# Malware Detection
# Based on Machine Learning Algorithms


**Student Name: Jing Liu**

**Student ID：18231917**

**Supervisor: Dr. Michael Schukat**

**Course: Data Analytics**

**Academic Year: 2018/2019**

# Abstract

This paper illustrates that machine learning algorithms are effective in detecting malicious software by analysing network traffic. The data set used in this study consists of malware and benign network (TCP/IP) packets expressed by raw bytes. The research presents different approaches of feature engineering, machine learning and deep learning algorithms for detecting malware.

# Table of Contents

# 1.Introduction

*The purpose of this paper is to investigate whether machine learning methods and neural networks can detect the presence of viruses in software by analysing network traffic. Our experimentwill use different feature analysis methods and machine learning models to verify this hypothesis.*

Nowadays, the development of malware is a big threat to modern information technology.There are lots of malware like backdoors, viruses, and worms that can cause very serious damages to the security and privacy of computer systems. Malware is a powerful and illegal instrument, so this is the reason why there is a growing market for anti-malware products.

Malware detection analysis generally consists of either static analysis or dynamic analysis[1]. Static analysis (also known as code analysis) is the actual viewing of code or information about the program or its expected behaviours to get a better understanding of the malware. Dynamic analysis (also known as behavioural analysis) is a kind of analysis that relies on information that is collected from the operating system at runtime, or how malware behaves when executed, i.e. network access, system calls and so on. The research of this paper focuses on dynamic analysis by focusing on analysing network traffic.

There are lots of antimalware products and most of the malware detection systems are signature-based, they recognize the malware by matching a signature. Signature-based detection may involve searching a series of bytes or packet queues in a network flow to find known malicious programs. Signature-based approach also can be static analysis or dynamic analysis, for example, a static-based approach only uses structural information (such as a sequence of bytes) to determine maliciousness.Jha et al [36] proposed an executable static analyser, and Sulaiman et al. proposed to identify malware by comparing PUI's (program (static)/process (dynamic) under inspection) assembly code with known malicious signatures, where the PUI (program (static)/process (dynamic) under inspection) was disassembled using the ASM code PE explorer. Dynamic methods take advantage of information about the runtime of the program (e.g. monitoring the system stack behaviour in the runtime stack), for example, IIgun et al. proposed a rule-based IDS detection method, which models the attack process as a transition diagram, and then compares the program behaviour with the state transition diagram during the detection process. Alice et al. proposed a signature-based worm detection method based on known malicious behaviour by identifying the signature by monitoring the data stream. The biggest benefit of signature-based is that this kind of method is easier to develop and understand if you know the network behaviour that you want to find, i.e. you can utilize a particular signature to look for a specific string in an exploitable vulnerability to detect whether or not it's a malware. However, since the signature only detects known attacks, a signature must be made for each type of attack, and new attacks cannot be detected. Comparing with the signature-based method, AI algorithms can detect threats for which there exists no signature yet. AI algorithms usually also can be combined with static analysis or dynamic analysis. AI based static analysis usually analyse the file itself, binary program or PE header. For example,Chatchai et al [27] proposed use n-gram based sequential features extracted from content of the files. Edward et al [6] proposed a static analysis approach that just look at the information from the binary program file, and build a

neural network to determine maliciousness. While AI based on dynamic analysis is more popular, that is, using AI algorithms to analyse behaviour of collected malware through monitoring in a specially instrumented environment, it automatically process the reliable and accurate information that obtain on execution of malicious program. For example, Dmitri et al [5] proposed method to extracts 972 behaviour features across different protocols and network layers referring to transaction, session and flow and so on and using various supervised algorithms to determine the network is malicious.

The goal of this research is to detect malware by analysing some specific network layers protocols (IP and TCP).The n-gram approach will extract a set of features for the effective classification of malware families. Several classification algorithms (Decision Tree, LSTM and Naïve Bayes and SVM) will be trained and a combinationof these algorithms (by using a majority voter) will be used to improve the performances of these algorithms. Finally, these proposed models will be evaluated by some evaluation methods and investigate their respective malware detection rate.

The layout of this paper consists of following parts:

- Chapter 2contains the literature review, which introduces some techniques I have researched and used in my project.
- Chapter 3 is about my system architecture, which concisely described what my system looks like.
- Chapter 4 explains my dataset and feature selection.
- Chapter 5 is about feature analysis, using some machine learning feature extraction methods to optimize these original features in dataset collection to get better performance.
- Chapter 6 describes what classifiers I choose and how do they work.
- Chapter 7 entails a model enhancement, which shows how to improve these machine learning or neural network architectures and get a better detection performance.
- Chapter 8 is about the classification evaluation methodology that I used to evaluate these classifiers.
- Chapter 9 shows the result and performance of these classifiers.
- Chapter 10 shows the discussion and limitations of my system.
- Chapter 11 outlines conclusion and future work.

# 2.Literature Review

## 2.1 Malware Introduction

Malware generally means malicious software. It is intended to harm computers and computer users by stealing some important information, files or destroying internal procedures on purpose [28]. Rabia [28] mentioned that with the widespread use of malware security incidentshavemassively increased. Hacker (malware maker) always try write some programs that are hardly detectable, they successfully embed these programs into the software or application, from a simple program to encrypted code, and then go to some complex viruses, for example, polymorphic and metamorphic viruses [28].

Based on the malware generation process and characteristics, the following categories are common malware types:

- Virus: A virus is a malicious program that copies itself to other applications, files, or even boot sectors. It can do anything, for example, steal information, keystroke logs and infect the computer system, the characteristics of a virus are self-replication and insertion of malicious code into other programs without the user's consent.
- Trojans: Trojans is a standalone malicious program, a Trojan looks like a regular application, media or any other files, but it contains a malicious payload. Most Trojans contain backdoors that an attacker can use to steal information, spread other malware, or use the resource of an infected machine in a botnet, some popular Trojans include: Nerbus, SubSeven, and back orifice.
- Worm: A worm is a type of malware that replicates itself to spread and infect other systems. Computer worms use the network, links, P2P networks, email and exploit vulnerabilities to spread themselves. The difference with a virus is that the virus inserts the code into other programs, but worms just copy itself. Worms do not necessarily contain payloads, but most worms contain them and some worms are designed to spread only with without payload.
- Botnet: Botnet refers to the use of one or more means of communication to infect a large number of hosts with bots to create a one-to-many network between the controller and the infected host. Bot program is the abbreviation of a robot, which refers to the program code that implements malicious control function.
- Rootkit:Rootkit is a malware program which creates a backdoor into the system for providing full access to the hacker to stealing malware information. Rootkits can be difficult to detect and remove based on where the rootkit resides. For example, firmware-level rootkits may require hardware replacement, and kernel-level rootkits may require reinstallation of the operating system.

## 2.2 Networking and IP/TCP

Networking is a way to connect computers to communicate, for example, users always like sending email, sharing files and exchanging images on Internet. In fact, all of this requires computers to access multiple networks and share their resources. These networks generally

include the local area network (LAN), the campus area network (CAN), the metropolitan area network (MAN), and the area network (WAN) [31].

Networks are defined in terms of the protocol, the networking protocol is a set of rules established for users to exchange information. On this paper we focus on transmission control protocol (TCP) / internet protocol (IP).TCP/IP is the protocol suite used for communications between hosts in most local networks and on the Internet [31]. TCP/IP consists of four layers, they are application layer, transport layer, Internet layer, and network interface. Each layer performs different functions and is implemented by several protocols, and the upper layer protocol uses the services provided by the underlying protocol. This paper will introduce the purpose and use of each layer.

The application layer is responsible for the display and acquisition of data. The data link layer, network layer, and transport layer are responsible for handling network communication details. This part must be stable and efficient, so they are all implemented in kernel space. The application layer is implemented in user space because it handles a lot of logic, such as file transfers, name queries, and network management.

The transport layer provides end-to-end communication for applications on both hosts. Unlike the network layer, the transport layer only cares about communication instead of the transit process of the data packet. The transport layer encapsulates an end-to-end logical communication link for the application, which is responsible for data transmission and reception, link timeout retransmission and so on. There are three main transport layer protocols: TCP protocol, UDP protocol an SCTP protocol. The TCP layer header detail is shown as Fig.2.1 and it will be explained in network traffic introduction.

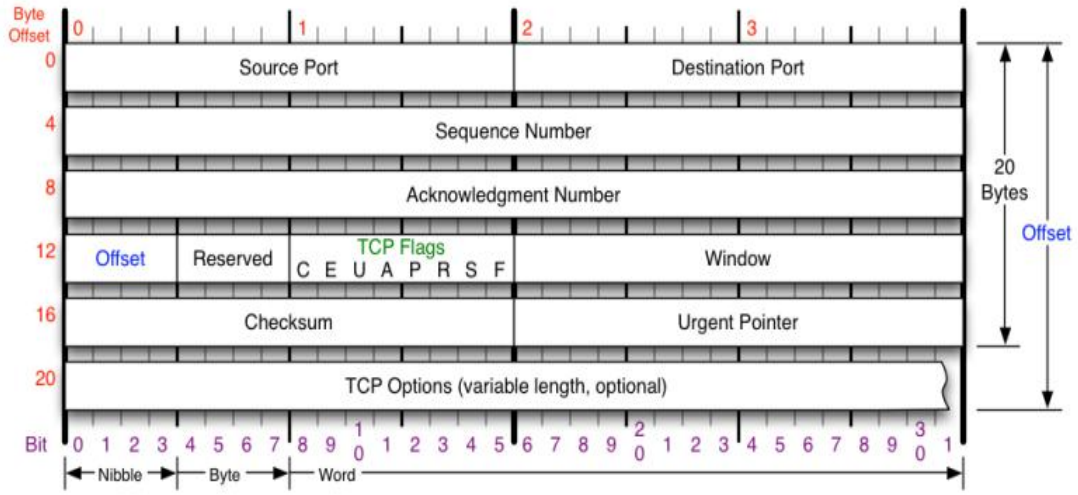


**Fig.2.1** TCP Header [16]

The internet layer implements routing and forwarding of data packets. WAN (Wide Area Network) usually uses a number of hierarchical routers to connect to a decentralized host or LAN (Local Area Network), therefore, the two hosts of the communication are generally not directly connected, but are connected through multiple intermediate routers, the tasks of the network layer is to select these intermediate nodes and the communication path between the two hosts has been determined. At the same time, the network layer hides the details of the

network topology connection from the upper-layer protocol, so that in the view of the transport layer and the network application, the two sides of the communication directly connected. The IP layer header detail shows in Fig.2.2 and it will be explained in chapter 2.3.

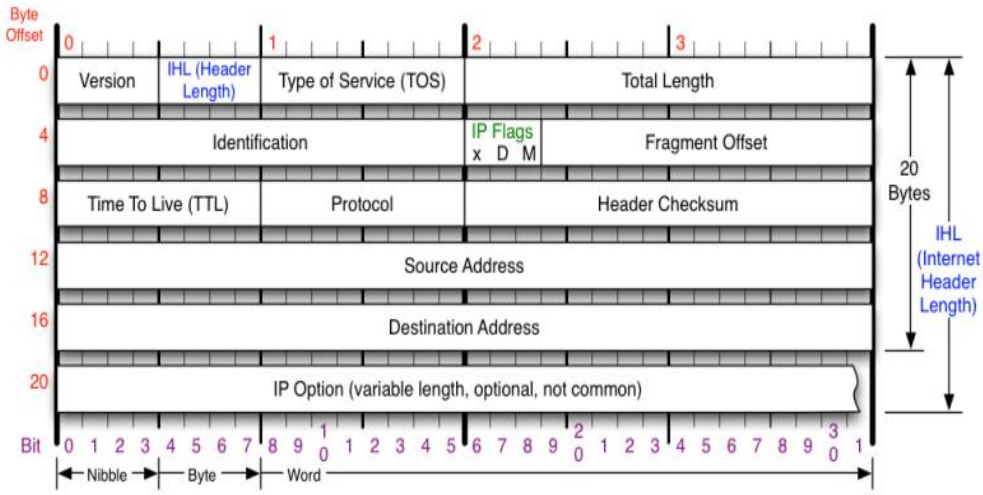


**Fig.2.2**IPv4 Header [16]

The network interface layer defines how the host connects to the network and how data packets are to be formatted for transmission and touring.It corresponds to the data link and physical layers of the OSI reference model, it has two commonly protocols: ARP and RARP, they achieved the mutual conversion between the IP address and the machine physical address.

## 2.3 Network Traffic Introduction

On the perspective of network analysis, here majorly based on the paper “malware introduction”[1], this paper published by SANA Institute Reading room, and it illustrates the goal of malware analysis is to gain an understanding of how a specific piece of malware functions so that defences can be built to protect an organization's network, this paper also gives some introduction of how to capture network traffic how to analysing network traffic, for example, malware can be captured in both dynamic and static way, static malware analysis is like viewing of code or content of a malware file while dynamic is explaining that how the malware behaved when executed,(the details of static and dynamic analysis has explained in chapter 1 introduction), we can capture lots of data when it executed and then we can analyse the virus through analysing these "dynamic data", the paper highlighted the way of analysing behavioural analysis is through researching any changes or unusual behaviour on an infected system, for example, files may be added or modified, new services have been installed, new processes that are running and so on, besides the behaviour of the system itself, network traffic also is a big part to be considered.

The common network traffic type is shown as follows[34]:

- HTTP traffic: The core protocol of the World Wide Web,HTTP is the most widely used protocol on the Internet. With the sharing of video sharing sites such as

YouTube, Weibo, the HTTP protocol's network traffic has been the first in the past four years.

- FTP traffic: A protocol that enables a client to send and receive complete files from a server, since the advent of the Internet, FTP has been one of the most frequently used application services for download files, the importance has been reduced but it is still one of the important applications and way.
- SMTP traffic: One of several key protocols that are used to provide e-mail services, Email is an important part of the overall Internet business. According to statistics, the main purpose of ¾ or more users online is to send and receive mail, and hundreds of millions of emails are delivered globally every day.
- VoIP traffic: In 2006, global IP telephone users increase of 83%, and VoIP calls reached 75% of all calls, therefore, the VoIP traffic on the Internet is also very important and worthy of administrator attention.
- P2P traffic: P2P protocol is for file-sharing or transfer. The current major network bandwidth consumer is P2P file sharing, globally, 95% of the network bandwidth during the night hours is occupied by P2P.

Network traffic classification can be performed with two different ways [5]:

- Packet level methods mean to check each packet's characteristics and application signatures
- Flow level methods mean the statistical analysis and extraction of characteristics from the flow

Network traffic classification is decided by different major attributes [5]:

- Port-based attributes in 7 layers inISO reference model, for some specific layers especially for transport layer (target TCP port number, flag, source port, destination port, and header length), it contains some traffic characteristicinformation and these information may identify malware type.
- Payload based attributes at the application layer level, then analysing malware based on n-gram method.
- Statistically based attributes relate to traffic statistical, for example, packet size, flow duration, packet's arrival time and packet length, etc.

This research selects the way of port-based to analyse maliciousness, especially for network layer and transport layer. Fig.2.3shows detailed information of internet protocol,IPis the core protocol in the TCP/IP, it provides unreliable, connectionless services, whichrely on protocols at other layers for error control. In a LAN environment, IP isgenerally in Ethernet frames, all TCP, UDP, ICMP and IGMP data are encapsulated in IP datagram for transmission. There are some important features can be used for malware detection in the network layer:

- Flags: it takes 3 bits, it marks whether a datagram requires segmentation
- Segment offset: it takes 13 bits, if a datagram requires segmentation, this filed indicates where the segment offset begins with the original datagram.

- Protocol: it takes 8 bits, it indicates the upper-layer protocol type is encapsulated by the IP layer, such as ICMP(1), IGMP(2), TCP(6), UDP(17) and so on.
- Source IP and destination IP: each occupying 32 bits respectively, it is used to indicate the source host address for sending IP data packets and the destination host address for receiving IP packets.

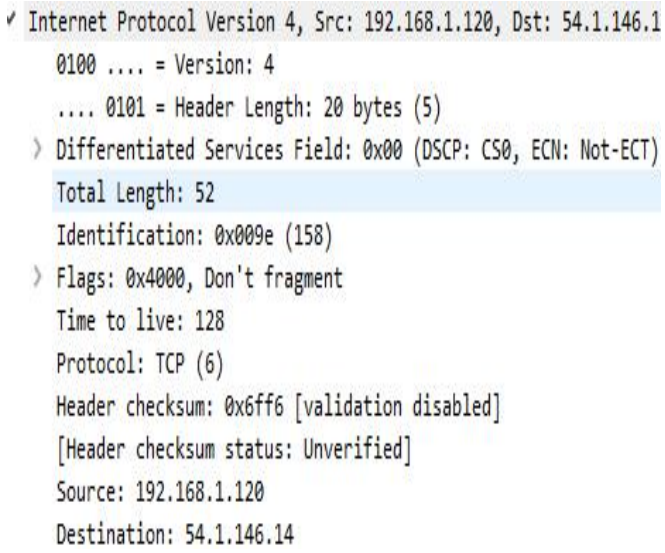


**Fig.2.3** Internet Protocol

Fig.2.4 shows more detailed attributes of transport control protocol, this layer is responsible for getting all information, so it must keep track of data unit fragmentation, out-of-order packets, and other hazards that can occur during transmission. This layer provides a transparent, reliable data transfer service for the upper layer and transfer the most useful information for layer's communication, there are many important attributes inthe transport layerthat can be considered for analysing maliciousness:

- Source Port and Destination Port: It takes 16 bits, the TCP uses "ports" to identify application processes at the source and destination. The port number can use any number between 0 and 65535, the operating system can dynamically assign a port number to the client's application when receiving a service request. On the server-side, each service serves the user at the "Well-Known Port".
- Header Length: It takes 4 bits for giving the length of head, the TCP header length without any option field is 20 bytes long, and it can have up to 60 bytes.
- Flags: it takes 6 bits, this field has following identifiers: SYN, FIN, ACK, PSH, RST, URG, among them, the first five identifiers mentioned above are most useful for the malware analysis. URG means urgent pointer is valid, SYN indicates that a connection is established, FIN means the connection is closed. ACK indicates response and PSH means there is data transmission, and RST indicates a reset connection.

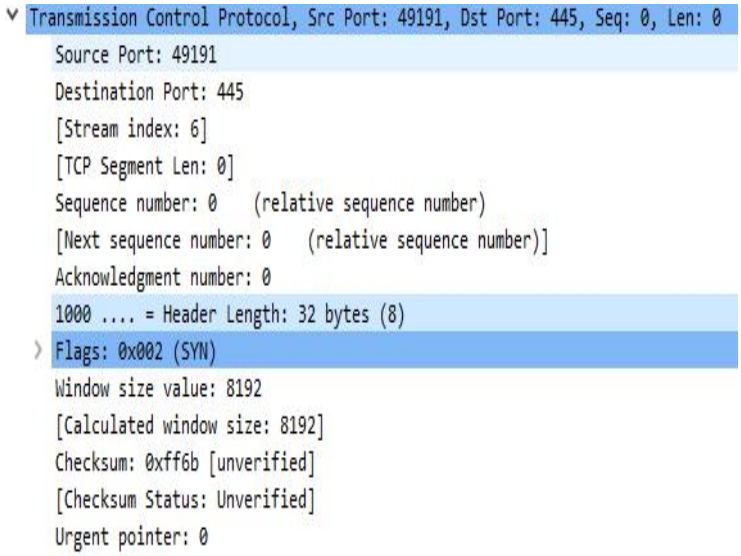


**Fig.2.4** Transmission Control Protocol

Fig.2.5presents the raw byte of one packet, the raw byte sequence length of each layer is fixed if there is not any option filed and all information can be represented by hexadecimal, such as, the flags in the transport layer can be represented as 0002. In the feature extraction section, we ignore TCP payloadfor next research because there are lots of noise in payload, which will affect the overall learning ability of models.

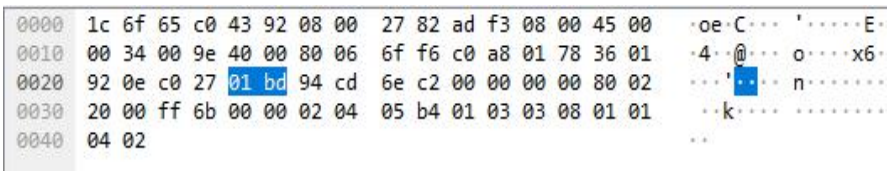


**Fig.2.5** Raw Byte Information

## 2.3 Malware Classification Techniques

With the development of scientific computer, AI technology is becoming more and more involved, such as face recognition, transportation, biology, medical, etc., but AI technique is relatively less applying into the area of cybersecurity, my research is focusing on automatically malware detection using AI techniques based on network traffic analysis, this section introduced some related work and research based on this topic.

Even the goal of each paper is to detect malware, there are lots of different ways for dataset selection, for example, Paul Prasse et al[22] proposed to analyse HTTP/HTTPStraffic, they collect 171 small to large computer networks as dataset; Dmitri et al[5] analyse network based on internet, transport and application layers to select some key attributes, for example, TCP source port, destination port, packet size, number of packets with the push bit set,

number of out-of-order packets and so on; Irfan et al[23] used packet contents as dataset, and Wei, Ming et al[16] proposed that use seven layer's network traffic information and application payload as dataset and using convolution layer to extract feature automatically. Edward et al [6] instead took a static analysis approach, and they looked at information from the binary program without running the application, and also build the neural network by analysing the binary program to determine maliciousness. On my paper, I manually analyse each layer and generate the feature based on some feature selection methods, especially focusing on the internet and transport layers.

After analysing the network traffic dataset, the next step is feature extraction, CHIH, NAI-JIAN, et al.[8] on paper "feature selection and extraction for malware classification" proposed two primary techniques for feature extraction: N-gram and TF-IDF, they try conversion feature format from unigram to sixth-gram and compared the accuracy with same algorithm, beside, they also focused on the feature space and computing time, for this problem, they used MapReduce to reduce the overhead time to improve performance by more than 30%.Edward.Jon et al.[6] also put the N-gram technique into to their research, Richard Zak et al[20] also proposed that by creating a file as a sequence of bytes, and using byte n-gram to extract the unique combination of every n consecutive bytes as an individual feature are very effective and their experimental result also proved this point based on two-part malware data sets with different n-grams, and comparing features without n-gram and with n-gram, the accuracy have a great improvement. On the aspect of feature statistics, the CHIH, NAI-JIAN et al [8] Chatchai et al [27] and Edwardet al [20] used TF-IDF to generate the vector and TF-IDF provides a good data structure for the byte classification, compare with one-hot space vector model, TF-IDF using term-frequency and term frequency-inverse to calculate the weight of each feature, on my project, here also has a comparison of frequency counting and TF-IDF (it will be explained in chapter 5.2), and we found frequency counting is better than TF-IDF, because the feature is not continuous raw byte data but the specific attributes raw byte we selected. Feature selection and reduction is proposed in many different ways in these research papers, such as in Chatchai et al [27] proposed sequential floating forward selection (SFFS) procedure, Edward et al come up with information gain, KL, Gini as feature selection models, and CHIH et al [8] using PCA algorithm to determine the optimal reduced feature set that yields the optimal accuracy and learning times.

My researched papers list lots of classification algorithms foe malware detection, like Chin, Naiet al [8] mentioned that SVM is a good machine learning algorithm for classification and Chatchai et al [27] used C4.5 decision tree, ANN and SVM with different n-gram, this paper conduct that 4-gram can get a high accuracy especially with SVM model from fig.2.6, the details of SVM and decision tree will be explained in chapter 6 as well.

TABLE II. MALWARE CLASSIFICATION PERFORMANCE

| N-GRAMS | Accuracy | | |
|---|---|---|---|
| | *DT (C4.5)* | *ANN (RProp MLP)* | *SVM* |
| 1-gram | 71.83% | 72.29% | **75.44%** |
| 2-gram | 82.10% | 82.85% | **83.91%** |
| 3-gram | 88.74% | 84.82% | **92.96%** |
| 4-gram | 91.25% | 88.31% | **96.64%** |

**Fig.2.6**Accuracy Comparison with Different Algorithms [27]

Dmitri et al [5] proposed Naive Bayes, J48 and Random Forest is good to detectunknown malware, as observed in Fig.2.7, the Random Forest algorithm is able to detect all new families with very high accuracy, while the Naïve Bayes—a very simple algorithm—it got much lower accuracy than the Random Forest algorithm in some complex case.

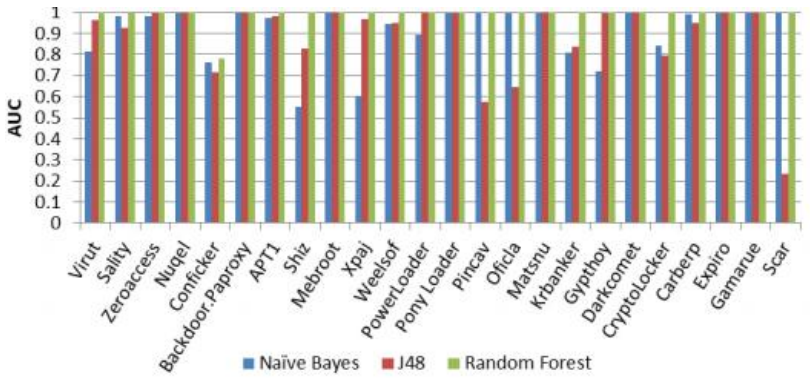


**Fig.2.7**Accuracy Comparison with Different Algorithms [5]

Wei, Ming et al [4] proposed analysing malware raw byte traffic data by applying representation learning approach, the main technique they used is a convolutional neural network (CNN), which is currently one of the most popular representation learning methods, they didn't use any feature extraction technique to analyse or process the traffic data but took raw byte of all layers as " an images " and then used CNN to training data as image classification, their evaluations score (shown as Fig.2.8) is higher than other algorithms, most of classification scores over 98% and some of the malware or benign instances are 100% correctly classified.

| DATA | PR | RC | F1 | DATA | PR | RC | F1 |
|---|---|---|---|---|---|---|---|
| Cridex | 100 | 100 | 100 | BitTrt | 100 | 100 | 100 |
| Geodo | 100 | 99.9 | 99.9 | Facetime | 100 | 100 | 100 |
| Htbot | 99.8 | 100 | 99.9 | FTP | 100 | 100 | 100 |
| Miuref | 100 | 100 | 100 | Gmail | 98.4 | 99.3 | 98.8 |
| Neris | 96.3 | 92.9 | 94.6 | MySQL | 100 | 100 | 100 |
| Nsis-ay | 99.8 | 99.0 | 99.4 | Outlook | 99.2 | 98.1 | 98.7 |
| Shifu | 99.9 | 99.9 | 99.9 | Skype | 99.8 | 100 | 99.9 |
| Tinba | 100 | 100 | 100 | SMB | 100 | 100 | 100 |
| Virut | 90.7 | 95.6 | 93.1 | Weibo | 100 | 100 | 100 |
| Zeus | 100 | 100 | 100 | Wow | 100 | 99.9 | 99.9 |

**Fig.2.8**Evaluation Score with CNN [4]

Prasse et al [22] proposed focusing on HTTP/HTTPS analysis, they selected 60 features of URL strings as analysis object, and the main model they proposed is LSTM with two LSTM layers and followed by a fully connected layer with softmax cell to determine the score of the classes malicious and benign. Fig.2.9 shows the architecture of the design, this network is suitable for sequence analysis, and finally LSTM model detects that 50% of all malware at 70% precision or 32% of all malware at 90% precision.

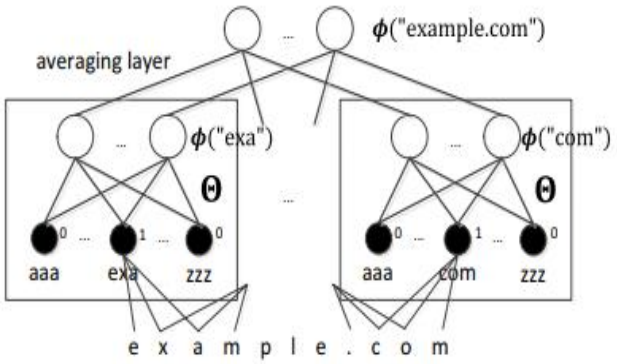


**Fig.2.9**Neural Language Model Network Architecture [22]

Edward, Jon et al [6] proposed a new network architecture MalConv, which based on a combination of convolution and LSTM to process the raw byte sequence, in order to capturing the location invariance, they decide to use convolutional layer and max-polling layer as first layer, because the model is able to generate its activation without consideration of the location of the detected feature, this also is one of CNN's characteristics, and they just put the index of each raw byte into network and training the embedding jointly with the convolution. The use of dilated convolutions to deal with sequence length and then feed embedding into RNN that has been historically considered as sequences analysis, and then followed by fully connected layer, the result shown as Fig.2.10, this algorithm also used in my project and details will be explained in chapter 7.2.

| | MalConv | |
|---|---|---|
| Test Set | Accuracy | AUC |
| Group A | **94.0** | **98.1** |
| Group B | 90.9 | **98.2** |

**Fig.2.10**Result with MalConv[6]

# 3.System Overview

The implementation of this system is mainly based on three steps: datasetcollection, feature extraction, classification; basically, on the aspect of data collection, there are some previous studies. On the Edward et al [6] research, they used static analysis approach to get information from the binary program without in executed environment, On the Boyun et al [7] study, they collected malicious code from malware as the dataset and then attempts to use machine learning and data mining to identify new or unknown malicious code. This paper analysing malware dynamically by focusing on network traffic, the detailed reason for feature selection already mentioned in chapter 2.3.

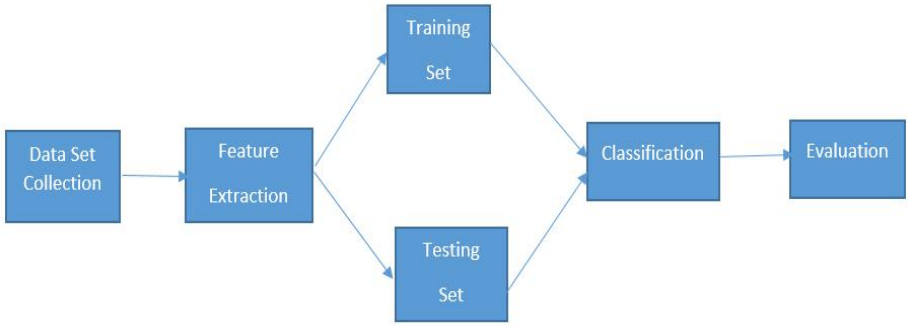


**Fig.3.1**Architecture of the Proposed System

On this paper, we choose packet level as analysis object; the analysis way is focusing on port-based attributes. Fig.3.1 above is a brief architecture of system archit. Firstly introduce dataset that I have collected and then analyse features through feature extraction, feature selection, and feature transform, here focus on n-gram analysis as well, and next we will classify malware and benign network traffic by some classification algorithms, lastly using some classification evaluation methods to evaluate these algorithms, for example, recall, precision, ROC/AUC, etc.

# 4.Dataset Collection

This paper research for malware based on network traffic, and the final dataset consists of two sub datasets: CTU-13 dataset [2] and USTC-TFC2016 dataset [3], the ctu-13 dataset is a botnet traffic dataset and collected by CTU University, Czech Republic in 2011, this dataset has a large capture of real botnet traffic mixed with normal traffic and background traffic. The CTU-13 dataset consists of thirteen captures of different botnet samples [2], this project only use the malware captures dataset as part of my datasets, Fig.4.1 below shows the duration of the malware, the number of packets, the number of NetFlow and the size of packets and the malware names and how many computers are infected.

| Id | Duration(hrs) | # Packets | #NetFlows | Size | Bot | #Bots |
|---|---|---|---|---|---|---|
| 1 | 6.15 | 71,971,482 | 2,824,637 | 52GB | Neris | 1 |
| 2 | 4.21 | 71,851,300 | 1,808,123 | 60GB | Neris | 1 |
| 3 | 66.85 | 167,730,395 | 4,710,639 | 121GB | Rbot | 1 |
| 4 | 4.21 | 62,089,135 | 1,121,077 | 53GB | Rbot | 1 |
| 5 | 11.63 | 4,481,167 | 129,833 | 37.6GB | Virut | 1 |
| 6 | 2.18 | 38,764,357 | 558,920 | 30GB | Menti | 1 |
| 7 | 0.38 | 7,467,139 | 114,078 | 5.8GB | Sogou | 1 |
| 8 | 19.5 | 155,207,799 | 2,954,231 | 123GB | Murlo | 1 |
| 9 | 5.18 | 115,415,321 | 2,753,885 | 94GB | Neris | 10 |
| 10 | 4.75 | 90,389,782 | 1,309,792 | 73GB | Rbot | 10 |
| 11 | 0.26 | 6,337,202 | 107,252 | 5.2GB | Rbot | 3 |
| 12 | 1.21 | 13,212,268 | 325,472 | 8.3GB | NSIS.ay | 3 |
| 13 | 16.36 | 50,888,256 | 1,925,150 | 34GB | Virut | 1 |

**Fig.4.1** Information on each Malware Scenario[2]

USTC-TFC2016 dataset consists of malware traffic and normal traffic, the CTU researchers' collected 10 malware traffics from real network environment, the Fig.4.2 below introducethe description of each malware and some attributes. The normal traffic contains 10 normal flows collected by IXIA BPS, which is a professional network traffic simulation device, to reflect as much traffic as possible, there are eight classes of generic applications traffic out of ten network traffic. Fig.4.3 below shows the information of ten normal network traffic.

| Name | CTU num | Binary MD5 | process |
|---|---|---|---|
| Cridex | 108-1 | 25b8631afeea279ac00b2da70fffe18a | original |
| Geodo | 119-2 | 306573e52008779a0801a25fafb18101 | part |
| Htbot | 110-1 | e515267ba19417974a63b51e4f7dd9e9 | original |
| Miuref | 127-1 | a41d395286deb113e17bd3f4b69ec182 | original |
| Neris | 42,43 | bf08e6b02e00d2bc6dd493e93e69872f | merged |
| Nsis-ay | 53 | eaf85db9898d3c9101fd5fcfa4ac80e4 | original |
| Shifu | 142-1 | b9bc3f1b2aace824482c10ffa422f78b | part |
| Tinba | 150-1 | e9718e38e35ca31c6bc0281cb4ecfae8 | part |
| Virut | 54 | 85f9a5247afbe51e64794193f1dd72eb | original |
| Zeus | 116-2 | 8df6603d7cbc2fd5862b14377582d46 | original |

**Figure 4.2** USTC-TFC2016 PART1 (malware traffic)[3]

| Name | Class | Name | Class |
|---|---|---|---|
| BitTorrent | P2P | Outlook | Email/WebMail |
| Facetime | Voice/Video | Skype | Chat/IM |
| FTP | Data Transfer | SMB | Data Transfer |
| Gmail | Email/WebMail | Weibo | Social NetWork |
| MySQL | Database | WorldOfWarcraft | Game |

**Fig.4.3** USTC-TFC2016 PART2 (normal traffic)[3]

I choose part of CTU-13 dataset [2] and USTC-TFC2016 dataset [3] as the final dataset, and it contains different kind of network traffic, HTTP, P2P, FTP, SMTP and so on, it is shown in Fig.4.4, the network traffic data can be represented in follow layer: physical layer, data link layer, network layer, transport layer, and session layer, presentation layer and application layer, on this paper we just focus on transport layer and internet layer, because the data on these layers contain some traffic characteristic information. For example, port information in the transport layer can identify most applications using standard port numbers and sometimes, tag information can identify network attacks such as SYN attacks and RST attacks[4]. Fig.4.4 below represents the dataset for malware and benign traffic distribution situation. Here are 23 malware pcap files and 8 benign pcap files, each pcap file contains the different number of packets,all features of each instance is from one packet,apacket is the block of control information and data for one transaction between a host and its network[32].

| Pcap Name | Count of instances | Type |
|---|---|---|
| BitTorrent | 209 | benign |
| Cridex | 136 | malware |
| FTP | 205 | benign |
| Geodo | 113 | malware |
| Gmail | 204 | benign |
| Htbot | 500 | benign |
| Menti | 74 | malware |

| | | |
|---|---|---|
| Miuref | 500 | malware |
| Murlo | 65 | malware |
| Neris | 40 | malware |
| Nsis-ay | 27 | malware |
| Outlook | 500 | benign |
| Rbot | 95 | malware |
| Shifu | 211 | malware |
| Skype | 500 | benign |
| SMB1 | 599 | benign |
| Tinba | 215 | malware |
| Virut | 44 | malware |
| Weibo | 487 | benign |
| Zeus | 500 | malware |
| Dridex | 373 | malware |
| MSILSpy.Agent | 509 | malware |
| PUA AdToolbar | 33 | malware |
| Rbot | 99 | malware |
| RemoteAdmin Ammyy | 151 | malware |
| Sality | 199 | malware |
| TrikBot | 68 | malware |
| Trojan Generic | 77 | malware |
| Upatre | 56 | malware |
| Ursnif | 52 | malware |
| webCompanion | 142 | malware |

**Fig.4.4** Final Dataset

# 5.Feature Extraction Methodology

## 5.1 N-gram analysis

An n-gram is an n-character slice of a long string, which has explored in many tasks that I mention in lit review, this model is highly effective and usually used for text analysis. To my surprise, this technique also worked well surprisingly in network security, here I try using the n-gram to analyse raw byte and I found the accuracy do significantly improved. On this research, Firstly, I use Wireshark to capture all malware and benign traffics and then extract content of each packet into a long string in the form of hexadecimal, based on some cybersecurity and network knowledge, here I choose 7 attributes at network layer and transport layer (the details of these attributes also explained in Chapter 2.3), on the network layer: flags, protocol, and source; on the transport layer: source port, destination port, header length, flags; thus, the original features looks like this: [‘40' , ‘00', ‘06', ‘01', ‘01', ‘21', ‘9e', ‘a1', ‘67', ‘01', ‘bb', ‘80', ‘18'] , after 2-gram extraction, these features look like : ['4000' ,'0006', '0601', '0101', '0121', '219e', '9ea1', 'a167', '6701','01bb', 'bb80', '8018']. The simplest n-gram is 1-gram model, it means each raw byte as a feature, then is 2-grams, tri-grams and so on, fig.5.1 below shows the N-gram formsof 1-gram, bi-grams, and tri-grams. On this project, I choose bi-gram as the feature extraction model because it had the best performance.

Original: [‘40’ , ‘00’, ‘06’, ‘01’, ‘01’, ‘21’, ‘9e’, ‘a1’, ‘67’, ‘01’, ‘bb’, ‘80’, ‘18’]

Bi-grams: ['4000' ,'0006', '0601', '0101', '0121', '219e', '9ea1', 'a167',...].

Tri-grams: [‘400006’,’000601’,’060101’,’010121’,’01219e’,’219ea1’,’9ea167’,...]

**Fig.5.1**N-gram Sample

Fig.5.2 is accuracy comparison of 1 to 5 grams with different algorithms, the experiment result shows as below and best accuracy are given in 2-grams with all algorithms.

| N-gram /Algorithm | LSTM | SVM | Naïve Bayes | Decision Tree |
|---|---|---|---|---|
| 1-gram | 82.3% | 85.2% | 55.5% | 83.2% |
| 2-gram | 87.8% | 88.6% | 88.2% | 86.2% |
| 3-gram | 80.9% | 85.2% | 81.3% | 81.0% |
| 4-gram | 81.0% | 82.1% | 78.0% | 80.0% |
| 5-gram | 72.3% | 75.1% | 74.3% | 72.0% |

**Fig.5.2**N-gram Accuracy Comparison

## 5.2 Bag-of-Word

Bag of word model usually used for text representation, text representation means that words are processed into vectors or matrices so that the algorithm can process them. Text

representation is divided into discrete representations and distributed representations. The representative of the discrete representation is bag of word model, the bag of word contains many different algorithm patterns, e.g. one-hot, TF-IDF, n-gram, all can be considered as bag of word model, and the distributed representation is word embedding, and the classic model is word2vec, Glove, ELMO and so on. On this paper, here the original data set is raw bytes, thus, after extracting features and convert the format into n-gram, the next thing is that using bag of word model to represent raw bytes. The fig.5.3 below showing how the bag of model works:

```
Vocabulary:['4000','0006','0601','0101','0121','219e','9ea1','a167','3214','c432','34a1',060
1, 219e]
Vector:['4000','0006','0601','0101','0121','219e','9ea1','a167']
Bag of word vector:[1,1,1,1,1,1,1,1,0,0,0,0,0]
```

**Fig.5.3**Bag of Word Vector

TF-IDF also is one of the models of bag of word, the main idea of TF-IDF is that if a term or phrase appears in an article with high frequency and rarely appears in other articles, the term or phrase is considered to have good classify ability and is suitable for classification, it generally used for analysing text, but for my project, TF-IDF model didn't play a big role comparing with term counting, although the dataset is byte sequence, each raw byte can be regarded as unit and here are not big relationship between contexts, the fig 5.4 shows that the accuracy of term frequency is better than TF-IDF on average (only decision tree got improvement when using TF-IDF).

| | LSTM | SVM | Naïve Bayes | Decision Tree |
|---|---|---|---|---|
| TF-IDF | 84.8% | 88.2% | 86.6% | 87.6% |
| Term Counting | 87.8% | 88.6% | 88.2% | 86.3% |

**Fig.5.4**Result between TF-IDF and Term Counting

## 5.3 Principal Component Analysis

PCA is a very famous algorithm for feature dimension reduction, feature extraction is a subset of feature reduction, thus, PCA also is a feature extraction algorithm, PCA generally used for analysing numerous interrelated variables, and especially suitable for the dataset which has as much variation as possible, the basic PCA implementation steps are:

- For a matrix X, the dimension shape is [m, n], it means there are m samples and for each sample has n features.
- Get the transpose of the matrix after subtracting the mean from each dimension of data on the original matrix and then standardize it, fig.5.5 shows some basic functions of mathematical statistics.

Mean : $$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

Standard Deviation: $$s = \sqrt{\frac{\sum_{i=1}^{n} (X_i - \bar{X})^2}{n-1}}$$

Variance: $$s^2 = \frac{\sum_{i=1}^{n} (X_i - \bar{X})^2}{n-1}$$

**Fig.5.5**Functions of Mathematical Statistics [9]

- Find the eigenvalue(ew) and eigenvector(ev) of XTX or Cov(XT), where the dimensions are: ew(1,n), ev(n,n), here the Cov(XT) means the covariance, covariance is a statistic method used to measure the relationship between two random variables. However, the covariance only can describe 2-dimensions problem, for multiple dimensions, the covariance matrix should looklike Fig.5.6 below. According to the covariance function, here we can calculate the covariance of every two dimensions, the example is shown as Fig.5.7, the first image of Fig.5.8 shows the matrix and the second and last matrix shows the eigenvalues and eigenvalues of the covariance by using the function of eig (cov).

$$cov(X,Y) = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{n-1}$$

**Fig.5.6**Covariance Function[9]

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

**Fig.5.7**Covariance Matrix for three Dimensions Dataset[9]

$$\text{COV} = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

**Fig.5.7** Covariance Matrix[9]

$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & .677873399 \\ .677873399 & .735178656 \end{pmatrix}$$

**Fig.5.8** Eigenvalues and Eigenvectors[9]

- According to the decreasing order of ew, sort the ew and then sort the corresponding column of ev according to ew's order, and find the first K columns of ev to get a new matrix V, the dimension of V is (n,k) at the moment, if we see the Fig.5.8 above, the eigenvalue 1.2840 is far better than 0.49083, so the second dimension in eigenvectors is better than the first one, the newest matrix V is fig.5.9 below.

$$\begin{pmatrix} .677873399 & -.735178656 \\ .735178656 & .677873399 \end{pmatrix}$$

**Fig.5.9**Sort eigenvectors according to the order of eigenvalues[9]

$$\begin{pmatrix} .677873399 \\ .735178656 \end{pmatrix}$$

**Fig.5.10**First K dimension[9]

- Xnew=X*V, the dimension of Xnew is (m,k) at the moment, Where rowFeatureVector is the transpose of the matrix composed of the pattern vector as the column, so its row is the original pattern vector, and the feature vector corresponding to the largest eigenvalue is in the top row of the matrix. rowdataAdjust is the transposition of the matrix formed by subtracting the mean from each dimension of data.

$$FinalData = rowFeatureVector \times rowdataAdjust$$

**Fig.5.11**Final Data[9]

The research and experiment proved that PCA algorithm does works for some algorithms, and the executing time reduced, however, for some algorithms, the accuracy sharply reduced, the Fig.5.12 shows the experiment results about detection rate and execution time with the same dataset before and after using PCA, overall, it shows a bad detect rating if use PCA in my work .

| algorithms | Accuracy without PCA | Accuracy with PCA | Time interval without PCA(s) | Time interval with PCA(s) |
|---|---|---|---|---|
| Decision Tree | 88.7% | 84.3% | 11.04 | 10.4 |
| Naïve Bayes | 86.0% | 63.7% | 6.06 | 1.49 |
| SVM | 87.6% | 87.6% | 8012 | 1963 |

**Fig.5.12**ExperimentAccuracy and Time Result with PCA and without PCA

# 6.Classier Description

## 6.1 Decision Tree

A decision tree is used to learn a classification function which concludes the value of a dependent attribute given the values of the independent attributes [10], its central idea is very simple, similar to the process that people make decisions. The decision tree structure is same with a tree logically, fig.6.1 shows a very simple decision tree structure, and it includs root nodes, internal nodes, and leaf nodes, and there are some important concepts: entropy, information gain, Gini coefficient.

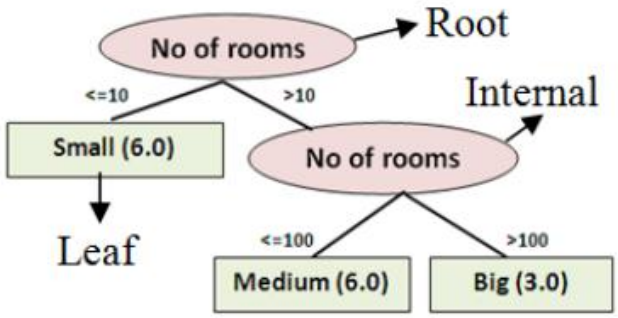


**Fig.6.1**Decision tree structure[11]

Entropy is a measure of the disorder of data, and also called a measure of uncertainty in any random variable. From the entropy, we can easily get the point that the higher the entropy, the higher of the uncertainty, conversely, the lower of the entropy, the smaller of uncertainty. The function of entropy is shown below Fig.6.2 where the Pk means the proportion of samples with feature K in the current node.

$$Ent(D) = -\sum_{k=1}^{|\gamma|} p_k log(p_k)$$

**Fig.6.2**Entropy and conditional Entropy[12]

Information gain is used for measuring the association between inputs and outputs [11]. Fig.6.3 is the function of information gain, Ent(D) represents the information entropy of D and there are no associations with attribute a, so $\sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$ represents the information entropy of branch node, if it is lower, it means the greater of the overall purity of the branch node, the larger the information gain becomes, this also is our goal to maximize the information gain.

$$Gain(D,a) = Ent(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Ent(D^v)$$

**Fig.6.3**Information Gain[12]

The basic concepts of the decision tree are information entropy and information gain, which are used to judge which features purer as a branch node. On this paper, the experiments prove that the decision tree model has good classification ability for malware analysis, ROC graph (Fig.6.4) shows the average AUC of the total dataset and specific AUC of each class respectively.

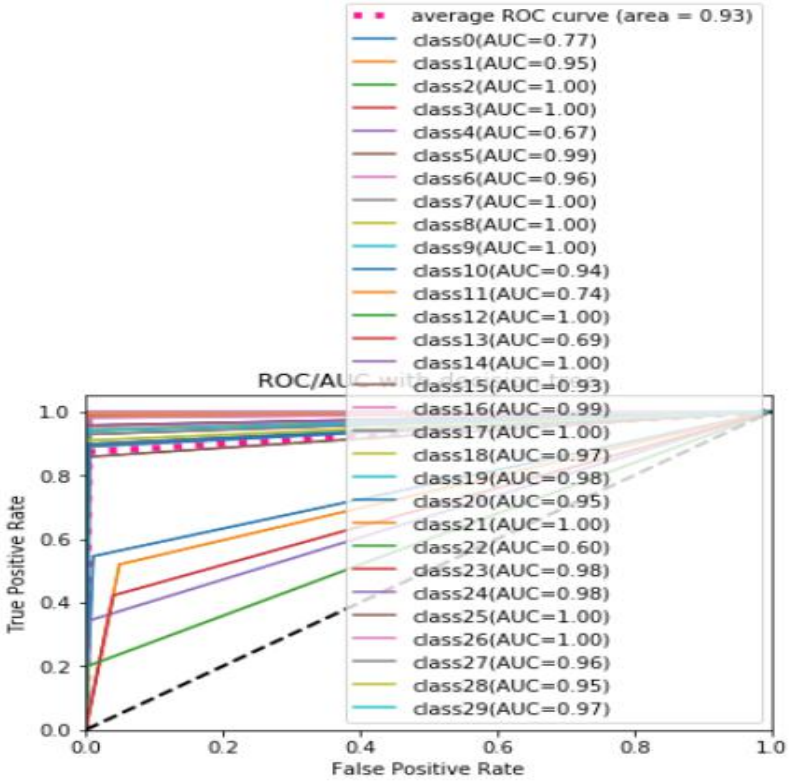


**Fig.6.4** ROC of Decision Tree

## 6.2 SVM-Support Vector Machines

Support vector machines are supervised, learning model, it generally can be used for classification and regression, the principle of SVM basically is finding optimal hyperplane as classification boundary, and make sure that the optimal hyperplane is to be as far away as possible from all data points to make the margin largest, thus, maximized the margin between two classes 1 and -1 is the way to find the optimal hyperplane. The block line in the middle in fig.6.5 is optimal hyperplane, it has the largest distance with class -1 and class 1 respectively, and fig.6.6 is the formula of distance function from point to hyperplane.

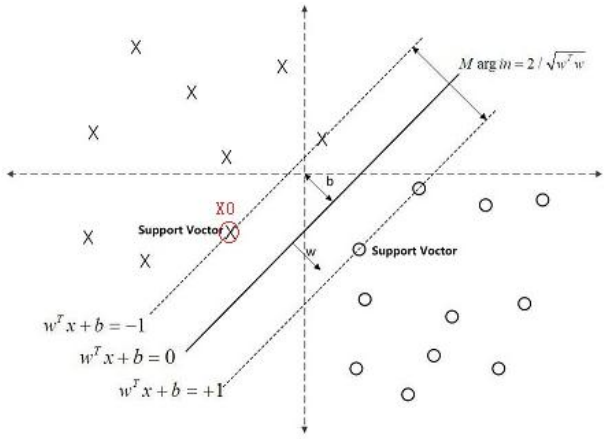


**Fig.6.5** LinearSupport Vector Machine[33]

$$\frac{1}{\| w \|} | w \bullet x_0 + b |$$

**Fig.6.6** DistanceFunction from Point to Hyperplane[33]

To maximizing the margin to find optimal hyperplane, specifically, this problem can be represented as the following constraint optimization problem:

$$\max_{w,b} \quad \gamma$$

$$\text{s.t.} \quad y_i \left( \frac{w}{\| w \|} \cdot x_i + \frac{b}{\| w \|} \right) \geqslant \gamma, \quad i = 1, 2, \cdots, N$$

**Fig.6.7**Constraint Optimization Problem[33]

Meanwhile, we want to maximize the hyperplane (w,b) margin of the training dataset ϫ , the constraint indicates that the hyperplane(w,b) has a geometric margin with each training sample point, it is at least ϫ . Considering some relationships between geometric margin and function margin, and the function equivalence problem, the following function optimization problem of linear separable support vector learning is finally obtained through transformation. Fig.6.8 is an optimization problem of linear separable support vectors, and it also is convex quadratic programming.

$$\min_{w,b} \quad \frac{1}{2} \| w \|^2$$

$$\text{s.t.} \quad y_i (w \cdot x_i + b) - 1 \geqslant 0, \quad i = 1, 2, \cdots, N$$

**Fig.6.8** Primal optimal problem[33]

In order to get hyperplane and decision function, here introduced Lagrange function, where apply the Lagrangian duality, and then get the optimal solution of primal problem through

analysing dual problem, this is dual algorithm for linear separable support vector machine, according to Lagrangian duality, the dual problem of primal problem is maximal and minimal, it is shown as fig.6.9, in order to solve dual problem, firstly, we find the minimum of L(w,b,a) for parameter w and b, then calculating the maximum value of a, this is a dual optimization problem.

$$\max_{\alpha} \min_{w,b} L(w,b,\alpha)$$

**Fig.6.9**Maximal and Minimal Problem[33]

After optimization, the final expression can be obtained, which is shown as fig.6.10, considering the primal optimization problem (shown as fig.6.8) and dual optimization problem (shown as fig.6.10), the primal problem meet the condition of principle c.2, thus here existing w*, a*, b* that w* is the solution of primal problem and a* and b* is the solution of dual problem, this means that solving the primal problem can be converted to solving the dual problem. Therefore fit a given linearly separable training data set, the solution a* can be solved first and then the solution w*,b* of a primal problem can be obtained, thus the final separate hyperplane and classification decision function can be obtained.

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i y_i = 0 \\ & \alpha_i \geqslant 0, \quad i = 1,2,\cdots,N \end{aligned}$$

**Fig.6.10** Dual optimal problem[33]

Finally, getting the optimal solution of separate hyperplane and classification decision function after some mathematical calculations that I mentioned above, it's shown as fig.6.11 and fig.6.12.

$$\sum_{i=1}^{N}\alpha_i^* y_i (x \cdot x_i) + b^* = 0$$

**Fig.6.11** Hyperplane Function[33]

$$f(x) = \text{sign}\left(\sum_{i=1}^{N}\alpha_i^* y_i (x \cdot x_i) + b^*\right)$$

**Fig.6.12**Decision Function[33]

Generally to say, SVM only can be used for binary classification, but some methods are developed for multiple classifications, On this paper, we use “one to rest” method, one to rest means in the training, the samples of certain category are classified into one class and the other remaining samples are classified into another class, so that the samples of the k categories construct K SVMs, when classifying dataset, the unknown samples are classified as the one with the largest vote value, the disadvantage of this algorithm is that the one to rest method will cost expensive runtime with the increase of label class numbers, thus, SVM model with one to rest method has high time complexity (it’s shown in Fig.6.12 above, the SVM execution time up to 8012 s with 30 classes), but the SVM classification has a good and stable performance as well, fig.6.13 shows that the accuracy reach 88% and the average AUC is 0.94.

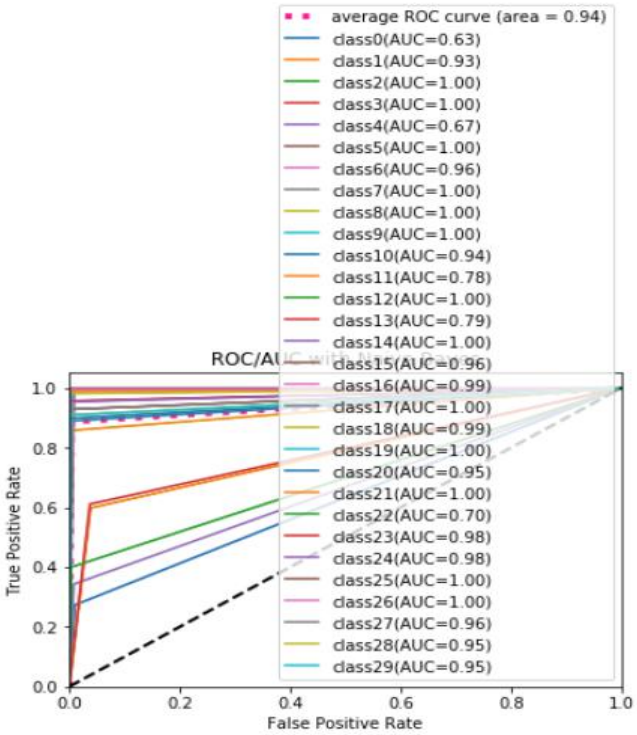


**Figure 6.13** ROC with SVM

## 6.3 Naïve Bayes

Naive Bayes algorithm is a supervising algorithm, which can be used for classification. It is based on "naive" assumption of conditional independence between every pair of features given the value of the class variable [13]. The Bayes formula is shown as fig.6.14, on the condition of x1,…xn, the probability of y is equal to the probability of y times the probability of x,..xn of y given divided by the probability of x1,…xn. A naive condition independence assumption is shown as fig.6.15, that is, given the condition of y, the probability of generation between X1,…., Xn is completely independent, the generation process is that:

- Firstly get y and the probability of y to decide it’s positive or negative.

- Then make sure the feature-length and feature probability of X1 …Xn on the condition of label y.

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)}$$

**Fig.6.14**Bayes' Relationship[13]

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

**Fig.6.15** NaïveConditional Independence Assuption[13]

Naive Bayes algorithm is extremely fast compared with other algorithms such as decision tree, SVM, because each feature distribution can be estimated as one-dimensional distribution. However, Bayes classifier is known as a bad estimator, it is a stronger assumption, but there are few applications to meet this situation, especially for text classification, each feature in the text are totally independent and there is no relationship between each word, thus this is a weak classifier for malware detection task, the accuracy reduced very fast when the input vector format changed or dimension reduced, for example, if using TF-IDF model to extract features, and then using different algorithms to test it, the result is: the accuracy of Naïve Bayes model reduces to 60% while other algorithms don't decrease. Fig.6.16 shows the AUC value of each class with Naïve Bayes, the average AUC is 93%.

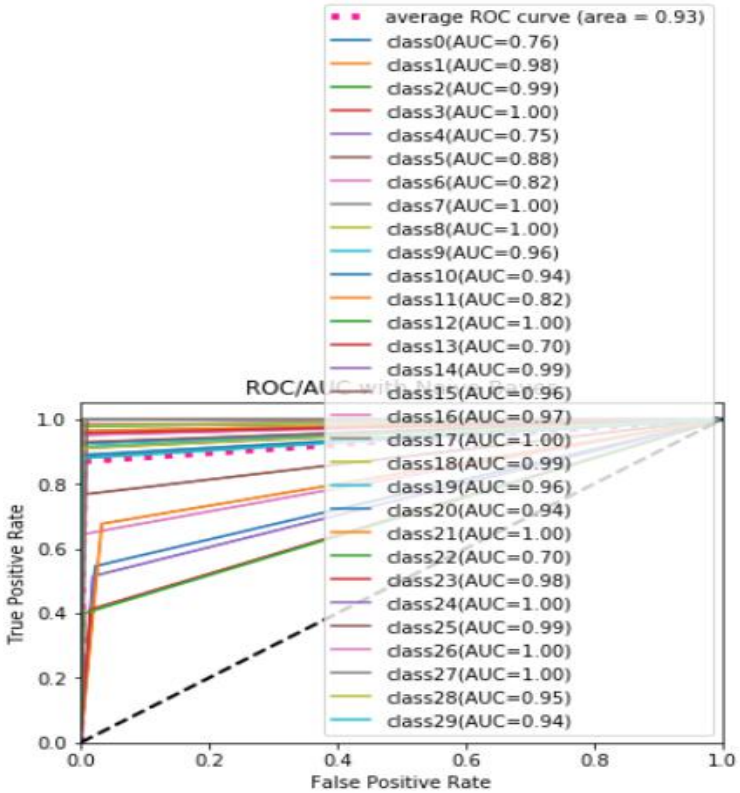

**Fig.6.16** ROC with Naïve Bayes

## 6.4 LSTM- Long Short Memory

Before illustrated LSTM, this paper introduces another network called recurrent neural network, RNN contains input, hidden layer, and output. The difference between RNN and neural network is that it introduced recurrent concept, RNN assumes that our samples are sequence-based, like fig.6.17 below, there are n index numbers and for each corresponding sample sequence, it has a corresponding input x(t), the hidden state h(t) of the model at the position of the sequence index t is determined by x(t) and the hidden state h(t-1) at the t-1 position. In each sequence index number t, we have a corresponding model prediction o(t) and loss value l(t) and real label y(t) to get a backpropagation.

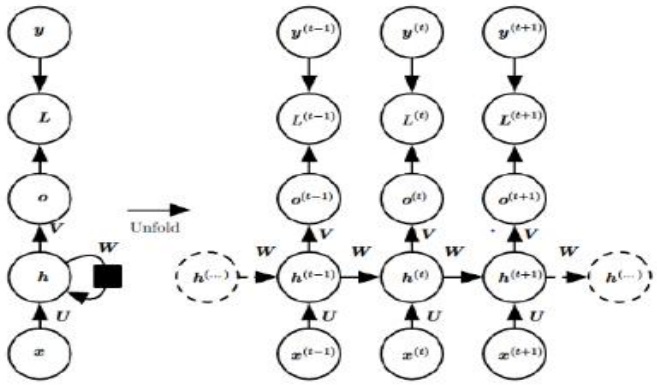


**Fig.6.17**RNN Architecture[15]

Although RNN can theoretically solve the sequence data very well, it also has the problem that the gradient disappears like DNN. When the sequence is very long, the problem is especially serious. Therefore, the above RNN model is generally nor directly applicable to some applications domain, for example. In the field of NLP in speech recognition, handwriting, and machine translation and so on. This also is a reason why LSTM model is introduced, due to the disappearance of the RNN gradient, people have improved the hidden structure of the sequence index position t, it's shown as fig.6.18, here majorly add three gates into a cell, they are read gate, forget gate and write gate respectively.

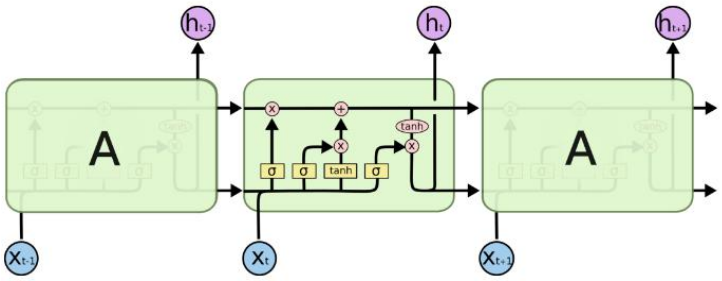


**Fig.6.18**LSTM Architecture[14]

Forget gate can control memories. In LSTM, the probability of forgetting the hidden cell state of the last layer is controlled with a certain probability, which is shown as fig.6.19, the process is when input hidden state h(t-1) of last layer and new sequence input x(t), a sigmoid function will be activated and then get the forget output f(t).

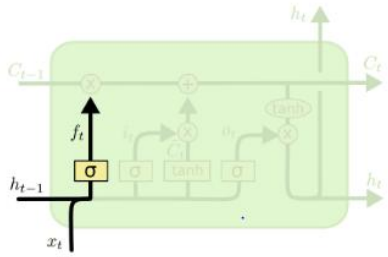


**Fig.6.19**Forget Gate Architecture[14]

The input gate is responsible for processing the input of the current sequence position. Its substructure is shown as fig.6.20, from the graph we can see that first part use a sigmoid activation function and then feed input into a tanh activation function, and using two results product to update cell state.

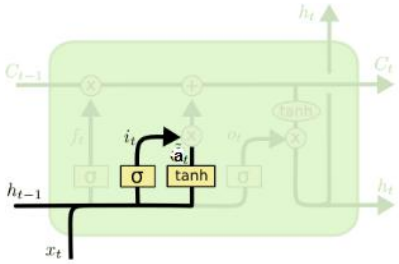


**Fig.6.20**Input Gate Architecture[14]

With the new hidden cell state C (t), we can look at the output gate, from the fig.6.21 we can see the hidden state consists of two parts, one is o(t), which is from last hidden state h(t-1) and new sequence data x(t) and activation function sigmoid, another part is from hidden state C(t) and tanh activation function.

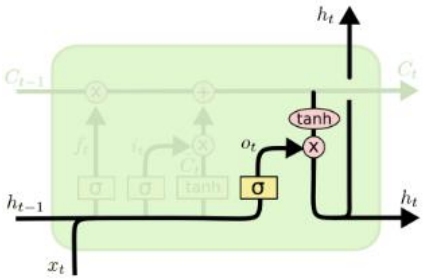


**Fig.6.21**Output Gate Architecture[14]

The difference of LSTM and other networks is that it is a sequence model and each hidden state contains last hidden state, the network not only learn the new text input, the text context

also can affect the model parameter, thus, LSTM model is suitable for text classification, on this paper, we firstly use a  normal neural network with 1 embedding layer, 1 LSTM layer and 2 dense layers after feature extraction and reduction by PAC, the performance is shown as fig.6.22, when epoch number reached to 3, the loss value and accuracy nearly stop increasing, from the graph we can see there is a very low accuracy about 59%.

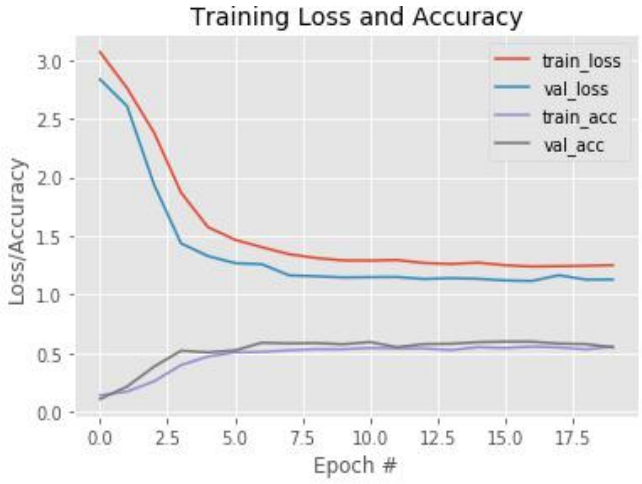


**Fig.6.22** Loss/Accuracy with LSTM Architecture

# 7.Model Enhancements

## 7.1 Ensemble Learning –Majority Vote

Majority vote is a combination strategy, it generally used for the classification task. The majority vote usually used for solving the weak classifier problem, when I use three or more classifiers to training dataset and get low accuracy for each classifier, a better way to enhance the accuracy is that ensemble three or more classifiers together, and then using three or more trained classifiers to predict and using majority vote way to get major prediction. Fig.7.1 is a formula of majority vote, where dt, j is 1 or 0 depending on whether classifier t chooses j, or not, respectively. The ensemble then chooses class J that receives the largest total vote [18]:

$$\sum Tt=1 dt,J(\mathbf{x})=\max j=1,\cdots,C\sum Tt=1 dt,j$$ [18]

**Fig.7.1**Majority Vote Formula

The majority vote in this paper is that vote for major prediction, Thus, under the condition of the classifier output is independent, it can be shown that the majority voting combination will always result in an effectively large number of classifier performance improvements. If there are a total of T classifiers for two types problems, then if there is at least the [T/2+1] classifiers have a correct prediction, and the overall decision will be correct. Fig.7.2is a result of four algorithms comparison, the experiment result proves that the majority vote has the best detection rate in all algorithms.

| | Naïve Bayes | SVM | Decision Tree | Majority Vote |
|---|---|---|---|---|
| Accuracy | 0.88 | 0.88 | 0.86 | 0.89 |

**Fig.7.2**Comparison of Majority Vote and Other Algorithms

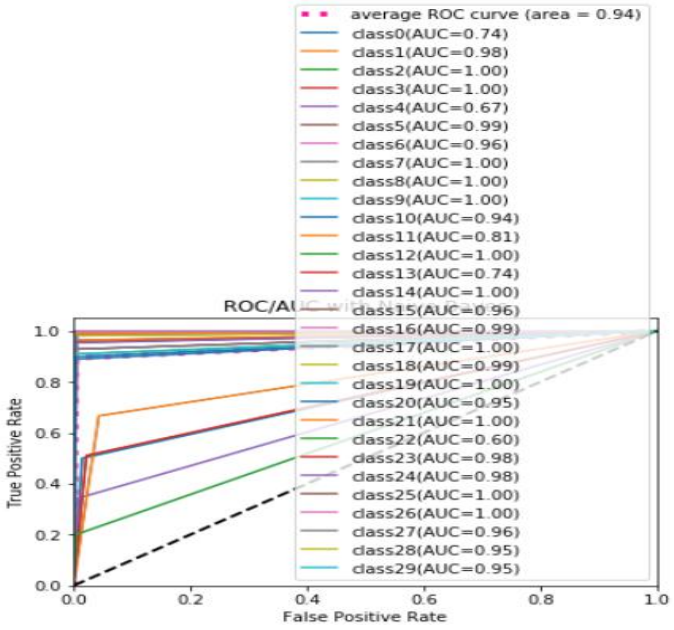


**Fig.7.3**ROC/AUC with Majority Vote

## 7.2 Combination of Network Architecture

Fig.7.4 is an enhanced network architecture based on Edward, Jon et al [6] research, the major difference with the first network [chapter 6.4] is that here add a convolutional layer and max-pooling layer as feature filter. This neural network regard malware dataset as an image, the malware ‘image' is like a dilated 1D convolution. Here we just choose processing each vector with large features in a large convolution, ignoring the feature order and feature position and feature weights and so on, the reason for using max-pooling is that solve the issue of information sparsity problem, and then feed all features into LSTM time sequence, for this network, the advantage is that after first epoch we can see the validation accuracy up to around 78%, it can learn very fast compared with the first neural network, but it also has some issues, there has over-fitting after 3rd epoch if we see the validation loss value, even the validation accuracy is slowly increasing. However, overall the accuracy reached 88% and more effective than the first neural network model. Fig.7.5 shows the loss and accuracy situation.

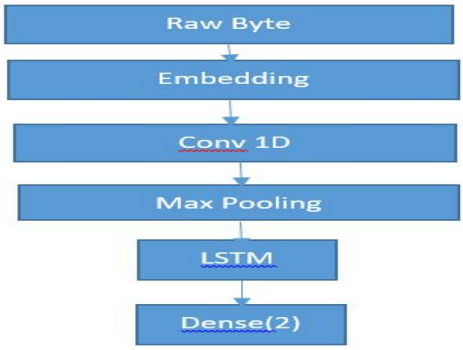


**Fig.7.4**Structure Combination of LSTM and CNN

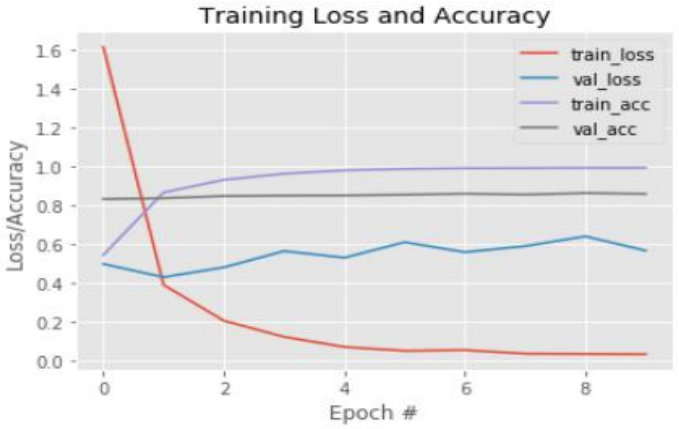


**Fig.7.4**Loss/Accuracy with CNN and LSTM architecture

# 8.Classification Evaluation

## 8.1 ROC /AUC

ROC/AUC is a model evaluation indicator used for the evaluation of binary classification. AUC is the abbreviation of 'Area under Curve ', and this Curve is Roc curve. It illustrates how much model is capable of distinguishing between classes, the higher the AUC values, the better the model is for classifying.

## 8.2 Precision

This is an evaluation of positive prediction, i.e., the percentage of detected malware of all samples.

$$Precision = \frac{TP}{TP+FP}$$

## 8.3 Recall

Recall means the percentage of correct malware out of all malware samples.

$$Recall = \frac{TP}{TP+FN}$$

## 8.4 F-1 Score

F-1 score is a combination of precision and recall.

$$F-measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

## 8.5 Confusion Matrix

The confusion matrix is an error matrix, which is used to visually evaluate the performance of supervised learning algorithms. The confusion matrix size is the square matrix of (n_classes, n_classes), each row of this matrix represents an instance in the real class, and each column represents an instance in the prediction class.

# 9.Results

## 9.1 Binary Classification Result

The overall results in Fig.9.1 is for the evaluation of binary classification, only for malware and benign, Fig.9.2 shows the confusion matrix of each algorithm, from both results we can see, most of the instances can be classified correctly to positive (malware) or negative (benign), The average accuracy over than 98% for each algorithm, only the false-negative rate of Naïve Bayes is lower than other algorithms.

| | Naïve Bayes | LSTM | SVM | Decision Tree | Majority Vote |
|---|---|---|---|---|---|
| Accuracy | 0.98 | 0.98 | 0.99 | 0.99 | 0.99 |
| Precision | 0.99 | 0.98 | 0.98 | 0.98 | 0.99 |
| Recall | 0.97 | 0.99 | 0.99 | 0.99 | 0.99 |
| F1_score | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 |

**Fig.9.1**Evaluation of each Algorithm

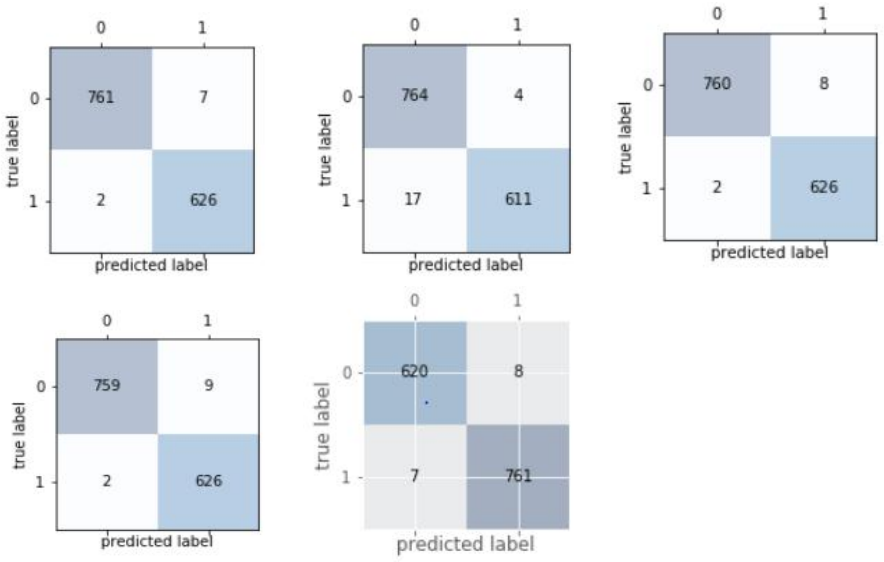


**Fig.9.2** Majority Vote (1), Naïve Bayes (2), Decision Tree (3), SVM (4), LSTM (5)

## 9.2 Multiple Classification Result

Fig.9.3 and Fig.9.4 represent the result of multiple classification results, it's for the result of each kind of malware and benign families, Fig.9.4 is AUC of multiple classification, from the image we can see the AUC of BitTorrent, Gmail, Outlook, Skype are much lower than others, like only 60%-70%, it proves that this system actually is not very effective for normal

network traffic classification.Fig.9.5 is a multiple confusion matrix, and we can figure out how many instances are correctly andincorrectly classified.

| | Naïve Bayes | SVM | Decision Tree | Majority Vote | LSTM with 1D CNN |
|---|---|---|---|---|---|
| Accuracy | 0.88 | 0.88 | 0.86 | 0.89 | 0.88 |
| AUC(avg) | 0.93 | 0.94 | 0.93 | 0.94 | 0.93 |

**Fig.9.3**Accuracy for Multiple Classifications

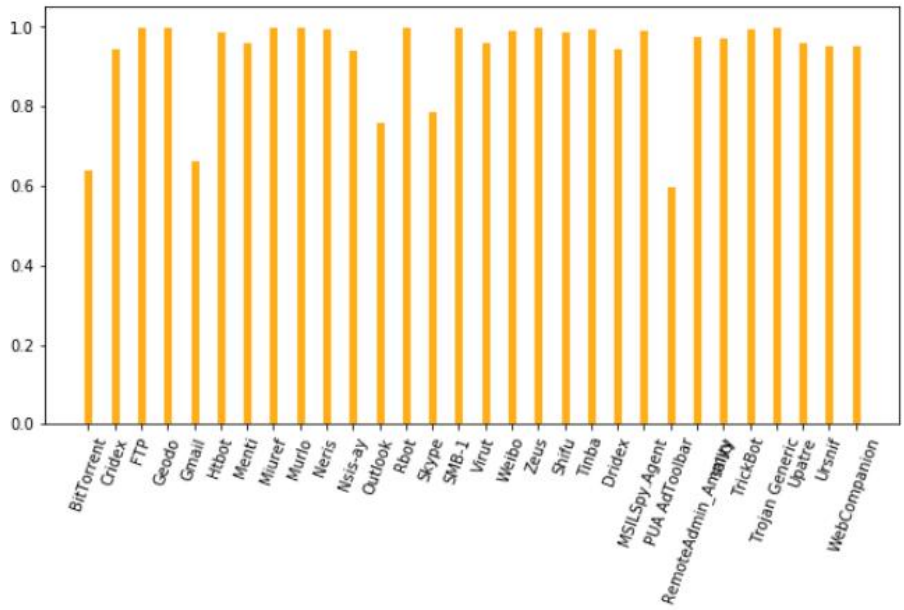


**Fig.9.4** Majority vote with each class flow AUC

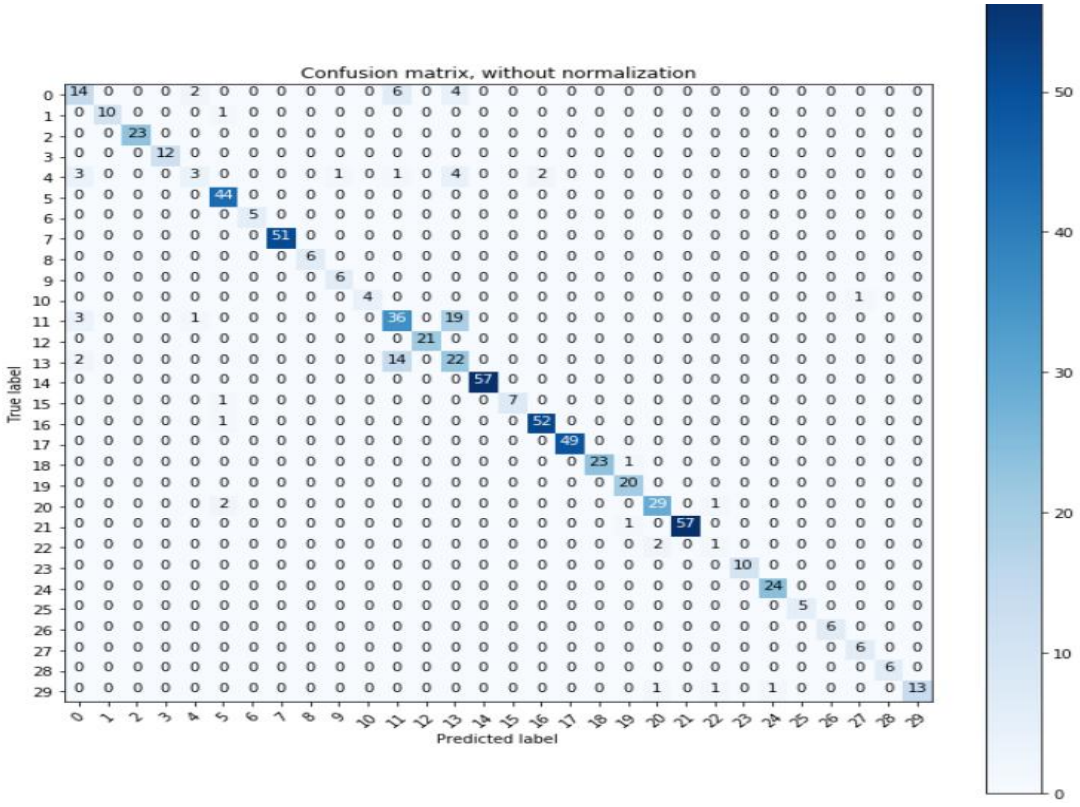


**Fig.9.5**Majority vote with each class flow AUC

## 9.3 Dataset Size Result

This section is the result of whether the dataset size affects the result, the result of this experiment is presented in fig.9.6, X value is the dataset split size and Y value is the accuracy of each algorithm, actually from the dataset split range 9/1 to 1/9 (training dataset size/testing dataset size) to analyse, even there is a small reduction of each algorithm, it's still relativelystable, this result confirms that a suitable amount of data in train set is enough for high accuracy detection of malicious network traffic.

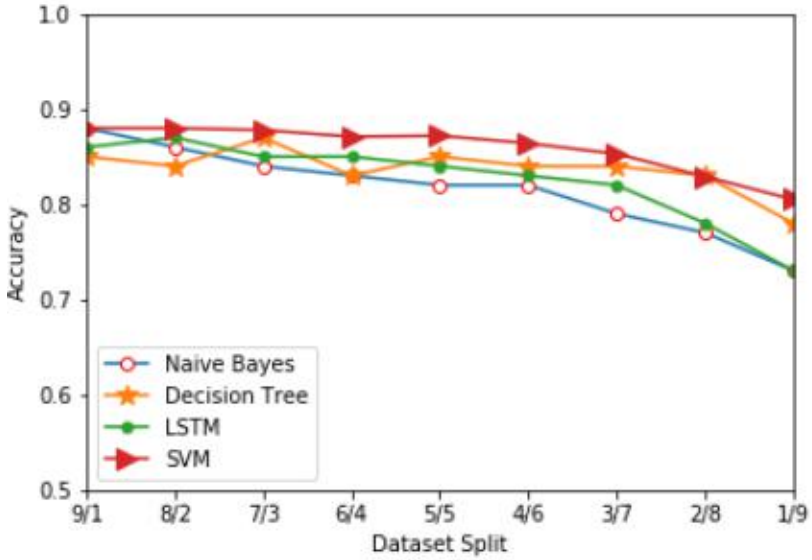


**Fig.9.6**Malicious Network Split Recording Result

# Discussion and Limitations

We have developed and studied a malware-detection model based on some machine learning algorithms that use network traffic (TCP and IP) as an analysis object. We used an ensemble model, that is, a combination of three machine learning algorithms: Decision Tree, SVM, Naïve Bayes, and then use majority vote method to generate the final prediction, in addition, LSTM is another model we developed to classify malware and benign, the nerual network architecture consists of one embedding layer, one1D Cov layer, one max pooling layer, one LSTM layer and two connected layers with softmax activation. Thus, for the malware detection problem, both of two algorithms have a good detection rate, they are 89% and 88%respectively.

Even two models got a very good accuracy for detect malware and benign,  here also has some limitations in my system, the first limitation is feature reduction problem, when using PCA analyses traffic features, the result shows that the variance of these features are similar, it means that each feature has similar importance, on this paper I tried PCA and Univariate selection to reduce feature dimension, even the runtime is greatly reduced, the accuracy also is reduced, for example, the time for Naïve Bayes is  1.2s, the accuracy is 86% before using PCA model, there is an significantly decrease is that the accuracy reduced to 56%, and the time is 0.71s after using PCA, I think the reason is because that here actually only 13 fixed features and then regards these raw byte as a string or a sequence, but there are not too many connections between each feature, which unlike packet content or payload analysis, here just existing few noisy in these attributes, thus, PCA is not effective for dimension reduction in this case.

The second limitation is the execution time cost for SVM is expensive, this project is multiple classifications, the proposed algorithm SVM is a binary classifier, thus, one to one or one to rest algorithms are introduced to this project, but the shortcoming of these algorithms are that with the increase of data classes and data features, the number of classifiers also increase and the running time will increase as well. For example, we have 30 classes of malware and benign in total, we will build 30 classifiers if using one to rest algorithm, the execution time cost of SVM classifiers reached over 3 hours.

The last limitation is that the consideration of feature extraction is not comprehensive initially. This project is based on the analysis of IP/TCP. The analysis results are shown in Fig.8.4. From the graph, we can see that the AUC of BitTorrent, Gmail, Outlook and Skype are very low, the 99% malware can be correctly classified and only 60% benign traffic can be correctly classified, that’s because when we analyse attributes we just focusing on what characteristics of malware, it's not comprehensive in the aspect of feature analysis, I think not only layer attributes, other attributes like payload or statistical-based attribute also should be considered.

# 10. Conclusion and Future Work

*The experimental results prove that the malware detection based on machine learning can successfully detect whether the software carries viruses by analysing network traffic, through the feature analysis and selection and the improvement of classification methods. The highest detection rate is about 89%. Therefore, using machine learning algorithms to analyse the raw byte sequence based on TCP/IP is effective for automatically detecting malware. Below are detailed summary and future workabout this topic.*

In this work, we analysed the network traffic and feature engineering and evaluated some machine learning algorithms and neural network based on this topic. Firstly, we analysed the network traffic with virus, here we get three sections of network traffic are most easily to expose viral behaviours and characteristics that are different from normal traffic. One is 7 layers attributes, especially for transport layer and internet layer, because it responsible for getting all information and responsible for communicating with layers; second is payload based on attribute, third is statistical based on attribute. We proposed sequences classification by using raw byte, the feature engineering is majorly using N-gram technique to get new features and then comparing term frequency and TF-IDF vector space model[8], then get a conclusion that the term frequency is better than TF-IDF because TF-IDF is more suitable for contextual text analysis, feature reduction is a problem in the project, we proposed two dimension reduction algorithms, they are PCA (primary component analysis) and univariate selection, univariate selection can be used to select these features that have the strongest relationship with the output variable, unfortunately, even these algorithms can reduce execution time cost through reducing feature dimension, the accuracy also has a great decreasing. Especially for Naïve Bayes and LSTM, the accuracy sharply reduced from 88% to 56%. In order to evaluate the robustness of these algorithms, I collect different network traffic with some virus, for example, there are two different network traffics with Robt, and it can make sure that even the server or other condition is different; it still can be correctly classified. For the algorithm analysis, comparing SVM, Naïve Bayes and Decision Tree, with these fixed length of training data, the Naïve Bayes and SVM have a better performance because their accuracy can reach 88% with 2-gram, but compare with robustness, the SVM and decision tree are better, because when change the way of the feature extraction, the accuracy of SVM and Decision Tree just reduced about within 5% while the accuracy of the simple algorithm- Naïve Bayes reduced over 30%. If considering the execution time cost, Naïve Bayes cost the shortest time, only a few seconds, but since SVM is a binary classifier, it will last for a few hours. On the aspect of model enhancement, there are two-part, firstly for machine learning algorithms, here we use majority vote to ensemble three algorithms, the result improved 2%, which is shown in the chapter 10. For neural network, even LSTM has a strong learning ability and has long-short memory for byte sequence analysis, it execution time is too long and causing out-of-memory problem, and the accuracy only reached 60%, then here we proposed a new architecture is adding convolutional layer and max-pooling layer into neural network, because convolution layer can focus on the relationship of local data, and max-pooling layer can reduce the feature dimensions, then feed the embedding into LSTM to learn, the result is that the execution time reduced and the detect accuracy improved over 20%. Overall, for the simple sequence of traffic analysis rather than complex sequence, the performance of machine learning algorithms is better than the neural network.

In future research, I will focus on the network traffic analysis, as I mentioned in the last section, one of the biggest limitations is that the benign traffic analysis result is not very well, the accuracy only reached 60% on average, so for the future research, I would like to further explore the benign software classification issues, like what features influence the benign classification; besides, I hope to change the perspective of feature analysis. TCP payload is a good choice, because payload is a contextual sequence and AI algorithms has a strong ability to automatically learn useful features, which maybe can get better detect rate when put it into LSTM model or using N-gram method, even there is a lot of noise in payload, I think some reduction methods or extraction ways can make payload as dataset to effectively analyse malware and benign.

# References


[1] “Malware Analysis: An Introduction”. Address available: https://www.sans.org/reading-room/whitepapers/malicious/paper/2103

[2]“Ctu13-datasets”. Address available: https://www.stratosphereips.org/datasets-ctu13

[3]“Dataset (ustc-tfc2016)”. Address available: https://github.com/echowei/DeepTraffic

[4] Wang, Wei et al. “Malware traffic classification using Convolutional Neural Network for Representation Learning.” 2017 International Conference on Information Networking (ICOIN) (2017): 712-717. Addressavailable: https://www.jianshu.com/p/42a166d22874

[5] Bekerman, Dmitri et al. “Unknown malware detection using network traffic classification.” 2015 IEEE Conference on Communications and Network Security (CNS)(2015): 134-142. Address available: https://cyber.bgu.ac.il/wp-content/uploads/2017/10/07346821.pdf

[6]Raff, Edward et al. “Malware Detection by Eating a Whole EXE.” ArXiv abs/1710.09435 (2017): n. pag. Address available: https://pdfs.semanticscholar.org/4417/dfcfc722b8b31278a0ebcc1595963dab5a1c.pdf?_ga=2.69544146.37040152.1563540223-1769586260.1563540223

[7]Zhang, Boyun et al. “New Malicious Code Detection Based on N-gram Analysis and Rough Set Theory.” 2006 International Conference on Computational Intelligence and Security 2 (2006): 1229-1232.Address available: https://abuisa.github.io/reff/New-Malicious-Code-Detection-Based-on-N-Gram-Analy.pdf

[8] Lin, Chih-Ta et al. “Feature Selection and Extraction for Malware Classification.” J. Inf. Sci. Eng. 31 (2015): 965-992. Address available: https://www.iis.sinica.edu.tw/page/jise/2015/201505_11.pdf

[9]“PCA introduction”. Address available: https://blog.csdn.net/a8039974/article/details/81285238

[10] “PCA thesis”. Address available: https://www.youtube.com/watch?v=gxOb6DirCJM&t=121s

[11] Bhargava, Neeraj et al. “Decision Tree Analysis on J48 Algorithm for Data Mining.” (2013).Address available: file://fs2/18231917/Downloads/Decision_Tree_Analysis_on_J48_Algorithm.pdf

[12] “Decision tree introduction
”. Address available: https://blog.csdn.net/bravery_again/article/details/81104914

[13]“Naive Bayes”. Address available: https://scikit-learn.org/stable/modules/naive_bayes.html

[14] “LSTM introduction”. Address available: https://blog.csdn.net/zhangbaoanhadoop/article/details/81952284

[15] “RNN introduction”. Address available: https://blog.csdn.net/zhaojc1995/article/details/80572098

[16]“TCP/IP Reference”. Address available: https://nmap.org/book/tcpip-ref.html

[17] “Majority Voting”. Address available: https://www.sciencedirect.com/topics/computer-science/majority-voting

[18] “Voting Based Methods”. Address available: http://www.scholarpedia.org/article/Ensemble_learning#Voting_based_methods

[19]“Wireshark Documentation”. Address available: https://www.wireshark.org/download/docs/user-guide.pdf

[20] Raff, Edward et al. “An investigation of byte n-gram features for malware classification.” Journal of Computer Virology and Hacking Techniques 14 (2016): 1-20. Address available:https://link.springer.com/article/10.1007/s11416-016-0283-1

[21] Celik, Z. Berkay et al. “Malware traffic detection using tamper resistant features.” MILCOM 2015 - 2015 IEEE Military Communications Conference (2015): 330-335.Address available:https://ieeexplore.ieee.org/abstract/document/7357464

[22] Prasse, Paul et al. “Malware Detection by Analysing Network Traffic with Neural Networks.” 2017 IEEE Security and Privacy Workshops (SPW) (2017): 205-210. Address available: https://ieeexplore.ieee.org/abstract/document/8227308

[23]Ahmed, Irfan and Kyung-suk Lhee. “Classification of packet contents for malware detection.” Journal in Computer Virology 7 (2011): 279-295. Address available: https://www.researchgate.net/publication/225106006_Classification_of_packet_contents_for_malware_detection

[24] Burguera, Iker et al. “Crowdroid: behaviour-based malware detection system for Android.” SPSM@CCS (2011). Address available:https://www.ida.liu.se/labs/rtslab/publications/2011/spsm11-burguera.pdf

[25] Kolosnjaji, Bojan et al. “Adversarial Malware Binaries: Evading Deep Learning for Malware Detection in Executables.” 2018 26th European Signal Processing Conference (EUSIPCO) (2018): 533-537. Address available:https://arxiv.org/pdf/1803.04173.pdf

[26] Karbab, ElMouatez Billah et al. “Android Malware Detection using Deep Learning on API Method Sequences.” ArXiv abs/1712.08996 (2017): n. pag.Address available: https://arxiv.org/pdf/1712.08996.pdf

[27]Liangboonprakong, Chatchai and Ohm Sornil. “Classification of malware families based on N-grams sequential pattern features.” *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)* (2013): 777-782. Address Available: https://pdfs.semanticscholar.org/be3b/d545fba1cf243e9b06a9b0e906d431841cae.pdf

[28] Tahir, Rabia. “A Study on Malware and Malware Detection Techniques.” (2018). Address available:http://www.mecs-press.org/ijeme/ijeme-v8-n2/IJEME-V8-N2-3.pdf

[29] Bhojani, Nirav. “Malware Analysis”. Address available: https://www.researchgate.net/publication/267777154_Malware_Analysis

[30] C. Kessler, Gary. “An Overview of TCP/IP Protocols and the Internet”. Address available: https://www.garykessler.net/library/tcpip.html#intro

[31] Beasley, Jeffrey.S. ”Networking”. Address available:https://sovannarith.files.wordpress.com/2012/07/prentice-hall-networking-2nd-edition-sep-2008.pdf

[32]“TCP/IP Terminology”. Address available:

https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/network/tcpip_terms.html

[33] Hang, Li. “Statistical Learning Method”. [Not available address].

[34]“Introduction of Some Common Network Traffic’’. Address available:

http://www.west999.com/info/html/caozuoxitong/WinXP/20190418/4634571.html

[35] XiaoYue, Zhang. “Malware Review”. Address Available: https://blog.csdn.net/vivid_moon/article/details/80498563

[36] M. Christodorescu, and S. Jha, “Static Anlaysis of Executables to Detect Malicious Patterns”, In Proceeding of the 12th USENIX Security Symp Security 2003, pp.169–186.

[37] Rieck, Konrad et al. “Learning and Classification of Malware Behavior.” DIMVA (2008).

[38] Idika N, Mathur A P. A survey of malware detection techniques[J]. Purdue University, 2007.