Architecture

Lets dive into details for each layers.

First is the **Convolution (ConV) Layer**: Convolution is first used in digital signal processing (Signal processing). Thanks to the principle of information conversion, scientists have applied this technique to digital photo and video processing . To make it easier to imagine, we can consider convolution as a sliding window imposed on a matrix. You can follow the mechanism of convolution through the figure 4.2 below. On the left side is input image with size 5x5. Applied kernel (size 3x3x1) slide forward and from left to right. Kernel represented in color yellow. The output value due to the product of these components adds up. The result of convolution is a convoluted feature that results from sliding the filter matrix and performing convolution on the entire image matrix on the left.

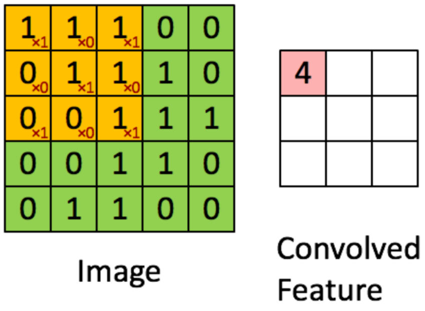


Figure 4.2: Filter apply
[33]

Second is **Pooling Layer**: In addition to the previously described convolution classes, the Convolution neural network also contains pooling layer. Pooling layer are often used after the convolution layer. What pooling classes do is simplify the information at the output from the convolution layers. For example, each unit in the pooling layer could collapse a region of $2*2$ neurons in the previous layer. A common pooling procedure is max-pooling. In Max-pooling, a pooling unit is simply an output that activates the maximum value in a $2*2$ input area, as shown in the figure 4.3 below. Max-pooling is not the only technique used for pooling. Another method is called L2 pooling. Here, instead of taking the maximum activation value of a $2 * 2$ neuron region, we take the square root of the sum of squares of activation in the $2 * 2$ region. While the details are different, the intuition is similar to max-pooling: L2 pooling is a way to condense information from convolution layers. And with the dimension reduction we can see that it serves two main purposes. The first is the number of weight parameters that can be reduced, thus reducing the computational cost. The second is that it will control over-fitting.

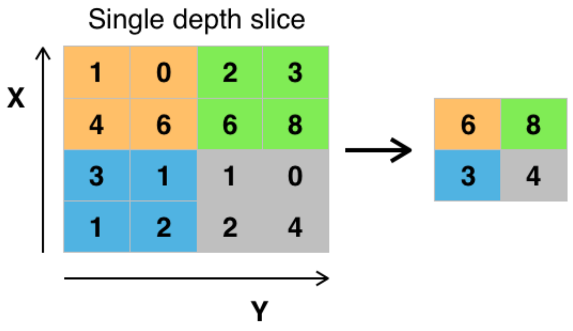


Figure 4.3: Max-pooling example
[9]

The third is **Fully Connected Layer**: Adding a fully connected layer is a common way to learn nonlinear combinations of advanced features as indicated by the output of the ConV layer.

LeNet-5 is a convolution neural network structure proposed by Yann LeCun et al [22] in 2015. In general, LeNet refers to lenet-5 and is a simple Convolution neural network This is the most typical example of a basis CNN network. The figure below show us the architecture of this model.

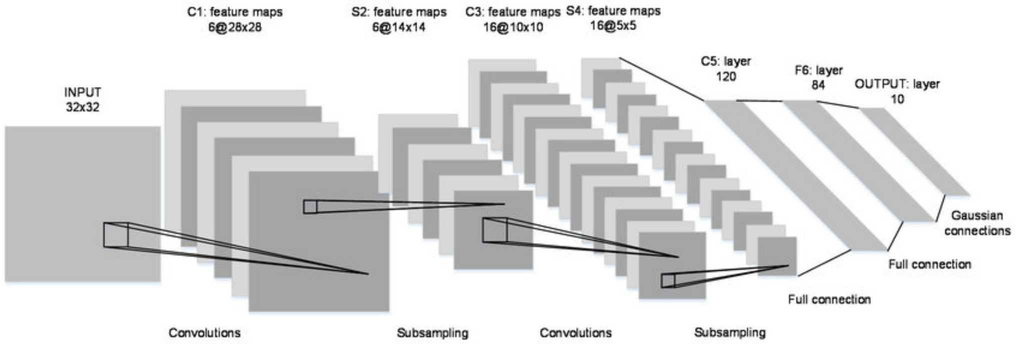


Figure 4.4: LeNet5 by Yann Lecun
[22]

4.3 Long Short Term Memory

4.3.1 Introduction

Vanishing Gradient Descent (when GD disappear or become zero) and Exploding Gradient Problems are common problems in using Recurrent Neural Network (RNN). Theoretically, RNN can carry information from the first layer to the next layer, but the fact is that information can only be carried over a certain number of states [2]. From that thought, people develop architectures to overcome the disadvantages of RNN. These are LSTM and GRU [4].

4.3.2 Structure

LSTM structure consists some components:

- The core component of LSTM is Cell state. Cell state is a conveyor-like form. It runs through all links (network nodes) and only has a linear interaction. So that the information can easily be transmitted smoothly without fear of being changed.

- Forgotten Gate: This gate determines which information in the current memory is kept and which is left. Input information is entered into the sigmoid function. The output of this function acts as a mask to filter information from the cell state.

- Input Gate: This gate is used to update memory with new information. There are 2 sigmoid and Tanh functions here. Their effect is the same. The output from the sigmoid function will filter the processed information from the Tanh output.

- Output Gate: This gate determines the output of the current word. It gets information from two sources: current cell state and input. The cell state after modification will go through the Tanh function and the current input is passed through the sigmoid function. From here we combine the two results above to get the output. Notice that both the output and the cell state are included in the next step.

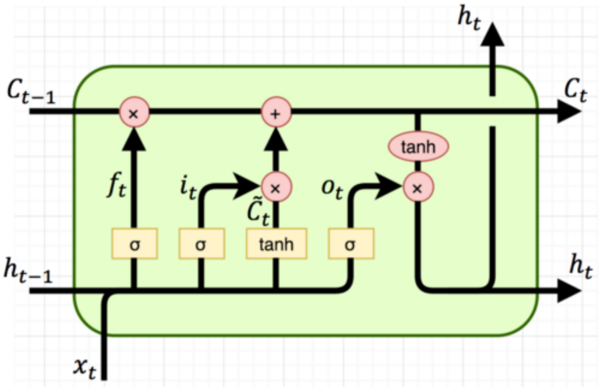


Figure 4.5: Overview LSTM structure
[2]

4.3.3 Mechanism of LSTM

Cell internal state C_t and output h_t can be calculated by following some steps:

1. In the first step, the LSTM cell decides what information needs to be removed in the internal state cell at the time step before C_{t-1}. Activation value ft of the forgotten gate at time t is calculated at the current input value x_t, the h_{t-1} output value from the LSTM cell in the previous step and the bias b_f of the forgotten gate.

The sigmoid function converts all f_t to a range of values from 0 (completely forgotten) and 1 (completely memorized). The formula for this step is:

$$f_t = \sigma(W_{f,x}x_t + W_{f,h}h_{t-1} + b_f) \tag{4.2}$$

2. Second step is LSTM cell decides what information needs to be added to the cell internal state C_t. This step involves two calculations for $\tilde{C_t}$ and f_t. The Candidate value $\tilde{C_t}$ representing the potential information to be added to C_t is calculated as follows:

$$\tilde{C_t} = tanh(W_{\tilde{C},x}x_t + W_{\tilde{C},h}h_{t-1} + b_{\tilde{C}}) \tag{4.3}$$

Activation value i_t of input gate also calculated as follow:

$$i_t = tanh(W_{i,x}x_t + W_{i,h}h_{t-1} + b_i) \tag{4.4}$$

3. Third step :The new value of C_t is calculated from the result obtained from the steps with Hadamard product for each element, denoted by $\circ$

$$C_t = f_t \circ s_{t-1} + i_t \circ \tilde{s}_t \tag{4.5}$$

4. Last step:The h_t output value of the LSTM model is calculated based on the following two formulas:

$$o_t = \sigma(W_{o,x}x_t + W_{o,h}h_{t-1} + b_o \tag{4.6}$$

And

$$h_t = o_t \circ tanh(C_t) \tag{4.7}$$

4.3.4 Gated Recurrent Unit

Another variant of LSTM is the GRU model, Cho et al [6] in 2014. This model combines a forgotten gate and an input gate to form an update gate(z_t and r_t). It has no cell state, only output h_t that is used to make decisions and to inform the next steps. Due to the loss of the cell state, the effects of the two gates are quite difficult to distinguish clearly, it can be said that both gates have the effect of filtering information from the input of the cell to give a satisfactory output. Both criteria are keeping historical information and being able to make current decisions most accurately.

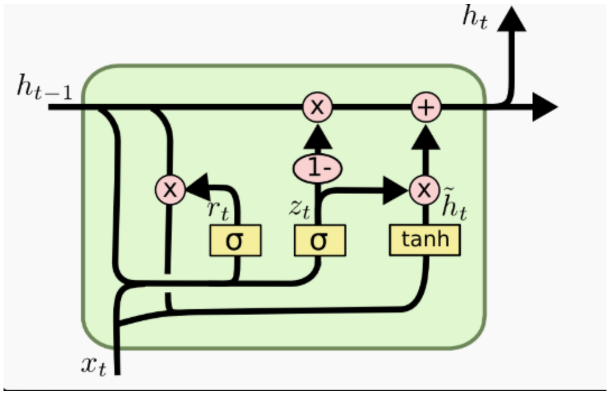


Figure 4.6: GRU structure
[2]

Some formulas were changed as follows:

$$z_t = \sigma(W_z.[h_{t-1}, x_t]) \tag{4.8}$$

$$r_t = \sigma(W_r.[h_{t-1}, x_t]) \tag{4.9}$$

$$\tilde{h}_t = tanh(W.[r_t \circ h_{t-1}, x_t]) \tag{4.10}$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t \tag{4.11}$$

This figure 4.6 above the structure of GRU. In terms of efficiency, it is difficult to draw accurate conclusions, but it can be said that the prevalence of LSTM is superior to that of GRU. Due to reduced complexity, GRU operates a little faster and simpler than LSTM.

Chapter 5

Methodology

5.1 Dataset

The dataset set I used in my work is VPN-nonVPN (ISCXVPN2016) [11]. Well-known data set for use to categorize unencrypted (non-VPN) and encrypted (VPN) traffic. This data set was created by the University of Brunswick and is about 28Gb in total size with Pcap and Pcapng format. Dataset were collected in their research labs. They used Wire Shark and Tcpdump to capture traffic. For the VPN, they used an external VPN service provider and connected to it using Open-VPN. They created 2 accounts for Bob and Alice to use network services like Youtube, Spotify or Chrome. Table 5.1 below are detailed information about the types of traffic that appear in this dataset.

The ISCX VPN-nonVPN dataset originally contained seven types of normal encrypted traffic and seven types of traffic encapsulated protocols. Moreover, this data set contains captured traffic of the Tor software. This flow is probably generated while using the Tor browser and it has labels like Twitter, Google, Facebook, etc. Tor is an open source, free Software developed for anonymous communications. The purpose is to focus on classifying encrypted traffic as well as identifying applications. Relabeled is needed when the current raw traffic in the dataset is unlabeled. I have categorized and relabeled the dataset, and formed 12 classes, corresponding to each class being the traffic of related applications. Details can be found in table 5.2.

Traffic	Description
Browsing	Summary of HTTPs traffic generated by users when they use web browsers to access information on the network.
Email	Sample traffic is generated by mail transfer between two accounts Alice and Bob via SMTP protocol.
Chat	Chat data is taken through the use of Facebook and Hangouts on website,and also Skype application.
Streaming	Captured from Youtube through Firefox and Chrome
File Transfer	Traffic refers to the exchange (send and receive) of files and documents, such as FTP over SSH or FTP over SSL.
VoIP	The Voice over IP label groups all traffic generated by voice applications.
P2P	To generate this traffic, they use Bittorrent to download .torrent files and capture those processes with uTorrent.

Table 5.1: Traffic Description

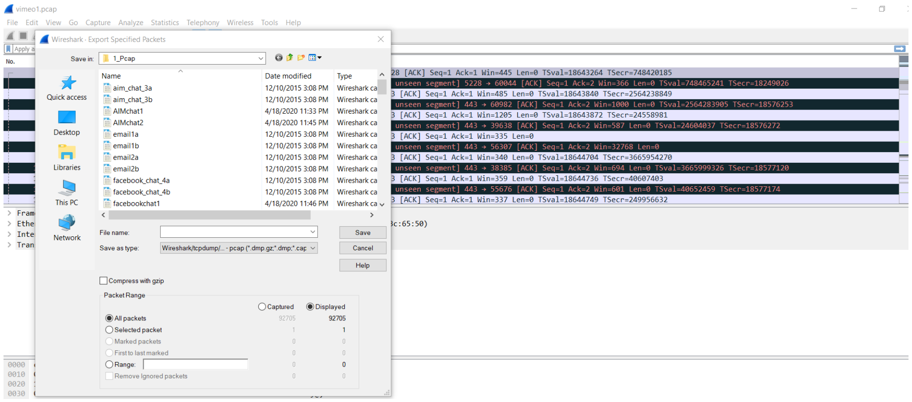
Class Name	Application	***Size(No.Packets)***
Chat	ICQ,AIM,Skype,Facebook,Hangouts	141179
Email	Email,Gmail(POP3,IMAP,SMTP)	56941
FileTransfer	FTPS,SFTP	158967
Torrent	uTorrent,BitTorrent	108541
Streaming	Vimeo,Youtube,Netflix,Spotify	250075
VoIP	Facebook,Skype,Hangouts,Voipbuster	949736
VPN:Chat	VPN-(ICQ,AIM,Skype,Fb,Hangouts)	86226
VPN:Email	VPN-(Email,Gmail)	21581
VPN:File Transfer	VPN-(FTPS,SFTP)	378089
VPN:Torrent	VPN-(uTorrent,BitTorrent)	40101
VPN:Streaming	VPN-(Vimeo,Youtube,Netflix,Spotify)	150939
VPN:VoIP	VPN-(Fb,Skype,Hangouts,Voipbuster)	2413699

Table 5.2: Traffic labeled

5.2 Preprocessing

The main purpose of this pre-processing is to optimize the size of the data set as well as create the IDX format file as input for the Neuron network model mentioned in the next section. Pre-processing contains some steps:

1. **The first step** is to explore the traffic files (.pcap or .pcapng) with Wire shark software. From there I can change the file ending with pcapng to pcap. That is necessary for the next step. Also in this step, I also used Wire Shark to export specific number of packets in each traffic file. It is helpful because later work may becomes optimal, when in some case the number of packets too large are not necessary. And the time for whole process is also shortened. The picture 5.1 below show about Wire shark interface.


vimeo1.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a
No.
Wireshark · Export Specified Packets
Quick access
Desktop
Libraries
This PC
Network
Save
Cancel
Help
Fram
Ethe
Inte
Tran
0000
0010
0020
0030

Save in: 1_Pcap

Name	Date modified	Type
aim_chat_3a	12/10/2015 3:08 PM	Wireshark ca
aim_chat_3b	12/10/2015 3:08 PM	Wireshark ca
AIMchat1	4/18/2020 11:33 PM	Wireshark ca
AIMchat2	4/18/2020 11:45 PM	Wireshark ca
email1a	12/10/2015 3:08 PM	Wireshark ca
email1b	12/10/2015 3:08 PM	Wireshark ca
email2a	12/10/2015 3:08 PM	Wireshark ca
email2b	12/10/2015 3:08 PM	Wireshark ca
facebook_chat_4a	12/10/2015 3:08 PM	Wireshark ca
facebook_chat_4b	12/10/2015 3:08 PM	Wireshark ca
facebookchat1	4/18/2020 11:46 PM	Wireshark ca

File name:
Save as type: Wireshark/tcpdump/... - pcap (*.dmp.gz;*.dmp;*.cap

□ Compress with gzip

Packet Range

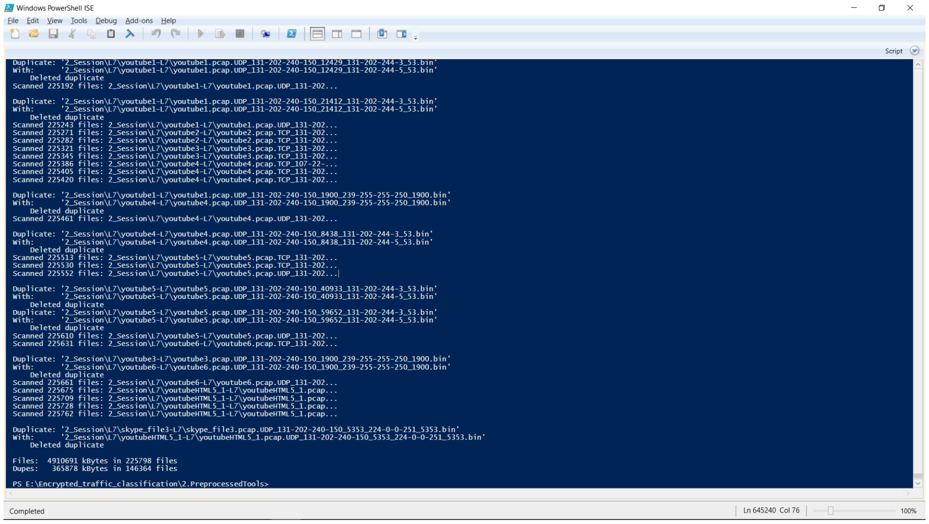
	○ Captured	● Displayed
● All packets	92705	92705
○ Selected packet	1	1
○ Marked packets	0	0
○ First to last marked	0	0
○ Range:	0	0
□ Remove Ignored packets	0	0

```
28 [ACK] Seq=1 Ack=1 Win=445 Len=0 TSval=18643264 TSecr=748420185
unseen segment] 5228 → 60044 [ACK] Seq=1 Ack=2 Win=366 Len=0 TSval=748465241 TSecr=18249026
3 [ACK] Seq=1 Ack=1 Win=485 Len=0 TSval=18643840 TSecr=2564238849
unseen segment] 443 → 60982 [ACK] Seq=1 Ack=2 Win=1000 Len=0 TSval=2564283905 TSecr=18576253
3 [ACK] Seq=1 Ack=1 Win=1205 Len=0 TSval=18643872 TSecr=24558981
unseen segment] 443 → 39638 [ACK] Seq=1 Ack=2 Win=587 Len=0 TSval=24604037 TSecr=18576272
3 [ACK] Seq=1 Ack=1 Win=335 Len=0
unseen segment] 443 → 56307 [ACK] Seq=1 Ack=2 Win=32768 Len=0
3 [ACK] Seq=1 Ack=1 Win=340 Len=0 TSval=18644704 TSecr=3665954270
unseen segment] 443 → 38385 [ACK] Seq=1 Ack=2 Win=694 Len=0 TSval=3665999326 TSecr=18577120
3 [ACK] Seq=1 Ack=1 Win=359 Len=0 TSval=18644736 TSecr=40607403
unseen segment] 443 → 55676 [ACK] Seq=1 Ack=2 Win=601 Len=0 TSval=40652459 TSecr=18577174
3 [ACK] Seq=1 Ack=1 Win=337 Len=0 TSval=18644749 TSecr=249956632

3c:65:50)
```

Figure 5.1: Wireshark Software

2. **Second step** is to split the traffic (.pcap files) . The purpose of this step is to divide the original flow into smaller components depending on the purpose of use. Here I split traffic to two types: Session and Flow (as mentioned above about these two components in chapter 3). Because they are also representing the characteristics of Traffic. Traffic clean is also done in this step. Some packages have no application layer, so the resulting file blank. Some packages create identical files when they have the same content and duplicate data, which can lead to deviations when training CNN. So empty and duplicate files need to be removed. The data format in this step is unchanged.



```
Duplicate: '2_Session\L7\youtube1-L7\youtube1.pcap.UDP_131-202-240-150_12429_131-202-244-3_53.bin'
With:      '2_Session\L7\youtube1-L7\youtube1.pcap.UDP_131-202-240-150_12429_131-202-244-5_53.bin'
    Deleted duplicate
Scanned 225192 files: 2_Session\L7\youtube1-L7\youtube1.pcap.UDP_131-202...

Duplicate: '2_Session\L7\youtube1-L7\youtube1.pcap.UDP_131-202-240-150_21412_131-202-244-3_53.bin'
With:      '2_Session\L7\youtube1-L7\youtube1.pcap.UDP_131-202-240-150_21412_131-202-244-5_53.bin'
    Deleted duplicate
Scanned 225243 files: 2_Session\L7\youtube1-L7\youtube1.pcap.UDP_131-202...
Scanned 225271 files: 2_Session\L7\youtube2-L7\youtube2.pcap.TCP_131-202...
Scanned 225282 files: 2_Session\L7\youtube2-L7\youtube2.pcap.TCP_131-202...
Scanned 225321 files: 2_Session\L7\youtube3-L7\youtube3.pcap.TCP_131-202...
Scanned 225345 files: 2_Session\L7\youtube3-L7\youtube3.pcap.TCP_131-202...
Scanned 225386 files: 2_Session\L7\youtube4-L7\youtube4.pcap.TCP_107-22-...
Scanned 225405 files: 2_Session\L7\youtube4-L7\youtube4.pcap.TCP_131-202...
Scanned 225420 files: 2_Session\L7\youtube4-L7\youtube4.pcap.TCP_131-202...

Duplicate: '2_Session\L7\youtube1-L7\youtube1.pcap.UDP_131-202-240-150_1900_239-255-255-250_1900.bin'
With:      '2_Session\L7\youtube4-L7\youtube4.pcap.UDP_131-202-240-150_1900_239-255-255-250_1900.bin'
    Deleted duplicate
Scanned 225461 files: 2_Session\L7\youtube4-L7\youtube4.pcap.UDP_131-202...

Duplicate: '2_Session\L7\youtube4-L7\youtube4.pcap.UDP_131-202-240-150_8438_131-202-244-3_53.bin'
With:      '2_Session\L7\youtube4-L7\youtube4.pcap.UDP_131-202-240-150_8438_131-202-244-5_53.bin'
    Deleted duplicate
Scanned 225513 files: 2_Session\L7\youtube5-L7\youtube5.pcap.TCP_131-202...
Scanned 225530 files: 2_Session\L7\youtube5-L7\youtube5.pcap.TCP_131-202...
Scanned 225552 files: 2_Session\L7\youtube5-L7\youtube5.pcap.UDP_131-202...

Duplicate: '2_Session\L7\youtube5-L7\youtube5.pcap.UDP_131-202-240-150_40933_131-202-244-3_53.bin'
With:      '2_Session\L7\youtube5-L7\youtube5.pcap.UDP_131-202-240-150_40933_131-202-244-5_53.bin'
    Deleted duplicate
Duplicate: '2_Session\L7\youtube5-L7\youtube5.pcap.UDP_131-202-240-150_59652_131-202-244-3_53.bin'
With:      '2_Session\L7\youtube5-L7\youtube5.pcap.UDP_131-202-240-150_59652_131-202-244-5_53.bin'
    Deleted duplicate
Scanned 225610 files: 2_Session\L7\youtube5-L7\youtube5.pcap.UDP_131-202...
Scanned 225631 files: 2_Session\L7\youtube6-L7\youtube6.pcap.TCP_131-202...

Duplicate: '2_Session\L7\youtube3-L7\youtube3.pcap.UDP_131-202-240-150_1900_239-255-255-250_1900.bin'
With:      '2_Session\L7\youtube6-L7\youtube6.pcap.UDP_131-202-240-150_1900_239-255-255-250_1900.bin'
    Deleted duplicate
Scanned 225661 files: 2_Session\L7\youtube6-L7\youtube6.pcap.UDP_131-202...
Scanned 225675 files: 2_Session\L7\youtubeHTML5_1-L7\youtubeHTML5_1.pcap...
Scanned 225709 files: 2_Session\L7\youtubeHTML5_1-L7\youtubeHTML5_1.pcap...
Scanned 225728 files: 2_Session\L7\youtubeHTML5_1-L7\youtubeHTML5_1.pcap...
Scanned 225762 files: 2_Session\L7\youtubeHTML5_1-L7\youtubeHTML5_1.pcap...

Duplicate: '2_Session\L7\skype_file3-L7\skype_file3.pcap.UDP_131-202-240-150_5353_224-0-0-251_5353.bin'
With:      '2_Session\L7\youtubeHTML5_1-L7\youtubeHTML5_1.pcap.UDP_131-202-240-150_5353_224-0-0-251_5353.bin'
    Deleted duplicate

Files:  4910691 kBytes in 225798 files
Dupes:   365878 kBytes in 146364 files

PS E:\Encrypted_traffic_classification\2.PreprocessedTools>
```

Figure 5.2: Second step

3. **Third step** is also called Image Generation step: Trims all files to uniform length. It means, if the files is larger than 784 bytes, cut it off to make files become 784 bytes. For shorter, add 0 to completion 784 bytes length. Then convert files into gray scale images size 28. As I know, one byte can be transformed into an integer in the range of [0, 255], so a byte can be viewed as a pixel. The main purpose of this step is to create a visual image of the traffic, thereby helping to analyze and make a more intuitive comment on the images I get from the traffic. Example traffic describe with this picture below.

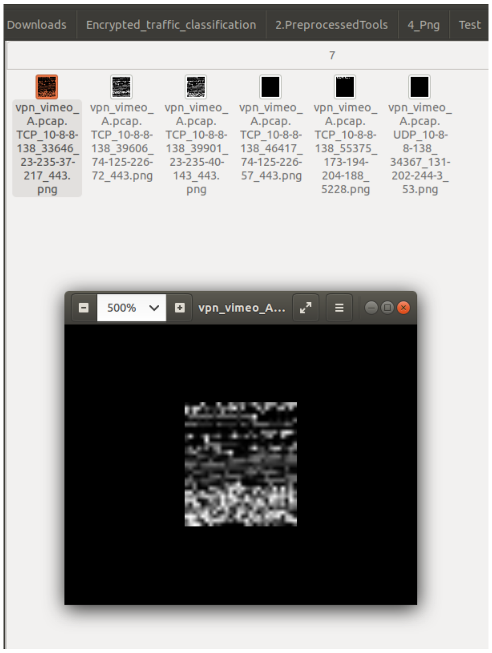

Downloads
Encrypted_traffic_classification
2.PreprocessedTools
4_Png
Test
7
vpn_vimeo_A.pcap.TCP_10-8-8-138_33646_23-235-37-217_443.png
vpn_vimeo_A.pcap.TCP_10-8-8-138_39606_74-125-226-72_443.png
vpn_vimeo_A.pcap.TCP_10-8-8-138_39901_23-235-40-143_443.png
vpn_vimeo_A.pcap.TCP_10-8-8-138_46417_74-125-226-57_443.png
vpn_vimeo_A.pcap.TCP_10-8-8-138_55375_173-194-204-188_5228.png
vpn_vimeo_A.pcap.UDP_10-8-8-138_34367_131-202-244-3_53.png
500%
vpn_vimeo_A...

Figure 5.3: Image Visualization

4. **The Last Step** is called IDX Conversion: This step convert Images generated from third step to files IDX format to feed into CNN. An IDX file contains all pixels and statistics information of a set of images. The IDX file format is binary and simple format for vectors and multi dimensional matrices of various numerical types. Final receive 2 types of files: IDX1 for label IDX3 for traffic data.

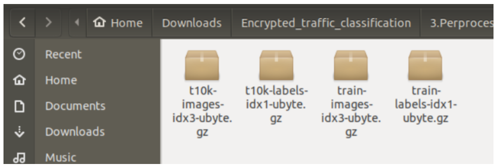

Home
Downloads
Encrypted_traffic_classification
3.Perproces
Recent
Home
Documents
Downloads
Music
t10k-images-idx3-ubyte.gz
t10k-labels-idx1-ubyte.gz
train-images-idx3-ubyte.gz
train-labels-idx1-ubyte.gz

Figure 5.4: IDX Files

5.3 Traffic Representation

Network traffic split granularity include: TCP connection, flow, session, service, and host. I have chosen two components of raw traffic: Session and Flow. As mentioned in Chapter 3, flow is a uni-directional packet stream from one host to another. Flow consist of 5-tuple,they are: Source IP, Destination IP, Destination port, Transport layer protocol. Each flow is associated with a specific services. And so on, to evaluate and classify applications, the use of flow is essential because it carries the characteristics of each services. And Session is quite similar to flow and is defined as bidirectional flow.

About layers choices, I have chosen two types of layers: Layers 7(OSI or Layer 4 TCP/IP) and All Layers. The 7th layer is the application layer, so this layer will best reflect the nature of the traffic. For example, the IMAP protocol represents traffic in the email exchange class, FTP protocol represents traffic in the File transfers class. So for summary, I created 4 cases for experiments,there are: Session+L7,Session + All Layer and Flow+L7,Flow+All Layer.

5.4 Network Architecture

With CNN in general as ll as 2D-CNN in particular, emerged as an effective model for image processing, along with that of LSTM, which processes sequence data. I thought of combining the two network models together, as in 2 articles [49] and [23]. Data coming out of the CNN network can serve as input to the LSTM network to release the flow features contained therein. In addition, I also incorporated a GRU network (variant of LSMT) in experiments to test the effectiveness of this new model. To get an overview of my model, can see the following figure: 5.5.

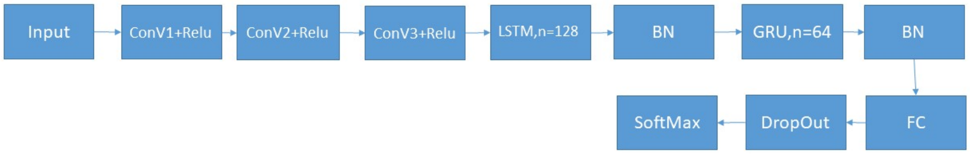


Figure 5.5: Overview Network model

Besides the familiar layers of CNN, I also use the following layers. A

Layer	Input	Filters	Output	Num Units
ConV-1	784*1	25*1	784*32	–
Maxpool-1	784*32	3*1	262*32	–
ConV-2	262*32	25*1	262*64	–
Maxpool-2	262*64	3*1	88*64	–
ConV-3	88*64	25*1	88*128	–
Maxpool-3	88*128	3*1	29*128	–
LSTM	29*128	–	29*128	128
GRU	29*128	–	29*128	64
FC	29*128	–	1024	–
Softmax	1024	–	12	–

Table 5.3: Details Parameters

Drop out layer to reduce overfitting, by dropping out (set to zero) a percentage of outputs from the previous layer. Batch Normalization(BN) makes convergence training faster and can improve performance. It is done by standardization, at the time of training, every feature at the batch level (divide the input ratio into a mean of 0 and the unit variance) and resize it again after reviewing the entire training data set.

My model is combine of 2D-CNN and LSTM architecture. It consist of three Layers Conv2d with ReLu as active funtion. And three Max Pooling Layers after three Conv2d Layers. CNN first reads traffic image of size of 28*28*1(or 784*1) from IDX files. Those image pixels are normalized to [0, 1] from [0,255]. For LSTM, it contains one LSTM Layers with number of hidden unit equal 128 and one LSTM-Gru Layers with numbers of hidden unit equal 64. And two more BN Layers add for normalize the data. Follow it are Fully connected Layer and Drop out Layer. And final is Read Out Layer(Sofmax Layer) for Classify 12 classes of Encrypted Classifycation. For more details about Parameters of each layers, we can see the table 5.3 above.

Chapter 6

Experiments

6.1 Evaluation

The Evaluation Metrics (EV) I use to evaluate the effectiveness of my model are familiar such as: Accuracy, Precision, Recall and F1. In order to talk more about EV I have used as well as an overview of the model evaluation, I will list a few definitions.

6.1.1 Confusion Matrix

The Confusion matrix is a matrix used to specify how each class is classified, which class is best classified, and which class data is often mis-classified into another class. Basically, confusion matrix (table 6.1 below) shows how many data points actually belong to a class, and is predicted to fall into a class.

In particular, the quantities TP, FN, FP, TN have the following meanings:

- TP - True Positive: This quantity tells us the number of correct predictions on the X label.

		Actual Value	
		Positive	Negative
Predicted Value	Positive	TP	FP
	Negative	FN	TN

Table 6.1: Confusion Matrix

- FP - False Positive: This quantity tells us the estimated amount of data is the X label but in fact it is not the X label. In this case our model is wrong.
- TN - True Negative: This quantity tells us the predicted amount of data is not label X and in fact they are not label X. In this case our model is correct because we did not predict that label X.
- FN - False Negative: This quantity tells us the predicted amount of data is not X label but in fact they are true X label. In this case our model is wrong because we did not predict X label.

6.1.2 Performance Metrics

Precision is a metric to determine when false positives are very dangerous. For example, the problem of identifying email spam. With this problem the positive sample will be a spam mail, so a false positive will be a prediction of a non-spam email being put in the spam mailbox. This will greatly affect the users.

$$\frac{TP}{TP + FP} \tag{6.1}$$

Recall is the metric shows that how many True positives are actually well defined. This metric is used to evaluate a model when incorrectly predicting an actual positive sample is dangerous. As for predicting ill patients.

$$\frac{TP}{TP + FN} \tag{6.2}$$

F1 Scoreis a measure that takes both precision and recall into account. A commonly used version of this is the F1-score which is an harmonic mean of precision and recall.

$$2 * \frac{Precision * Recall}{Precision + Recall} \tag{6.3}$$

6.2 Experiments Setting

For experiments,I use VMware Workstation Pro to create Ubuntu (18.04LTS) environment. On My Window Laptop (Processor Core i7-7700HQ, 16GB Ram, GPU GeForceGTX 1050Ti). The class labels are encoded as the one-hot vectors. Lost function is Cross entropy. Training method is mini-batch Stochastic Gradient Descent (SGD) size 50. Adam Optimizer was used with learning rate is 1e-4. Number of round training is 20000. TensorFlow is used as software framework . About data, 1/10 of data were randomly selected as test data, the rest is training data.

6.3 Results and Comparison

As mentioned in the traffic representation section, my experiment will have four different scenarios. There are: Session All Layers,Session Layer 7, Flow All Layers, Flow Layers 7. With the dataset pre-processed according to the above steps, I have created four different file folders base on those scenarios. However, through the experimental process, file folders Flow l7 and Session L7 I created did not give satisfactory results. So I used two files folders after pre-processing from USTC-2016 [42] instead of my files for experiments. The results I obtained for the four scenarios are given in the table 6.2 below (including Accuracy and F1 Score).

After that, I used the results which obtained for 2D-CNN and (2D-CNN + LSTM) models to compare with the 1D-CNN model that the author introduced in the article [42]. Detailed evaluation is given in the figure 6.1 below. As can be seen from the figure, the results of the 2DCNN + LSTM model achieved were better than the other two models for all four scenarios. The difference is not significant, but there has been a certain improvement (about 2-4% for each scenarios). This proves that the combination of CNN

	Accuracy	**F1 Score**
Session All Layers	88.8%	92.2%
Session Layer 7	85.9%	83.1%
Flow All Layers	90.6%	92.5%
Flow Layer 7	84.1%	80.7%

Table 6.2: Results Experiments

and LSTM models is more effective than the 1D-CNN model, a new layer like BN is also a bit effective to avoid overfitting and standardizing data. What makes CNN + LSTM model more effective than normal CNN is that LSTM is responsible for processing sequence data. It helps with the processing of flow time sequence features. This feature contained in packet flow when moving from source to destination. And this features also carry important features that help classify traffic more accurately.

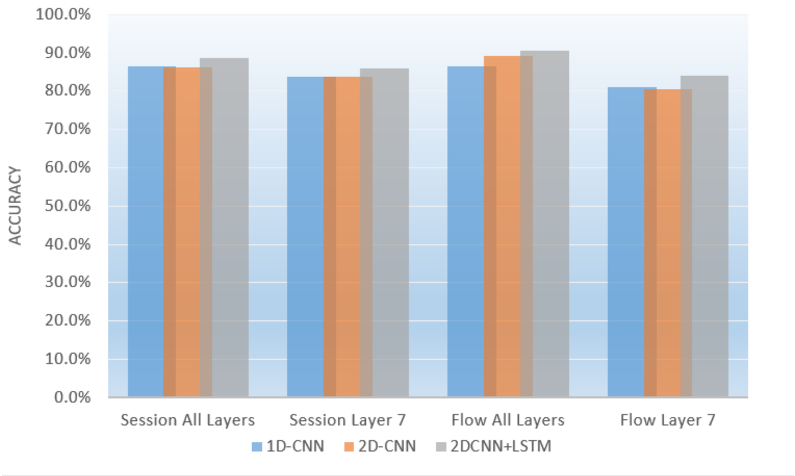
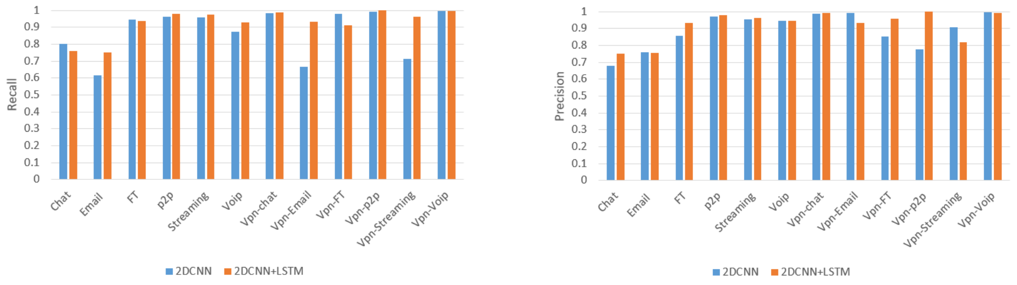


Figure 6.1: Accuracy comparison

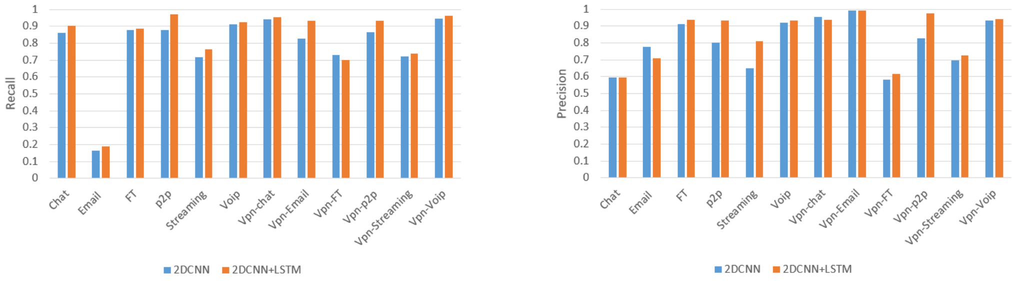
Look at the other side of the data-use. For the All Layers and Flow All Layer Session scenarios, I used my data I created. It can be said that my data generated is smaller than the original data that USTC (for size) has created but looking at the results, I can see the positive here. This proves that the data reduction is feasible while still keeping the model effectiveness, maybe even a little better. However, it must be said that, for two others scenarios Session L7 and FLow L7. The size reduction did not bring a positive results. Thereby I can raise the question that the data needed in Layer 7 is more, to be able to accurately analyze the characteristics of traffic compared to All Layers.

Further more, I also made comparisons for precision and recall for each scenarios (a total 8 figures from 6.2 to 6.5). This comparison helps me better understand the impact of the two models on the analysis for each specific class, helping to evaluate in more detail the classification of classes. Clearly see in the images, the results that I obtained in Model 2D-CNN + LSTM is



(a) (b)

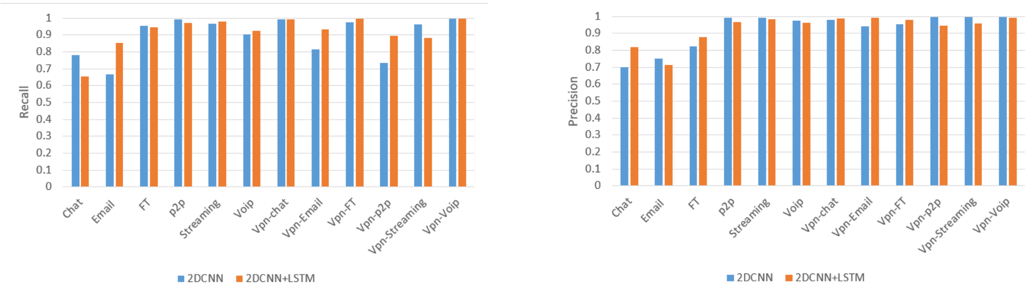
Figure 6.2: Session All Layers



(a) (b)

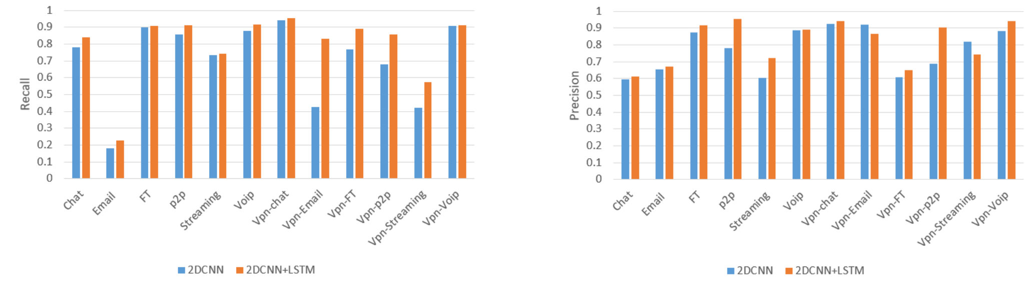
Figure 6.3: Session Layer 7

almost better than model only use CNN. This also shows the more efficiency of the 2D-CNN + LSTM model in classifying each corresponding class in the dataset.



(a) (b)

Figure 6.4: Flow All Layers



(a) (b)

Figure 6.5: Flow Layer 7

Chapter 7

Conclusion

Through the work of my research mentioned in this thesis, we can see the solution for using neural networks for classification of encrypted data. Although this is not a new solution and there has been a lot of research on this issue, but for me, improving the model and bringing higher efficiency than the other article is also an initial success in my work, for my research work later. The steps as well as the processing of data have been mentioned in detail in this thesis. Along with the model I used to get the results. However, there are still some limitations in my work such as using the created data does not bring the desired results (for 2 scenarios Session L7 and Flow L7), which also motivates me to continue my research to find out why the dataset I made a difference.

7.1 Future work

With a data set that is quite well known compared to VPN-nonVPN, it is also the Tor-nonTor dataset [21] which has been published by Canadian Institute of CyberSecurity. About this dataset, it consist of 8 types of traffic (browsing, chat, audio-streaming, video-streaming, mail, VOIP, P2P and File transfer) from more than 18 representative application (e.g., facebook, skype, spotify, gmail etc.). This dataset was created by 2 accounts users Alice and Bob when using services. The difference between tor and vpn is: Tor is a network but it is not a private network setup. It is set up to share services other than the Internet and you do not receive internet from a central place. Every user on the network can choose to volunteer their internet connection with every other user on the network. This also poses a security risk because the people you share with the Internet may see any unencrypted information you send via TOR. VPNs also see this, but you can place your trust in a company instead of random individuals. Because TOR connects to random people, there is no way to successfully block because you don't know which IP address the user connected to. VPN can however be blocked.

I will try to apply my modeling to verify whether the results are satisfactory. And although this data set has some other characteristics that I used in the thesis, the pre-processing is a little different at the same time. That makes me feel a lot more work to be done on this new dataset. Along with that is still improving my model to be able to bring better efficiency, maybe a combination with another neural network model will make more effective. But it all depends on thorough testing and won't take less time, I know. I will try to turn what I do today into a positive start to the upcoming research work.

Bibliography

[1] Prof.Eduard Aved'yan. *Learning Systems*. Springer, 1995.

[2] Colah's Blog. *Understanding LSTM Networks*. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[3] Jason Brownlee. *A Tour of the Weka Machine Learning Workbench*. URL: https://machinelearningmastery.com/tour-weka-machine-learning-workbench/.

[4] Minh Hoang Bui. *Huong dan chi tiet ve mang LSTM*. URL: https://blog.chappiebot.com/huong-dan-chi-tiet-ve-mang-LSTM-va-GRU-trong-NLP/.

[5] Niccolò Cascarano, Luigi Ciminiera, and Fulvio Risso. "Optimizing deep packet inspection for high-speed traffic analysis". In: *Journal of Network and Systems Management* 19.1 (2011), pp. 7–31.

[6] Junyoung Chung et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014).

[7] Susu Cui et al. "A Session-Packets-Based Encrypted Traffic Classification Using Capsule Neural Networks". In: *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems*. IEEE. 2019, pp. 429–436.

[8] Alberto Dainotti, Antonio Pescape, and Kimberly C Claffy. "Issues and future directions in traffic classification". In: *IEEE network* 26.1 (2012), pp. 35–40.

[9] DeepAI. *Machinelearning Glossary*. URL: https://deepai.org/machine-learning-glossary-and-terms/max-pooling.

[10] Adit Deshpande. *A Beginner's Guide To Understanding Convolutional Neural Networks*. URL: `https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/`.

[11] Gerard Draper-Gil et al. "Characterization of encrypted and vpn traffic using time-related". In: *Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)*. 2016, pp. 407–414.

[12] Laurene Fausett. *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice-Hall, Inc., 1994.

[13] Simon Haykin and Neural Network. "A comprehensive foundation". In: *Neural networks* 2.2004 (2004), p. 41.

[14] Bhupendra Ingre, Anamika Yadav, and Atul Kumar Soni. "Decision tree based intrusion detection system for NSL-KDD dataset". In: *International Conference on Information and Communication Technology for Intelligent Systems*. Springer. 2017, pp. 207–218.

[15] Thomas Karagiannis, Konstantina Papagiannaki, and Michalis Faloutsos. "BLINC: multilevel traffic classification in the dark". In: *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*. 2005, pp. 229–240.

[16] Thomas Karagiannis et al. "Is p2p dying or just hiding?[p2p traffic measurement]". In: *IEEE Global Telecommunications Conference, 2004. GLOBECOM'04.* Vol. 3. IEEE. 2004, pp. 1532–1538.

[17] Thomas Karagiannis et al. "Transport layer identification of P2P traffic". In: *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. 2004, pp. 121–134.

[18] Manjiri V Kotpalliwar and Rakhi Wajgi. "Classification of Attacks Using Support Vector Machine (SVM) on KDDCUP'99 IDS Database". In: *2015 Fifth International Conference on Communication Systems and Network Technologies*. IEEE. 2015, pp. 987–990.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[20] James F Kurose. *Computer networking: A top-down approach featuring the internet, 3/E*. Pearson Education India, 2005.

[21] Arash Habibi Lashkari et al. "Characterization of Tor Traffic using Time based Features." In: *ICISSP*. 2017, pp. 253–262.

[22] Yann LeCun et al. "LeNet-5, convolutional neural networks". In: *URL: http://yann. lecun. com/exdb/lenet* 20 (2015), p. 5.

[23] Manuel Lopez-Martin et al. "Network traffic classifier with convolutional and recurrent neural networks for Internet of Things". In: *IEEE Access* 5 (2017), pp. 18042–18050.

[24] Mohammad Lotfollahi et al. "Deep packet: A novel approach for encrypted traffic classification using deep learning". In: *Soft Computing* 24.3 (2020), pp. 1999–2012.

[25] Weizhi Meng, Wenjuan Li, and Lam-For Kwok. "Design of intelligent KNN-based alarm filter using knowledge-based alert verification in intrusion detection". In: *Security and Communication Networks* 8.18 (2015), pp. 3883–3895.

[26] Shane Miller, Kevin Curran, and Tom Lunney. "Multilayer perceptron neural network for detection of encrypted VPN network traffic". In: *2018 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA)*. IEEE. 2018, pp. 1–8.

[27] Andrew W Moore and Konstantina Papagiannaki. "Toward the accurate identification of network applications". In: *International Workshop on Passive and Active Network Measurement*. Springer. 2005, pp. 41–54.

[28] Andrew W Moore and Denis Zuev. "Internet traffic classification using bayesian analysis techniques". In: *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. 2005, pp. 50–60.

[29] Wendell Odom and Sean Wilkins. *CCNA Routing and Switching 200-125 Official Cert Guide and Network Simulator Library*. Cisco Press, 2017.

[30] Professor The Que Pham. *Mang May Tinh*. NXB Thong Tin va Truyen Thong, 2009.

[31] Jing Ran, Yexin Chen, and Shulan Li. "THREE-DIMENSIONAL CONVOLUTIONAL NEURAL NETWORK BASED TRAFFIC CLASSIFICATION FOR WIRELESS COMMUNICATIONS". In: *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE. 2018, pp. 624–627.

[32] Neha G Relan and Dharmaraj R Patil. "Implementation of network intrusion detection system using variant of decision tree algorithm". In: *2015 International Conference on Nascent Technologies in the Engineering Field (ICNTE)*. IEEE. 2015, pp. 1–5.

[33] Sumit saha. *A Comprehensive Guide to Convolutional Neural Networks*. URL: `https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53`.

[34] Andreas Rømer Hjorth Salik Lennert Pedersen. "Classification of encrypted traffic using deep learning". PhD thesis. Technical University of Denmark, 2018.

[35] Harshit Saxena and Vineet Richariya. "Intrusion detection in KDD99 dataset using SVM-PSO and feature reduction with information gain". In: *International Journal of Computer Applications* 98.6 (2014).

[36] Runyuan Sun et al. "Traffic classification using probabilistic neural networks". In: *2010 Sixth International Conference on Natural Computation*. Vol. 4. IEEE. 2010, pp. 1914–1919.

[37] Brendan Tierney. *Understanding, Building and Using Neural Network Machine Leaning Models using Oracle 18c*. URL: `https://developer.oracle.com/databases/neural-network-machine-learning.html`.

[38] Hu Ting, Wang Yong, and Tao Xiaoling. "Network traffic classification based on kernel self-organizing maps". In: *2010 International Conference on Intelligent Computing and Integrated Systems*. IEEE. 2010, pp. 310–314.

[39] Ly Vu, Cong Thanh Bui, and Quang Uy Nguyen. "A deep learning based method for handling imbalanced problem in network traffic classification". In: *Proceedings of the Eighth International Symposium on Information and Communication Technology*. 2017, pp. 333–339.

[40] Ly Vu et al. “Time Series Analysis for Encrypted Traffic Classification: A Deep Learning Approach”. In: *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*. IEEE. 2018, pp. 121–126.

[41] Pan Wang et al. “Datanet: Deep learning based encrypted network traffic classification in sdn home gateway”. In: *IEEE Access* 6 (2018), pp. 55380–55391.

[42] Wei Wang et al. “End-to-end encrypted traffic classification with one-dimensional convolution neural networks”. In: *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE. 2017, pp. 43–48.

[43] Wikipedia. *Deep packet inspection*. URL: `https://en.wikipedia.org/wiki/Deep_packet_inspection`.

[44] Wikipedia. *Port (computer networking)*. URL: `https://en.wikipedia.org/wiki/Port_(computer_networking)`.

[45] Kuai Xu, Zhi-Li Zhang, and Supratik Bhattacharyya. “Profiling internet backbone traffic: behavior models and applications”. In: *ACM SIGCOMM Computer Communication Review* 35.4 (2005), pp. 169–180.

[46] Sebastian Zander, Thuy Nguyen, and Grenville Armitage. “Automated traffic classification and application identification using machine learning”. In: *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN’05) l*. IEEE. 2005, pp. 250–257.

[47] Yi Zeng et al. “$Deep-Full-Range$: A Deep Learning Based Network Encrypted Traffic Classification and Intrusion Detection Framework”. In: *IEEE Access* 7 (2019), pp. 45182–45190.

[48] Huiyi Zhou et al. “A method of improved CNN traffic classification”. In: *2017 13th International Conference on Computational Intelligence and Security (CIS)*. IEEE. 2017, pp. 177–181.

[49] Zhuang Zou et al. “Encrypted traffic classification with a convolutional long short-term memory neural network”. In: *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems*. IEEE. 2018, pp. 329–334.

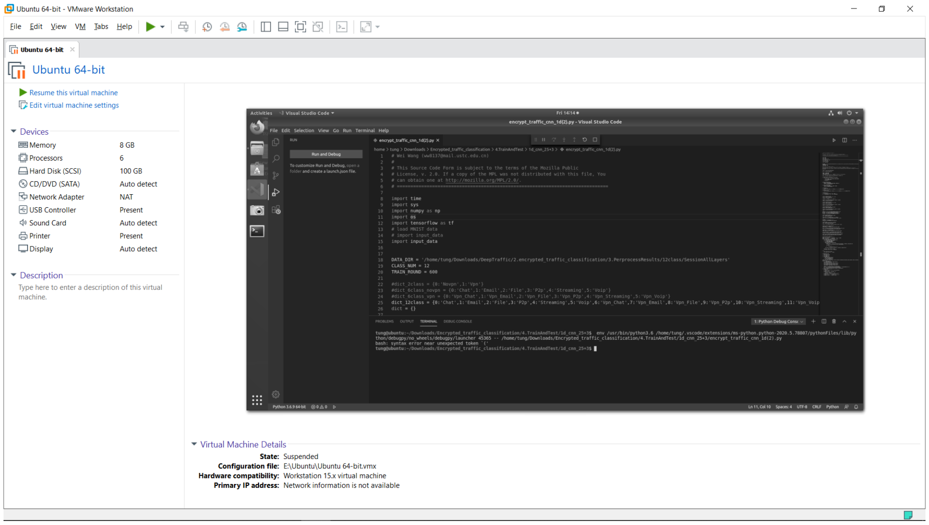
Appendix A

VMware Workstation

These Operating Systems (OS) will act as programs on the computer. Virtual machines are ideal for testing OS, such as newly released Windows 10 or Linux. You can also use virtual machines to run software on OS that are not compatible, for example, you can run programs for Windows on Macs using virtual machines. On the other hand, users may not have to pay anything because there are a few great free virtual machine programs to experience. Virtual machine is a program that acts like a virtual computer. It runs on the current operating system - the host operating system and provides virtual hardware to the guest operating system. The guest operating systems run on the windows of the host operating system, just like any other computer program. For guest operating systems, the virtual machine presents itself as a physical physical machine. Virtual machines provide virtual hardware, including virtual CPUs, virtual RAM, hard drives, network interfaces, and other devices. Virtual hardware devices are provided by the virtual machine and are mapped to real hardware on the real machine.

For example, the virtual hard drive is stored in a file located on the actual hard drive. You can install multiple virtual machines on real machines and are only limited by the amount of storage space available to them. Once you have installed some operating systems, you can open the virtual machine program and select the virtual machine you want to boot, start the guest operating system and run in a window of the host operating system or you can also run in virtual machine mode or full-screen mode .

For personal laptop, having two operating systems on it brings some inconvenience. So I decided to install VMware as a tool to simulate the Linux environment (Ubuntu) for my work. It gives me good customization when I can use the two operating systems in parallel without having to worry about problems. For more details, I can handle the step 1 and 2 in pre-processing phase with PowerShell on Windows. Another step like step 3 and 4 in pre-processing and main parts I can do it on Ubuntu easily.


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Ubuntu 64-bit
Ubuntu 64-bit
Resume this virtual machine
Edit virtual machine settings
Devices
Description
Type here to enter a description of this virtual machine.
Activities
Visual Studio Code
Fri 14:14
encrypt_traffic_cnn_1d(2).py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
RUN
Run and Debug
To customize Run and Debug, open a folder and create a launch.json file.
encrypt_traffic_cnn_1d(2).py
home > tung > Downloads > Encrypted_traffic_classification > 4.TrainAndTest > 1d_cnn_25+3 > encrypt_traffic_cnn_1d(2).py
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: Python Debug Consc
Python 3.6.9 64-bit ⊗ 0 ⚠ 0
Ln 11, Col 10 Spaces: 4 UTF-8 CRLF Python
Virtual Machine Details

Memory	8 GB
Processors	6
Hard Disk (SCSI)	100 GB
CD/DVD (SATA)	Auto detect
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

```python
# Wei Wang (ww8137@mail.ustc.edu.cn)
#
# This Source Code Form is subject to the terms of the Mozilla Public
# License, v. 2.0. If a copy of the MPL was not distributed with this file, You
# can obtain one at http://mozilla.org/MPL/2.0/.
# ==============================================================================

import time
import sys
import numpy as np
import os
import tensorflow as tf
# load MNIST data
# import input_data
import input_data


DATA_DIR = '/home/tung/Downloads/DeepTraffic/2.encrypted_traffic_classification/3.PerprocessResults/12class/SessionAllLayers'
CLASS_NUM = 12
TRAIN_ROUND = 600

#dict_2class = {0:'Novpn',1:'Vpn'}
#dict_6class_novpn = {0:'Chat',1:'Email',2:'File',3:'P2p',4:'Streaming',5:'Voip'}
#dict_6class_vpn = {0:'Vpn_Chat',1:'Vpn_Email',2:'Vpn_File',3:'Vpn_P2p',4:'Vpn_Streaming',5:'Vpn_Voip'}
dict_12class = {0:'Chat',1:'Email',2:'File',3:'P2p',4:'Streaming',5:'Voip',6:'Vpn_Chat',7:'Vpn_Email',8:'Vpn_File',9:'Vpn_P2p',10:'Vpn_Streaming',11:'Vpn_Voip
dict = {}
```

```
tung@ubuntu:~/Downloads/Encrypted_traffic_classification/4.TrainAndTest/1d_cnn_25+3$  env /usr/bin/python3.6 /home/tung/.vscode/extensions/ms-python.python-2020.5.78807/pythonFiles/lib/py
thon/debugpy/no_wheels/debugpy/launcher 45365 -- /home/tung/Downloads/Encrypted_traffic_classification/4.TrainAndTest/1d_cnn_25+3/encrypt_traffic_cnn_1d(2).py
bash: syntax error near unexpected token `('
tung@ubuntu:~/Downloads/Encrypted_traffic_classification/4.TrainAndTest/1d_cnn_25+3$
```

State:	Suspended
Configuration file:	E:\Ubuntu\Ubuntu 64-bit.vmx
Hardware compatibility:	Workstation 15.x virtual machine
Primary IP address:	Network information is not available

Figure A.1: VMware Workstation Pro 15

Appendix B

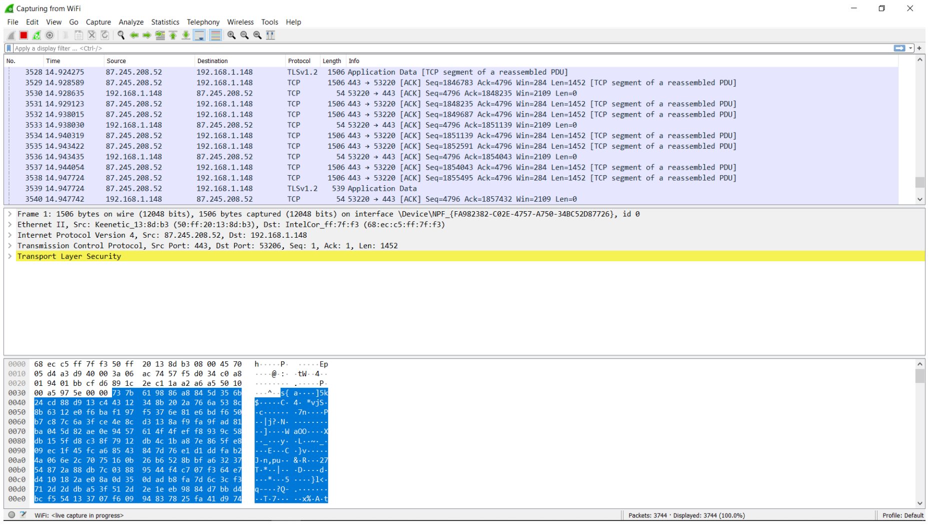
Wire Shark

Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and try to display that packet data as detailed as possible. A network packet analyzer is used as a measuring device to check what is happening inside the network cable, like the function of a voltmeter used by an electrician to check what is happening inside the power cable. In the past, these tools were often very expensive, or proprietary, or both. However, with the appear of Wireshark, everything has changed. Wireshark is probably one of the best open source packet analyzers available today (open source packet analyzer).Purpose of Wireshark:

- Network administrators use Wireshark to troubleshoot network problems.
- Network security engineers use Wireshark to check for security issues.
- QA engineers use Wireshark to verify network applications.
- Developers use Wireshark to debug protocol implementations.
- People use Wireshark to learn network protocol internals.

In addition it also has features such as: Supports both Unix and Windows, open files containing packet data captured by tcpdump / WinDump, Wireshark and some other packet capture programs, save captured packet data, export some or all packages in some capture file format.

We can see that Wireshark can do a lot of things and can aid in a lot of research work. Specifically in my work, it helped me analyze pcap files, convert pcapng to pcap extensions. Along with that is the limit on the number of packets, serving different requests.



```
No.   Time      Source          Destination     Protocol Length Info
 3528 14.924275 87.245.208.52   192.168.1.148   TLSv1.2  1506 Application Data [TCP segment of a reassembled PDU]
 3529 14.928589 87.245.208.52   192.168.1.148   TCP      1506 443 → 53220 [ACK] Seq=1846783 Ack=4796 Win=284 Len=1452 [TCP segment of a reassembled PDU]
 3530 14.928635 192.168.1.148   87.245.208.52   TCP        54 53220 → 443 [ACK] Seq=4796 Ack=1848235 Win=2109 Len=0
 3531 14.929123 87.245.208.52   192.168.1.148   TCP      1506 443 → 53220 [ACK] Seq=1848235 Ack=4796 Win=284 Len=1452 [TCP segment of a reassembled PDU]
 3532 14.938015 87.245.208.52   192.168.1.148   TCP      1506 443 → 53220 [ACK] Seq=1849687 Ack=4796 Win=284 Len=1452 [TCP segment of a reassembled PDU]
 3533 14.938030 192.168.1.148   87.245.208.52   TCP        54 53220 → 443 [ACK] Seq=4796 Ack=1851139 Win=2109 Len=0
 3534 14.940319 87.245.208.52   192.168.1.148   TCP      1506 443 → 53220 [ACK] Seq=1851139 Ack=4796 Win=284 Len=1452 [TCP segment of a reassembled PDU]
 3535 14.943422 87.245.208.52   192.168.1.148   TCP      1506 443 → 53220 [ACK] Seq=1852591 Ack=4796 Win=284 Len=1452 [TCP segment of a reassembled PDU]
 3536 14.943435 192.168.1.148   87.245.208.52   TCP        54 53220 → 443 [ACK] Seq=4796 Ack=1854043 Win=2109 Len=0
 3537 14.944054 87.245.208.52   192.168.1.148   TCP      1506 443 → 53220 [ACK] Seq=1854043 Ack=4796 Win=284 Len=1452 [TCP segment of a reassembled PDU]
 3538 14.947724 87.245.208.52   192.168.1.148   TCP      1506 443 → 53220 [ACK] Seq=1855495 Ack=4796 Win=284 Len=1452 [TCP segment of a reassembled PDU]
 3539 14.947724 87.245.208.52   192.168.1.148   TLSv1.2   539 Application Data
 3540 14.947742 192.168.1.148   87.245.208.52   TCP        54 53220 → 443 [ACK] Seq=4796 Ack=1857432 Win=2109 Len=0
```

```
> Frame 1: 1506 bytes on wire (12048 bits), 1506 bytes captured (12048 bits) on interface \Device\NPF_{FA982382-C02E-4757-A750-34BC52D87726}, id 0
> Ethernet II, Src: Keenetic_13:8d:b3 (50:ff:20:13:8d:b3), Dst: IntelCor_ff:7f:f3 (68:ec:c5:ff:7f:f3)
> Internet Protocol Version 4, Src: 87.245.208.52, Dst: 192.168.1.148
> Transmission Control Protocol, Src Port: 443, Dst Port: 53206, Seq: 1, Ack: 1, Len: 1452
> Transport Layer Security
```

```
0000  68 ec c5 ff 7f f3 50 ff  20 13 8d b3 08 00 45 70   h·····P· ·····Ep
0010  05 d4 a3 d9 40 00 3a 06  ac 74 57 f5 d0 34 c0 a8   ····@·:· ·tW··4··
0020  01 94 01 bb cf d6 89 1c  2e c1 1a a2 a6 a5 50 10   ········ .·····P·
0030  00 a5 97 5e 00 00 73 7b  61 98 86 a8 84 5d 35 6b   ···^··s{ a····]5k
0040  24 cd 88 d9 13 c4 43 12  34 8b 20 2a 76 6a 53 8c   $·····C· 4· *vjS·
0050  8b 63 12 e0 f6 ba f1 97  f5 37 6e 81 e6 bd f6 50   ·c······ ·7n····P
0060  b7 c8 7c 6a 3f ce 4e 8c  d3 13 8a f9 fa 9f ad 81   ··|j?·N· ········
0070  ba 04 5d 82 ae 0e 94 57  61 4f 4f ef f8 93 9c 58   ··]····W aOO····X
0080  db 15 5f d8 c3 8f 79 12  db 4c 1b a8 7e 86 5f e8   ··_···y· ·L··~·_·
0090  09 ec 1f 45 fc a6 85 43  84 7d 76 e1 d1 dd fa b2   ···E···C ·}v·····
00a0  4a 06 6e 2c 70 75 16 0b  26 b6 52 8b bf a6 32 37   J·n,pu·· &·R···27
00b0  54 87 2a 88 db 7c 03 88  95 44 f4 c7 07 f3 64 e7   T·*··|·· ·D····d·
00c0  d4 10 18 2a e0 8a 0d 35  0d ad b8 fa 7d 6c 3c f3   ···*···5 ····}l<·
00d0  71 2d 2d db a5 3f 51 2d  2e 1e eb 98 84 d7 bb d4   q--··?Q- .·······
00e0  bc f5 54 13 37 07 f6 09  94 83 78 25 fa 41 d9 74   ··T·7··· ··x%·A·t
```

WiFi: <live capture in progress>   Packets: 3744 · Displayed: 3744 (100.0%)   Profile: Default

Figure B.1: Wireshark