Detection of Ransomware using Machine Learning

Thesis

Submitted to the

G.B. Pant University of Agriculture & Technology, Pantnagar-263 145, Uttarakhand, India

By

Shivam Kumar Pujari
ID. No. 54104

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

(INFORMATION TECHNOLOGY)

February, 2021

ACKNOWLEDGEMENT

I am overwhelmed with joy to evince my profound sense of reverence and gratitude to Dr H L Mandoria Professor & Head, Information Technology Department, College of Technology and chairman of my Advisory Committee, for his insightful, critical criticism and invaluable guidance during the course of present investigation.

I am immensely indebted and owe my due regard to Mr Rajesh Singh, Assistant Professor and Dr. Ratnesh Shrivastav, the members of my advisory committee for their persistent encouragement and support. I am thankful to Dr. Alaknanda Ashok, Dean, College of Technology & Dr. Kiran P. Raverkar, Dean, Post Graduate Studies. I have immense pleasure to thank all the faculty and staff members of the Information Technology Department.

Last but not least, I record my sincere thanks from the core of my heart to all the well-wishers whose blessings propelled me to achieve my dreams and I ever remain thankful to all those who could not find separate names but had directly or indirectly helped me.

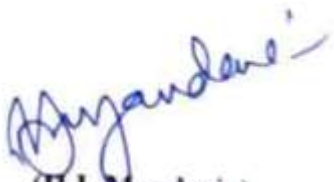
Shivam Pujari

Pantnagar *(Shivam Kumar Pujari)*
February, 2021 *Author*

CERTIFICATE-I

This is to certify that the thesis entitled "**Detection of Ransomware using Machine Learning**" submitted in partial fulfillment of the requirements for the degree of **Master of Technology** with major in **Information Technology** of the College of Post Graduate Studies, G. B. Pant University of Agriculture & Technology, Pantnagar, is a record of bonafide research carried out by **Mr. Shivam Kumar Pujari**, Id No.54104 under my supervision and no part of the thesis has been submitted for any other degree or diploma.

The assistance and help received during the course of this investigation have been acknowledged.



(H L Mandoria)
Chairman
Advisory Committee

Pantnagar ,
Febuary 2021

CERTIFICATE-II

We, the undersigned, members of Advisory Committee of **Mr Shivam Kumar Pujari**, Id. No. 54104, a candidate for the degree of **Master of Technology** with major in **Information Technology**, agree that the thesis entitled "**Detection of Ransomware using Machine Learning**" may be submitted in partial fulfillment of the requirements for the degree.

(H. L. Mandoria)
Chairman
Advisory Committee

30/9/21
(Rajesh Singh)
Member

(R. P. Srivastava)
Member

CONTENTS

a) List of Tables

b) List of Figures

c) List of Abbreviations

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviations	Extended/full form
RW	Ransomware
TPR	True Positive Rate
FPR	False Positive Rate
MCC	Matthews Correlation Coefficient
ROC	Receiver Operating Characteristic
WWW	World Wide Web
i.e.,	that is
PR	Public Relation
API	Application Program Interface
CTB	Curve Tor Bitcoin

Introduction

CHAPTER 1 INTRODUCTION

This chapter describes basics related to ransomware. This chapter has introduced Ransomware with its whole activities including types of ransomware attacks, lifecycle of ransomware, types of ransomware analysis, detection techniques of ransomware and all necessary details about ransomware. After detailed introduction, this chapter described the need of machine learning algorithm for security with ransomware.

1.1 Ransomware:

Ransomware is a malicious program intended to trick or compel a malicious application which forces paying a ransom to the recipient. Ransomware is designed to encrypt essential resources of the computer then ask the customer to pay money to restore those resources.

Ransomware is made of two words Ransom and ware where Ransom means money and ware is for Malware. So, this word ransomware means a malware which is designed for demands money. Ransomware uses encryption techniques to corrupt the data so that some important file or whole operating system would corrupt. Only attackers know the decryption key so that corrupted file may be converted into original files. Attacker demands the Ransom means money for sharing decryption key to the exploited user.

This is very famous nowadays because it deals with money. Attackers demands money in digital currencies such as Bitcoin to avoid any tracking by police or cyber security team. So, ransomware is very trending towards hackers for making money with no security issue.

Ransomware attacks are typically carried out using Trojan disguised as a legitimate file that the user is tricked into downloading or opening when it arrives as an email attachment. However, one high-profile example, the wannacry traveled automatically between computers without user interaction.

1.2 Types of Ransomware Attacks:

It is mainly divided into following types:

1.2.1 Locky Ransomware:

It locks the entire system or operating system to operating. Locky ransomware blocks signing into the device by the victim thereof. The device will, however, normally

be restored by Reboot process or run in Safe Mode. Therefore, this Ransomware is less dangerous and can be patched Very simply, quite quickly.

1.2.2 Crypto Ransomware:

It encrypts some important files of the user. Crypto ransomware encrypts types of files that are for the individual is considered important, such as records, Spreadsheets, databases and images. It is willing to recruit Symmetrical encoding, asymmetrical or combination. The encryption method, depending on the steps involved, we should categorize it into three: class A in which the file is encrypted, but not renamed or relocated; class B in which file is encrypted and renamed, but not moved; and the file is class C, Encrypted, renamed and transferred, rising the task to monitor and restore the file.

1.2.3 Wannacry:

Wannacry is the most widely known ransomware variant across the globe. WannaCry has infected nearly 125,000 organizations in over 150 countries. Some of the alternative names given to the WannaCry ransomware are WCry or Wanacrypt.

1.2.4 Bad Rabbit:

Bad Rabbit is another strain of ransomware which has infected organizations across Russia and Eastern Europe. It usually spreads through a fake Adobe Flash update on compromised websites.

1.2.5 Cerber:

Cerber is another ransomware variant which targets cloud-based Office 365 users. Millions of Office 365 users have fallen prey to an elaborate phishing campaign carried out by the Cerber ransomware.

1.2.6 Crysis:

Crysis is a special type of ransomware which encrypts files on fixed drives, removable drives, and network drives. It spreads through malicious email attachments with double-file extension. It uses strong encryption algorithms making it difficult to decrypt within a fair amount of time.

1.2.7 Cryptowall:

Cryptowall is an advanced form of Crypto-locker ransomware. It came into existence since early 2014 after the downfall of the original Crypto-locker variant. Today, there are multiple variants of Cryptowall in existence. It includes Crypto-defense, Crypto-bit, Crypto-wall 2.0, and Crypto-wall 3.0.

1.2.8 Golden Eye:

Golden Eye is similar to the infamous Petya ransomware. It spreads through a massive social engineering campaign that targets human resources departments. When a user downloads a golden eye infected file, it silently launches a macro which encrypts files on the victim's computer.

1.2.9 Jigsaw:

Jigsaw is one of the most destructive types of ransomware which encrypts and progressively deletes the encrypted files until a ransom is paid. It starts deleting the files one after the other on an hourly basis until the 72-hour mark- when all the remaining files are deleted.

1.3 Life cycle of Ransomware:

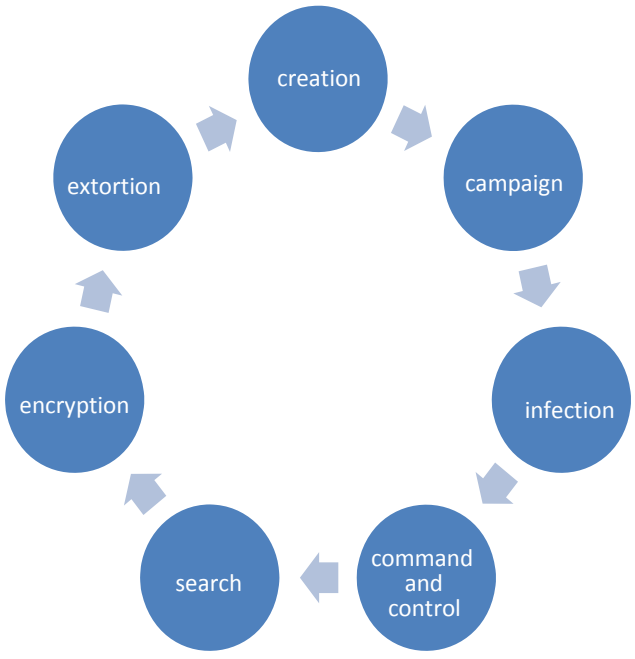


Figure 1.1 Life Cycle of Ransomware

1.3.1 Creation:

Ransomware is created by the programming code by hackers. At the end of each step, the creation stage also includes updating codes to maximize the effect of ransomware.

1.3.2 Campaign:

Distributing the ransomware to the other system to make the victim is the main task of this stage. There are two categories of target victims: individual and institutional victims. The dissemination objective of an individual victim is to reach many other victims as immediately as possible and is very simple as not more computer- savvy are victims. However, for institutionalist victim, the ransomware needs to be highly sophisticated and should be specifically targeted. This is because, normally, there is already some sort of security protection in place. Email connections, email attachments, social media, and infected websites are some of the prevalent infection vectors. The effectiveness of the campaign relies successfully on leveraging the human psychology of anxiety and insatiability.

1.3.3 Infection:

The ransomware setup behavior begins when payload reaches victim's system.

1.3.4 Command and Control:

Once the configuration is complete, for many reasons, the ransomware will contact the Command and Control center, the most important being to acquire the encryption key for the encryption process. Another potential explanation is to download more files with infections that are more advanced.

1.3.5 Search:

The ransomware will start looking for seemingly useful files such as `text messages, spreadsheets, slides, photographs, and databases until the encryption key is acquired.

1.3.6 Encryption:

If all the valuable file types were identified, then ransomware encryption process starts. Normally, use of three type of ransomware encryption technology were used: hybrid encryption, asymmetrical encryption, and symmetrical encryption.

1.3.7 Extortion:

The final move is to show the ransom demand in the cycle after the completion of the above steps. The victim will be alerted of the infection by the demand notice for the payment mode specification. In addition, the notice can also include the deadline for ransom payment, after which the ransomware may proceed to erase the valuable files.

1.4 Types of Ransomware Analysis:

Based on Ransomware analysis, defensive steps can be taken to prevent future attacks. Two types of analysis can be performed: dynamic analysis and static analysis.

1.4.1 Static Analysis:

It is based on the code source of the executable file. For dynamic analysis, the ransomware is carried in moderate environment, and all its actions are recorded for analysis.

(i) Pros: The malicious code is analyzed easily and rapidly. Effective identification also guarantees that the ransomware can be stopped without any risk of being executed.

(ii) Cons: It is vulnerable to code obfuscation. A mismatch with previously defined malicious codes may result from the simple insertion of regular operation codes. Even, when the code is encrypted, the analysis is not effective. Using brute force, there is no effective way to decrypt encryption. It is simply time intensive. Often, static analysis is not efficient for multi-phase attacks. The initial code may be a clear method for opening a backdoor for downloading additional codes and thus may not have a similar malicious behaviour.

Some techniques are used by static analysis are mentioned below:

I. Inspection of file format: file metadata may include information that is useful. Windows PE (portable executable) files, for instance, will have a lot of Details on time of compilation, functions imported and exported, etc.

II. String Extraction: This refers to the output examination of the software (e.g., status or error messages) and knowledge about the assumed ransomware process information.

III. Fingerprinting: This entails computing cryptographic hashes, discovering Ecological objects, such as hardcoded usernames, filenames, Strings for registry.

IV. AV scanning: more likely if the examined file is a well-known malware, It will be observed by all anti-virus scanners. While this recognition process may seem to

be unrelated, it is often used by AV vendors or sandboxes to "confirm" their results.

V. Disassembly: this corresponds to reversing the code of the computer to assembly Language and implied rationale and intentions for the program. This is the one with the most Popular and effective static analysis method.

1.4.2 Dynamic Analysis:

Dynamic analysis is also called behavioral-based analysis. In a managed and supervised environment, normally a sandbox, malicious code is executed. For study, all actions are documented. The actual behavior of malware is observed in this process,

Searching for signs of malicious activity during its execution: changing the host Data, registry keys, suspicious links set-up. - of these, by itself, Action may not be a fair indicator of malware, but it can increase the combination The extent of file suspiciousness. There is a degree of threshold of Established suspiciousness, and any malware that reaches this amount raises an alarm.

The level of accuracy of heuristics-based detection depends heavily on the Implementing. The best ones use the simulated world, such as the sandbox, In order to run the file and watch its actions. While this approach is more time intensive because the file is reviewed before it is fully executed, it is much simpler.

(i) **Pros:** This mode of study is less susceptible to obfuscation, and It is possible to analyses encrypted code. Malicious action must be carried out, a component of the mechanism to attain its goal. It is essential to decode encrypted code until the malware will conduct the action.

(ii) **Cons**: It is both expensive and time-consuming to set up this method of analysis. It is crucial that the configuration of the environment closely imitates a real environment to correctly capture the ransomware actions. As mentioned previously, one of ransomware's setup behaviors is environment mapping. The ransomware will find this and prohibit itself from showing all its actions if the research is conducted on a virtual computer, which can cut costs and resources.

1.5 Ransomware Detection Techniques:

The different identification methods used to discover and classify ransomware are discussed in this section.

1.5.1 Machine Learning:

To construct a model, ML requires learning the trends in data. When supplied with new data, this model forecasts the result.

(i) **Pros**: With sufficient training results, it can predict the result accurately. Because ML requires studying the pattern in the data, which is less vulnerable to obfuscation.

(ii) **Cons:** It is also not easy to find the best algorithm and may require some trial-and-error runs. In addition, if sufficient caution is not taken, biases and overfitting can occur.

1.5.2 Honeypot:

Honeypot involves setting up decoy files for the ransomware to attack. The ransomware can be found after such files are downloaded.

(i) **Pros**: It is necessary to set up traps or honeypot files and then they only wait to be targeted. Therefore, not much maintenance or processing power from the device is needed for the procedure.

(ii) **Cons:** There is no certainty that ransomware attack the honeypot archives. Therefore, understanding the features of files that the ransomware is going to target is critical.

1.5.3 Statistics:

To better understand significant functionality of Ransomware, statistics are used to evaluate ransomware. However, the challenge is to deploy this tool as a detection system.

1.6 Types of Machine Learning:

Machine Learning is a data analysis method that builds an analytical model using algorithms that learn from data and find interesting patterns from the data without being programmed on how to analzse the data. Machine Learning algorithms learn from Experience (E) with respect to some class of Task (T) and Performance measure (P).

Machine Learning algorithms have three phases:

- Training
- Validation
- Testing

Machine Learning approaches are classified into three categories:

1.6.1 Supervised Learning:

Supervised learning is a kind of machine learning approach that makes use of a known dataset for training/learning and build classifier model which is later used for predicting class labels. The training data includes input features (X) and output class labels (Y). Using the data provided for training a supervised learning algorithm builds a model that can predict output class labels (Y) for a new dataset (testing data) which is used to evaluate the accuracy of the model. Examples of supervised machine learning algorithms are: Decision Tree classifier, Support Vector Machine, Random Forest ensembles classifier etc.

1.6.2 Unsupervised Learning:

Most Machine Learning systems learn from labelled instances, nevertheless it is also possible to learn from unlabeled objects but it difficult to do so. Such an approach is called unsupervised learning. The most popular approach of generalizing unlabelled instances is conceptual clustering, where clustering is the task of grouping a set of objects on the basis of similarity of the objects. Examples of unsupervised machine learning algorithms are: K-means, Hierarchical clustering etc.

1.6.3 Reinforcement Learning:

Machine is trained to make decisive actions. The machine is exposed to an environment where it trains itself indefinitely using trial and error, and learns which actions yield the best rewards. Examples of Reinforcement Learning machine algorithms are: Greedy optimization algorithm and LTV (lifetime values) optimization algorithms.

Antivirus uses signature-based approaches in which antivirus analyses malware behavior and detect some common pattern of malware. Antivirus use these patterns to detect any threats. However, this process is unable to detect polymorphic malware, that has an ability to change its signatures, as well as new malware, for which signatures have not been created yet. As we know Ransomware is type of polymorphic malware and it misguide existing signature-based detection techniques. Need for the new detection methods is dictated by the high spreading rate of polymorphic viruses.

Machine learning is efficient to detect such polymorphic virus attacks because it learns from its behavior and update its information.

There are various algorithms under machine learnings. For detection by machine learnings, various classifiers are used.

1.7 Problem Description and Motivation:

Ransomware targets classified and private sector hacks in the return of sensitive data and demands for ransom hacked data or encrypted information. Around the country, Ransomware threats cost millions and millions for companies Millions of Currency. Recently, WannaCry attacked for money as they called for money, Ransom in exchange for decryption keys, but Petya later Any data encrypted and removed during attacks, problems happen:

1. Loss in productivity in downtime: 50 percent
2. Production of corporate income per hour: $24,000
3. 21 hours of inactivity before full recovery

Effects of Ransomware Attacks on Organizations:

1. Loss in employee productivity because their work can be destroyed by attackers.
2. Many companies are temporarily going to shut down now.
3. Risk of corporate sales failure.

1.8 Objective:

Following goals of our proposed work:

1. To develop the technique for detection of ransomware with some other malware like Dridex and Vawtrak.
2. To determine the classification of ransomware and polymorphic ransomware.
3. To develop the method for increasing the detection rate of malware.

1.9 Thesis Organization:

Thesis outline is described in various chapter:

Chapter 1 described the overall process of ransomware. It first introduced Ransomware with its whole activities including types, lifecycle, analysis, detection techniques of ransomware and all necessary details about ransomware. After detailed introduction, this chapter described the need of machine learning algorithm for security with ransomware.

Chapter 2 described the various research about the ransomware and other malware attack and cure of this, done by various researchers in previous year. It gives a rough idea about research field, necessary tools required for implementing the methods of Ransomware Detection.

Chapter 3 described necessary methods and tool which we used in our work. It gives the brief knowledge about Ransomware and malware sample which we used to test. It also talks about cuckoo sandbox which we use for feature selection required for implementing machine learning classifier.

Chapter 4 described about the result of our experiment. It also compares between all four results obtain and conclude that Random Forest Algorithm is best.

Chapter 5 outline the conclusion and brief overview of work done. At the end it discloses the future work and improvement for further research.

Review of Literature

CHAPTER 2 REVIEW OF LITERATURE

This chapter describes study of the research done in previous year. Well-grounded sources such as IEEE and other journals on Ransomware are taken into consideration in order to gain relevant information, which helped us in answering the research questions. There is massive collection of research papers on Ransomware, detection and classification, feature extraction, machine learning and other retrieval techniques. In this chapter ransomware detection using machine learnings have been studied. After studying them, following papers are found relevant for our present work:

2.1 Signature Based Approaches:##

A signature-based approaches typically monitors inbound network traffic to find sequences and patterns that match a particular attack signature. These may be found within network packet headers as well as in sequences of data that match known malware or other malicious patterns. An attack signature can also be found within destination or source network addresses as well as in specific sequences of data or series of packets. Some approaches are discussed below:

Shankarapani *et al.* **(2010)** present algorithms for identification that can enable the antivirus community to guarantee a version of a known malware without having to establish a signature, it can always be detected. By study of comparisons (based on specific quantitative measures) A matrix of similarity scores that can be generated, is performed to determine the likelihood that a piece of code under inspection contains a particular malware. Authors present two methods- SAVE and MEDiC.

MEDiC uses analysis assembly calls and SAVE uses API calls for analysis (Static API call series and Static API call set). Authors illustrate where assembly can be superior to API calls. This provides a more rigorous comparison of executables. On the other hand, API calls may be superior to Assembly for Its speed and smaller signature. A better detection efficiency can be given by both of proposed techniques against obfuscated malware.

Alazeb *et al.* **(2011)** Zero-day Identification of Malware based on Supervised Learning Algorithms the API functions were used for feature representation, again and again. With the Help Vector Machines algorithm, the best result was obtained with normalized polykernel. 97.6 percent accuracy was reached, with a false-positive rate of 0.025.

Zhao *et al.* (**2011)** concentrate on the actions of software dependent structure for the identification of malware, dubbed AntiMalDroid, by utilizing algorithm SVM. proposed structure dynamically extends malware characteristics into the database.

N Andronio *et al.* **(2015)** proposed the HelDroid system that Detects a ransomware class that is intended to hack Platforms for Android. The HelDroid framework utilizes code Characteristics, including manifests of the program and call Functions for using ransomware and its family class to classify Processing natural Language (NLP). The HelDroid phase is Educated to recognize popular messages appearing in the ransomware code to identify it.

Amin Kharraz *et. al.* **(2015**) proposed" A Look Under the Hood of Ransomware Attacks" studied ransomware attacks between 2006 and 2014. It tells that we can detect and stop zero-day ransomware attacks by keep view of I/O requests and securing the MFT (Master File Table) in the NTFS. The authors suggest mitigating ransomware attacks, system need real time monitoring.

Nolen Scaife *et al.* **(2016)** proposed "Cryptolock (drop it): Stopping Ransomware attacks on user data" talks about crytodrop that warn user during suspicious file activity. Indicator monitor the real time change of user data to track the suspicious process for reputation scores used to alert the user and suspend it.

Sgandurra *et al.* **(2016**) have suggested EldeRan tool, which checks Characteristic signatures of ransomware by examining a collection of Actions in the initial phases of the kill-chain assault flow. EldeRan detects and categorizes Ransomware dynamically by evaluating tasks such as registry operations Key operations, Windows API calls, directories, and files Operations of a machine. Logical Regression by EldeRanuses to identify each user's classifier algorithm and ML algorithm, Application, which has additional features for defining and identifying for as yet unknown ransomware, build signatures.

Brewer et al. **(2016)** analyzed ransomware attacks on business organisations and suggested that five separate stages of ransomware attacks be understood, including: exploitation.

Infection and infection, distribution and execution, backup spoliation, file encryption and user warning to recognize the vulnerability indication (IOC) to direct the creation or mitigation of a security.

Vinayakumar R *et al.* **(2017)** proposed "Evaluating Shallow and Deep Networks for Ransomware Detection and Classification" talks about supervised machine learning method of detection of ransomware. Multi-layer perceptron (MLP) used for ransomware detection and classification. It proposed a method using API calls for ransomware detection. As a feature for classification, it uses 131 API calls which is input of MLP architecture.

Rhode *et al.* **(2017)** presented a novel approach to obtain high accuracy. In the first 20 seconds, the proposed algorithm detects ransomware files during the execution stage. VirusTotal and VirusShare have collected the dataset from it. 23,145 benign and 2,286 malicious records are present in the dataset. To convert all alphabetic values into numerical ranges for the presentation of RW, a preprocess was performed. For prediction of ransomware recurrent neural networks (RNNs) were implemented. The precision is ninety percent in five seconds and ninety six percent in ten seconds. The minimum false negative rate (FNR) was 4.5 percent and only 3 percent was FPR. The real value of the simulator in twenty seconds is ninety-three in percent. The experiment was in Python 2.7 with use of Keras for implementation of the RNN model.

Carlin et al**. (2017)** highlighted the low-level study of both Dynamic and static opcodes to detect malware on the1,000 samples of labels in the runtime dataset to influence the typical AV lebels. They obtained the dataset from VirusShare. The reviewer chose the scale and facility modality. There are 180,000 malware records, and all records are called by message digest MD5 hash with no other metadata.

Data is going to be just 1,000 opcodes with a 1.0 percent margin are preprocessed.764 are benign and 18,827 malicious sample in the dataset. About samples. In WEKA version 3.8, the counterbased classifier uses RF and implements it. The RF's highest accuracy is 98.4% percent.

Takeuchi *et al.* **(2018)** introduced Ransomware Detection using Support Vector Machine (SVMs). There are 588 samples in the dataset, which have 312 benign and 276 RW, which were obtained from VirusTotal. The authors build the same vector symbols with different sequences of API calls. The author checked and educated the classifier of the SVM data type. The normal vector symbol accuracy is 93.52 percent, and 97.48 percent is the best SVM accuracy.

Chen *et al.* **(2018),** introduced RansomProber, a strictly dynamic detector of ransomware, which uses a series of guidelines to control various facets of the implementation of the app, such as the involvement of encryption or anomalous structures

for form. The results obtained suggest a very high accuracy, but the machine was not released publicly not (to the best of our knowledge).

Greg *et al.* **(2018)** recommend technique of network management for data from traffic so that features can be extracted from it. Those features are used in the classification of ransomware and the used algorithm is Random Forest Binary Classifier. They say the rate of detection is 86 percent.

Omar M. K. Alhawi *et al.* **(2018)** launched NetConverse, a machine learning assessment for reliable identification of Ransomware Network Traffic for Windows. They achieved a True Positive Rate of 97.1 percentage using a dataset created from conversation-based network traffic features with the J48 decision tree and 96.8 percentage detection rate accuracy with the LMT classifier.

2.2 Behavior Based Approaches:

Behavior based approaches applies Statistical, AI and machine learning to analyze giant amounts of data and network traffic and pinpoint anomalies.

Instead of searching for patterns linked to specific types of attacks, behavior-based solutions monitor behaviors that may be linked to attacks, increasing the likelihood of identifying and mitigating a malicious action before the network is compromised.

Manish Shukla *et al.* **(2016)** proposed "Viritualized environment for mitigating ransomware threats". Using dynamic analysis of system in term of I/O process, by the differences between normal behavior of uncompromised machine and abnormal behavior when machine has been infected.

Song *et al.* **(2016)** suggested a strategy for detecting and preventing Changed malware from Android platforms attacks. The system proposed has a very high and quick rate of detection. Since the tool is designed to be integrated within the instrument android source code Instead of becoming an external mobile Android source code. This makes it a really good strategy, as it may found the ransomware and its variations even though it does not have by surveillance, the signature template. The processor, I/O rates and memory usage is its key for unwanted behaviors conducts. If any discrepancy is found, to interrupt the operation, this method takes urgent action and uninstall the process, associated with the use of process memory and I/O rates is to detect irregular behaviors.

Shaukat *et al.* **(2018)** proposed a Layered tool designed to defend against crypto ransomware after a massive ransomware dataset was analyzed. Dynamic and static analysis, method is mixed by Ransom Wall for developing a distinctive compact package that detects Ransomware habits of behavior. Also, Ransom Wall is Built to detect zero-day vulnerabilities of ransomware and it's The Powerful Trap Layer feature can classify ransomware attacks at the initial level of kill-chain. It is achieved by the identification of suspicious activity in initial layers, and the classification of any unusual activity. If any files are discovered to have been changed by the rogue process, the files are backed up before the rogue process is confirmed to be either unsafe or not, to protect the user's records.

Krzysztof Cabaj *et al.* **(2018)** proposed "Software-Defined Networking-based Crypto Ransomware Detection Using HTTP Traffic Characteristics" Stopping Ransomware attacks on user data. Author tells in this paper that cryptowall and locky ransomware can be detected through http message sequence and the size of content. It used SDN because it provides more security in effective and flexible manner to the current network. we can arrange network in logically centralized way using SDN. Purposed work divided into three phases-Learning phase, Fine-tuning phase and Detection phase.

In learning phase ransomware were executed in the Maltester environment so that all outcomes would be in control for analysis and all traffic generated by the tested machine which is infected by ransomware was kept secure for further preprocessing.

In fine turing phase centroid vector of relative feature vectors, minimal and maximal Euclidean distances for all vectors were calculated. Information related to minimal and maximal distance to the centroid was used. As a result, all the required parameters used later in the detection phase calculated.

In last level, a record consisting of generated HTTP message sizes was created for every HTTP server listed. Thus, every new HTTP response is inserted at end of that record related to the domain or Internet Protocol address of that website. After updating this utility in the record, if size of utility was three or greater than three implies the potential ransomware was detected.

With 1-2% or 4-5% false positive, it was able to achieve detection rates of 97-98% when relaying triples, respectively, on domains or POST.

Michele Scalas *et al*. **(2019)** proposed R-PackDroid that can accurately detect novel samples in the wild and showed resilience against static obfuscation attempts.

Author recommend learning based detection strategies that depends on knowledge from the device API. Leveraging these methods, the fact is that ransomware attacks are extensively used to execute their behavior by the device API, and It enables generic malware and ransomware to be distinguished.

Author checked three different ways of using information from the Framework API, i.e., by packages, classes, and methods, and compared their output with other, more difficult one State-of-the-art techniques, proposed model obtained suggested that programs based on the method, Ransomware and generic malware can be found by the API with very good precision, comparable to systems that employ more complicated data.

2.3 Summary:

We have studied Review of Literature depicted that researcher have contributed a lot in the field of Ransomware analysis. Firstly, we found that detection techniques can be divided into signature based and behavior-based methods. Further we found that Ransomware is polymorphic malware which changes its signature with time to manipulate operating system. We conclude that signature-based method cannot detect ransomware accurately, so we use pattern of signature to detect ransomware and some other malware also. This research is focused on detection of Ransomware and similar type polymorphic Malware also. The malware we use for detection is-Dridex Locky, Teslacrypt, CTB-Locker, Vawtark, Zeus, DarkComet, Xtreme and Cyber Gate.

Materials and Methods

CHAPTER 3 MATERIALS AND METHODS

In proposed framework we want to detect ransomware with other malware like dridex, teslacrypt, vawtrak, zeus, darkcomet. This chapter gives the details about the materials required to implement the proposed work. Also, it gives the information about the tools and methods used for detection. This chapter also includes the research methodology and research analysis done for our proposed work. Ransomware type malicious activity is a very broad area for research, but the proposed work focuses on the area of providing better accuracy in detection of ransomware and their classification.

3.1 Materials:

This section deals with the brief introduction of the hardware and software tools, which has been used in the proposed work.

3.1.1 Hardware Used:

All of the proposed work mainly held around NetFlow cyber threat classification, so that only a single computer system as hardware tool is needed for realization of the proposed work. The system is developed and executed on a PC, which has the following hardware specifications:

- Processor: Intel(R)Core™ i3, 2.4200 GHz
- Installed memory (RAM): 4 GB
- System Type: Windows10, 64-bit operating system
- Hard disk: 1 TB

3.1.2 Software Used:

The entire working scenario performed on cuckoo sandbox and Weka 3.6.10 classifier tool.

3.1.2.1 CUCKOO Sandbox:

The study is based on and targeted to Cuckoo Sandbox. It is clear that to apply the machine learning algorithms to any problem, it is essential to represent the data in some form. For this purpose, Cuckoo Sandbox was used. The reports generated by the sandbox, describing the behavioral data of each sample, were preprocessed, and malware features were extracted from there. Cuckoo Sandbox is the open-source malware analysis tool that allows getting the detailed behavioral report of any file or URL in a matter of seconds.

Cuckoo has a scalable architecture that is extremely versatile and can be used Both as a standalone app and merged into the broader structures or framework. A host computer (the host machine) is the key component of the Cuckoo infrastructure (the tools for management) and a variety of guest computers (virtual or physical). Analysis machines). Its operational scenario is very simple: as soon as possible, A virtual world is dynamically presented to the server when the new file is sent. The file is transferred to it, and all the actions done in the system are executed.

3.1.2.2 Weka 3.6.10:

"Waikato Environment fozzr Knowledge Analysis" is abbreviated as Weka. Weka is a machine learning tool which was developed by university of Waikato. The Weka algorithms can either directly apply on the dataset or can be called from your Java code. Weka is written in Java so it is an open source tool and can run any of the platforms.

The major functions of Weka are preprocessing of data, classification, and clustering and association rule mining.

3.1.2.2.1 Weka GUI:

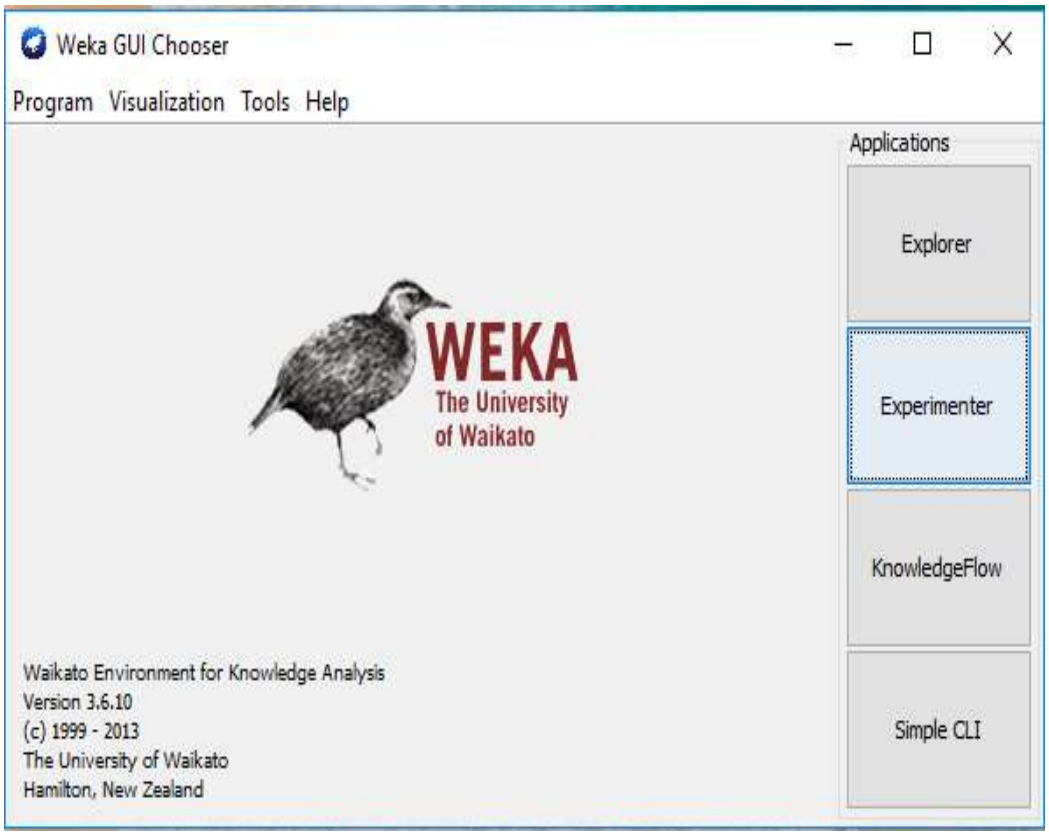


Figure 3.1 Weka GUI Chooser

In figure 3.1 the Weka GUI is shown. It supports: Explorer, Experimenter, Knowledge flow and simple CLI. The explorer supports preprocessing of data, attribute selection, learning and visualization of data. The experimenter facilitates the environment for testing and evaluating the ML algorithms. The KnowledgeFlow tab shows the flow of data.

The user can choose Weka components form the tool bar and place them on a layout canvas. After this, it connects to the components in order to form a KnowledgeFlow for processing and analyzing the data. The last one that is CLI (Command-Line Interpreter) is a simple interface for typing the instructions.

3.1.2.2.2 Weka Explorer:

The Weka explorer opens the data file with the help of Preprocess Button.

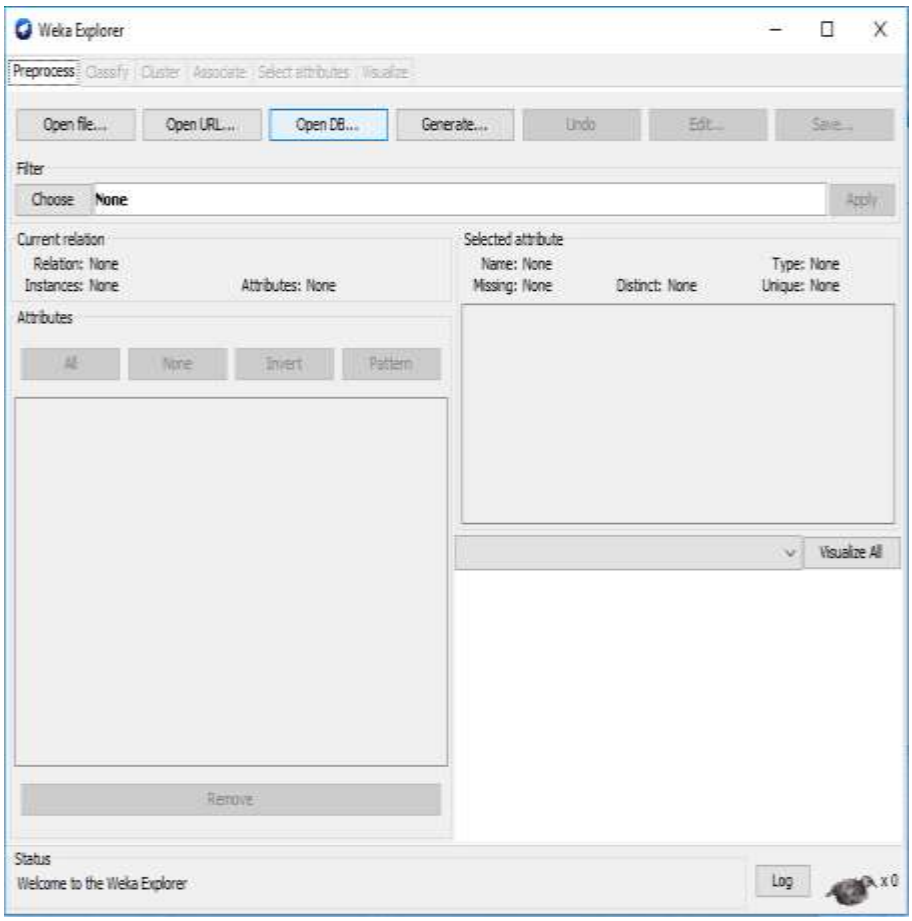


Figure 3.2 Weka Explorer

Figure 3.2 shows view of weka explorer in which we can mine data. We can load tha data with the help of preprocess button. For classification we can use classify button.

3.1.2.2.3 Visualization of Data:

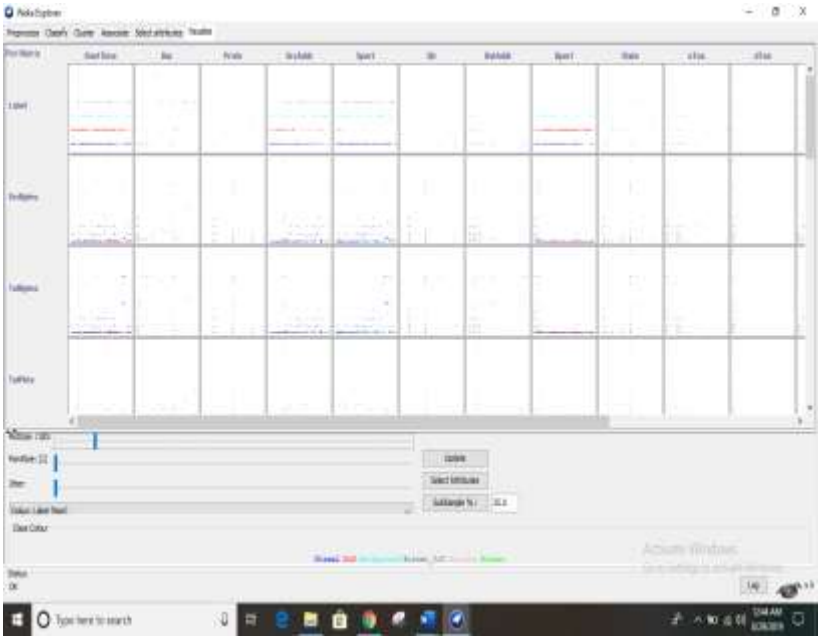


Figure 3.3 Visualization of Data

After opening the data file, visualization panel shows the scattered plot current dataset based on the attribute. The number of cells in the matrix can be reduced by pressing the "Select Attribute" button. Here, the figure 3.3 has taken from the PnatHoney_1 sample file. The relevant classifier is selected from the classifier tab. In our work J48 and Random Forest classifiers are selected.

3.1.2.2.4 Weka Data Format:

Weka can support file formats as given below:

- CSV
- ARFF

Mainly the ARFF (Attribute Relation File Format) file format is supported by Weka. In case the ARFF file is not available then the CSV (Comma Separated Value) file can be converted in ARFF format by Weka tool.

3.1.2.2.5 Preprocessing of Data:

For getting the better results preprocessing of data is must. There are the following ways to inject the data for preprocessing:

- **Open File-** it enables the user to load the data from local machine.
- **Open URL-** enables us to load the data from different locations.
- **Pen Database-** allows us to fetch the data set from database source.

3.1.2.2.6 Classification:

For the prediction of nominal or numeric quantities, we have used the classifiers in Weka. The available learning methodologies are decision-trees, SVM (Support Vector Machine), instance base classifiers, logistic regression.

Before executing any of the classification techniques, we fix the test options. The available test options are given below.

- **Use Training set:** estimation is based on how crisply it can estimate the class of the instances it was trained on.
- **Supplied Training set:** it is an external file that can be used as supplied set.
- **Cross-Validation:** in cross validation, we set 10-folds and among these we fixed 9-folds for training and 1-fold for testing.
- **Split Percentage:** estimation is based on how fine it can predict a percentage of the given data.

3.1.2.2.7 Specialization:

- Weka is specialized in Machine learning applications and it has ML tool which is better for mining association rule.
- It supports large number of ML algorithms like SVM (Support Vector Machine), MLP (Multi Layer Perceptron), LR (Logistic Regression).

3.1.2.2.8 Advantages: Few advantages of Weka are:

- Weka is an appropriate tool for developing new ML schemes.
- Weka supports ARFF, CSV, and J48 binary formats.

3.1.2.2.9 Disadvantages:

Some disadvantages of Weka are:

- Weka has very worse connection with Excel sheets database and non-java based databases.
- Weka is less strong in classical statistics.
- Weka does not save the parameters for future datasets.

3.1.2.2.10 Installation of Weka:

- Go to http://www.cs.waikato.ac.nz/ml/weka
- Click the Download and install button
- Choose which one to download:

i. the “stable” version (not the “developer” version)
ii. the appropriate version for your computer; Windows, Mac OS, or Linux.

3.2 Feature Extraction:

In any of the above instances, we ought to be able to extract the Attributes from the input data such that the algorithm can be fed to it. For instance, In the case of house prices, knowledge may be viewed as a multidimensional Matrix, where an attribute is represented by each column and rows represent the attribute for these properties, numerical values. In the case of the image, it can be data each pixel is interpreted as an RGB color.

These features are referred to as traits, and the matrix is known as Vector of functions. The data extraction method from the files is referred to as the Feature Extraction. The purpose of extracting features is to acquire a collection of insightful and Data that is non-redundant. It is crucial to understand that characteristics can reflect the important and valid details regarding our dataset, because we do not have it. An exact forecast cannot be made. That is why the extraction of features is always a Non-obvious assignment, which involves a lot of study and checking. Additionally, it is Quite domainspecific, but generic approaches apply badly here.

Non-redundancy is another major prerequisite for a good feature set. Getting redundant characteristics, i.e., characteristics that outline the same data as well as redundant attributes of knowledge, which depend closely on each other’s will skew the algorithm and thus have an incorrect one Outcome.

Furthermore, if the input data is too large to be fed into the algorithm (has too many characteristics), so it can be translated to a reduced vector function (vector, having a smaller number of features). The phase of diminishing the measurement of the vector is referred to as function collection. At the completion of this operation, the chosen features are supposed to detail the related data from the Initial set so that, without any precision loss, it can be used instead of initial data.

3.3 Classification Methods:

From the viewpoint of machine learning, the identification of malware can be used as a Classification or cauterization problem: unknown types of malware should be Centered on certain properties, clustered into multiple clusters, defined by the that algorithm. On the other hand, having trained a model on the large dataset. We will

minimize this concern to grouping because of malicious and benevolent files. This issue can be reduced to classification by established malware families, Just Having a small range of classes, one of which is definitely a sample of malware. It is easier to define the right class, because the results will be greater. Accurate than for algorithms of cauterization.

3.4 Dataset:

A total of 1,156 files were obtained for this initiative. Hashes for each of them, Incidence reports or reverse malware is found that mark files uniquely. Reports in engineering, and these hashes were subsequently used to extract the Matching samples from the VirusTotal service with the assistance of external samples Researchers on ransomware. (VirusTotal 2017) To be able to work for a wide range of applications Nine malware families have been included in the dataset, resulting in 1,116 malicious files and 40 archives that are benign.

Malicious families that were used are dridex, locky, teslacrypt, vawtrak, zeus, darkcomet, ctb-locker, cybergate, xtreme. These are discussed in detail below:

3.4.1 Dridex:

We used 115 unique files infected by dridex for analysis This malware belongs to the Trojan class, specifically, banking trojan. It caused a huge infection in 2015, resulting in 3 000 - 5 000 infections per month. Cridex, the malware that circulated in 2012, is the root of Dridex. Also, Cridex was a Stealer of bank credentials, but more specifically, it was a worm that used storage units connected as a spreading vector. A redesigned edition in 2014 It arose, moving from communications of command and control to peer to peer and thereby becoming more resilient to takedown operations.

The Dridex attacked customers of individual banks in order to rob their own banks. It is said that more than 500 institutions are targeted at and 40 regions, primarily concentrating on high-income English-speaking nations, Rates: the bulk of diseases in the United States, the United Kingdom and About Australia.

For spreading dridex, attacker generally uses real organization name as sender address in maximum cases. Figure 3.4 shows how dridex operate.

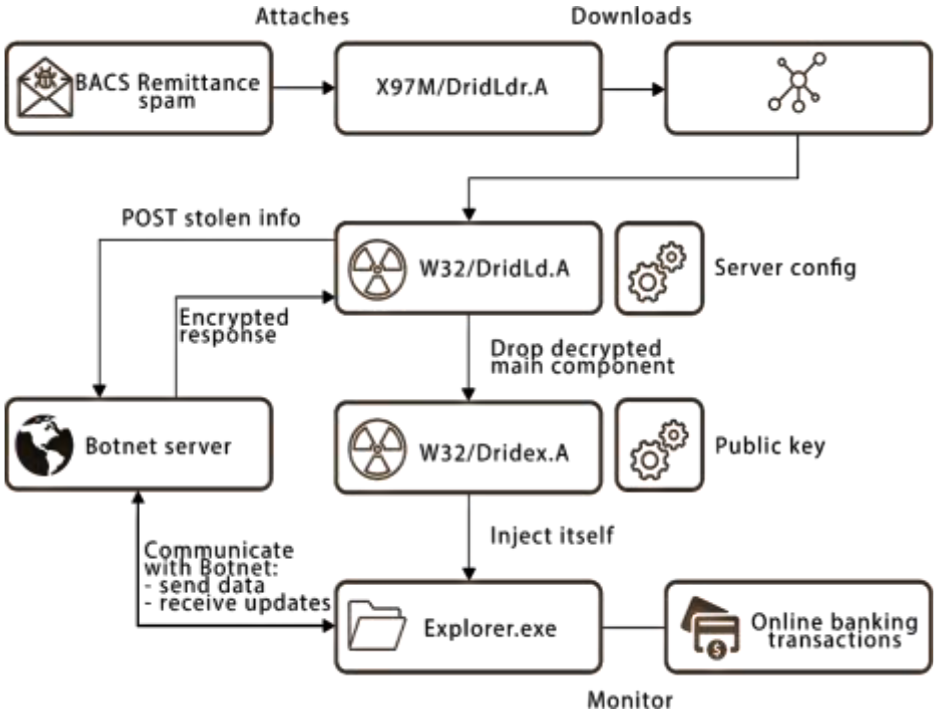


Figure 3.4: Illustration of Dridex Operation

3.4.2 Locky:

We used 115 unique files infected by locky ransomware for analysis. This is the Ransomware which uses RSA-20488 to encrypt all data on the victim's device AES-256 ciphers and adds an extension of. locky. Locky appeared in February 2016 and has since then been aggressively circulated. The Very Most spam campaigns are common delivery vectors, specifically fake invoices and platforms for phishing. This spam campaigns were strongly similar to those of the Used for the sale of Dridex in terms of its scale, the use of financial records and Macros, which means that the Dridex community is responsible for this, about ransomware. The price ranged from 0.5 to 1 bitcoin for decryption of device data.

Figure 3.5 shows how Locky Ransomware process for encryption. It also deletes initial files and shadow copies so that no one can get data back.

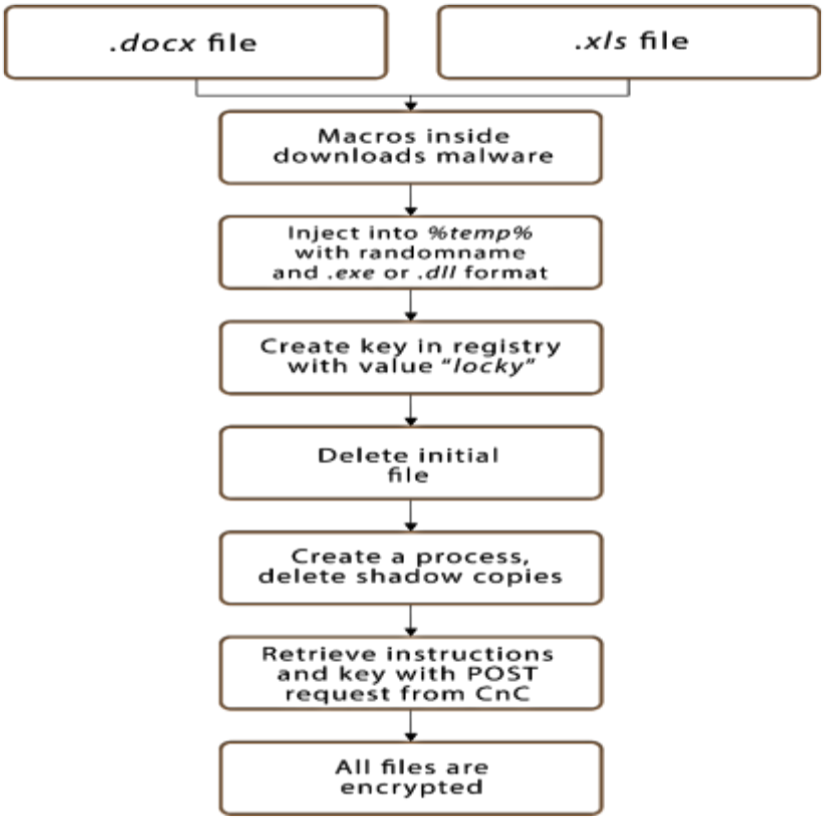


Figure 3.5: Illustration of Locky Ransomware process

3.4.3 Teslacrypyt:

We used 115 unique files infected by teslacrypt ransomware for analysis. The key vector of dissemination is infected portals and websites. Emails with links to malicious websites which once download the malware They're on a visit. Upon installation, the file is automatically executed.

Payment is needed to be made via PayPal or Bitcoin for a decryption key. TeslaCrypt, unlike other ransomware families, obvious data files, such as .pdf, .txt, .gif, etc., encrypted, yet often Files relating to the game, including Call of Duty, World of Tanks, Minecraft and World About Warcraft. Interestingly, the attackers behind TeslaCrypt announced in May 2016 that they the project closed, and the master decryption key was released. Many days more, A free decryption service has been published by ESET Antivirus.

Figure 3.6 shows how email phishing and malicious websites are encrypted using Teslacrypt RW.

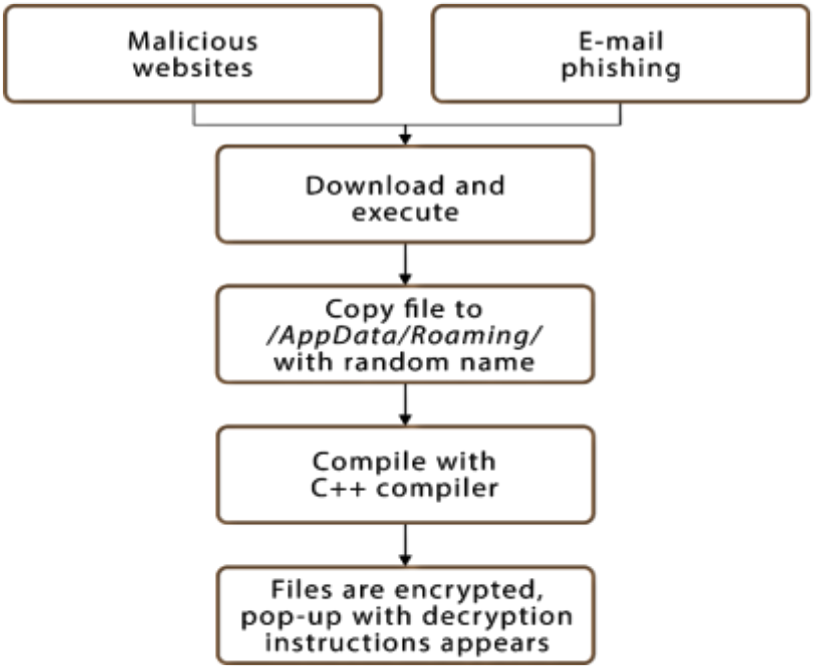


Figure 3.6:Illustration ofTeslacryptprocess for encryption

3.4.4 Vawtrak:

We used 74 unique files infected by vawtrak for analysis. Referred as well vawtrak, as neverquest or snifula, is another example of Trojan banking. The most diseases have arisen in the czech Republic, the United States, the United Kingdom and Germany. Spreading vectors include downloaders of ransomware, spam with malicious connections, or Additional drive-by updates. Vawtrak is worthy of winning after downloading Access to a victim's bank. accounts, as well as compromised passwords, Private keys, passwords, etc.Figure 3.7 shows how Vawtrak infect the system.

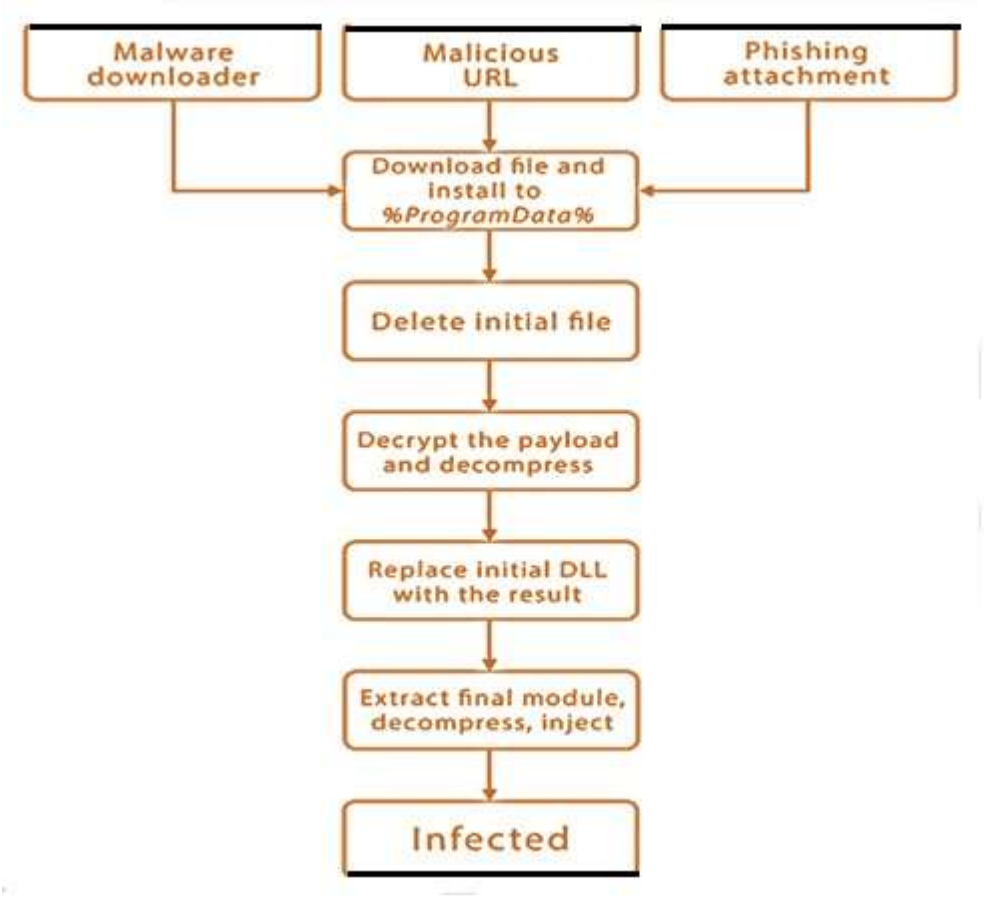


Figure 3.7: Illustration of Vawtrak Operation for infection of system

3.4.5 Zeus:

We used 116 distinct files injected with Zeus for analysis. Yeah, it's a Botnet kit, which can be quickly sold for about 70000 on the black-market USD. USD. Zeus formed since its introduction in 2007 and remains one of the Most famous members of botnet malware. Vector infection of Zeus differs widely, beginning with spam emails and ending with drive-by-drive. Only downloads, the malware injects itself into sdra64.exe after the update. Process and change the values of the register to execute it on the framework initiation. After that, in the winlogon exe phase, Zeus injects itself and closes the original executable format. Zeus' capabilities include hacking of system information, online stealing of system information, credentials, information for storage. The details of the knowledge to be compromised are either Hard coded or retrieved from the command and control in the binary the success of Zeus malware is attributed to the fact that it is fairly inexpensive. Figure 3.8 describes the Zeus operation.

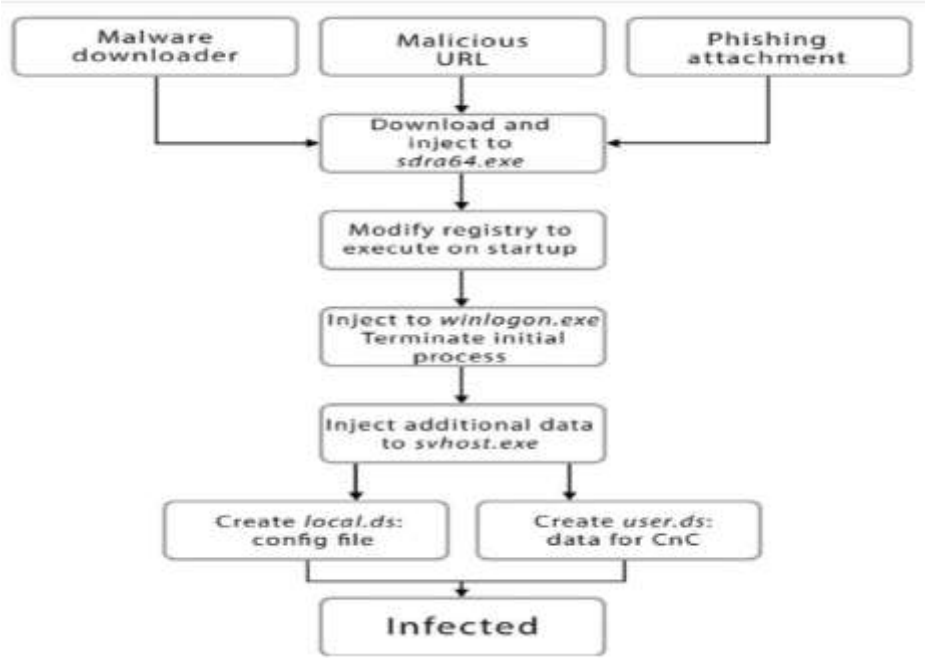


Figure 3.8 : Illustration Zeus Operation for infection of system

3.4.6 DarkComet:

We used 130 distinct files injected with darkcomet for analysis. It is an example of the remote administration method (RAT). That was the Used in 2012-2015 in multiple rapes.

DarkComet was not developed initially, However, it was a malicious instrument because of its design and functionality. The Syrian government eventually used it for spying, followed by others in the intervening years, other threats. It was used by the Syrian government during the Syrian war in 2014 for the purpose of spying on Syrian people who have bypassed official censorship about the Internet. The 'Je Suis Charlie' slogan was used in 2015 to trick people into The darkcomet download: it was masked as an image that compromised when downloaded by consumers.

Figure 3.9 shows communication between server and client where server is attacker and client is infected machine.

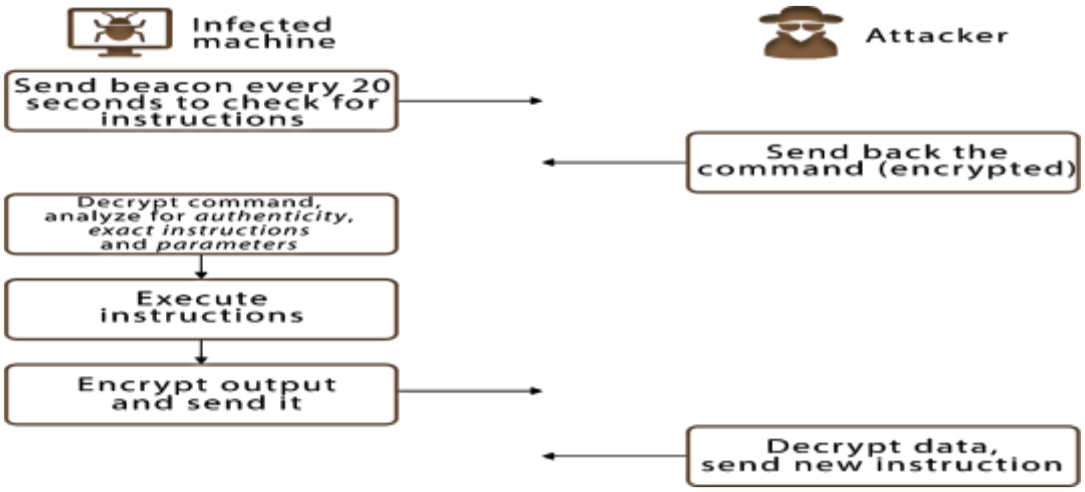


Figure 3.9: Illustration of DarkComet communication scheme

3.4.7 CyberGate:

We used 120 distinct files injected with cybergate for analysis. The remote management platform is another example of cybergate (RAT). Written in Delphi, it is constantly being developed, resulting in stability and extensive functionality. It should be mentioned that CyberGate can be considered “legal” malware since it was initially developed for legal purposes and is used in legal problems. However, it is often used for malicious activity, such as espionage.

3.4.8 Xtreme:

We used 120 distinct files injected with xtreme for analysis. Xtreme is yet another instance of Rodent. Developed in Delphi, it is freely accessible the source code and shares it with many other Delphi RAT malware, including cybergate.

In several government bombings, as well as in several attacks, Xtreme was used With Israel and Palestine threatened. Xtreme's architecture depends on the architecture of the network server, where the attacker is assumed to be a client. THE Settings are written to the percentage of the APPDATA percentage \Microsoft\Windows folder or to the percentage of the APPDATA percentage folder A folder named after the generated mutex. The information is then encrypted with the help of "As a password, RC4 and "Cfg" or "CYBERGATEPASS. The Settings "The extensions are contained in a “. ngo" or ".cfg" format. Data from the setup the configured file name, the injection method, FTP and CnCC are included. Data, name mutex. (2014 for Villeneuve and Bennett).

3.4.9 CTB-Locker:

CTB-Locker was the last malware family that was used in research. We used 78 distinct files injected with ctb-locker for analysis. This is another case of malware encrypting the files of users. Asking for the decryption key for money. The CTB-Locker samples spread by the malicious attachments emails. Attachments were represented by .zip files with an attachment Downloader inside.

Figure 3.10 describes the initial function of the CTB-Locker. Malware reduces itself to the percent temp percent folder with a random name when executed. And the svchost.exe process injects itself. A pop-up screen will appear upon complete completion of the malware, providing Data on payment data and encryption. Spain, France and Austria were mainly attacked by CTB-Locker.

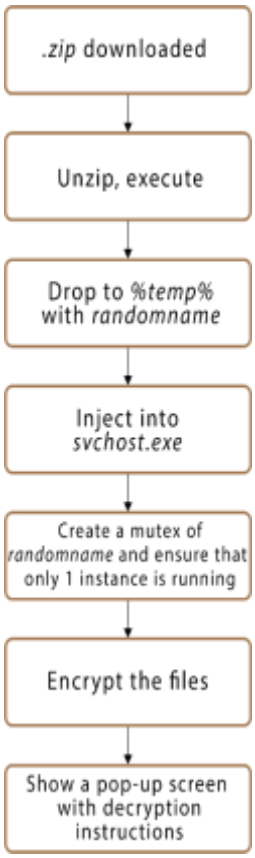


Figure 3.10: Illustration of CTB-Locker operation for encryption

3.5 Reports and Features:

We need to find out What kind of information and how it can be viewed should be extracted for apply machine learning algorithm. Any field study uses string properties or properties of file formats as a base for representation of functions. For Windows-based malware, for example, Samples, data found in PE headers, are also used for analysis as a base. The implementation of format-specific function extraction, however, is not the highest Solution, since evaluated file formats can differ drastically.

3.5.1 API Calls:

API stands for Programming Interface for Applications and refers to the package tools that offer an interface to connect with the various Components for apps. During the execution of the API calls are documented Malware and link to the form in question. They outline it all the operating system, and the operations on the files, Registry, mutexes, protocols and other previously described characteristics.

For API calls, for example, OpenFile, OpenFileEx, CreateFile, CopyFileEx, and so on Specify file processes, OpenMutex, CreateMutex, and OpenMutex calls, and CreateMutexEx explains the opening and development of mutexes, etc.

API Call traces provide a broad overview of the behavior of the sample, including all the above-mentioned estate. Besides that, they have a wide variety of Set of values which are distinct. In addition, they are easy to classify in numeric Format, and this is why they have been picked as functions. Here, the functionality. The collection is specified by the number of specific API calls and the number of returns codes.

3.6 Feature Selection:

It is necessary to delete the not required feature and, thus, in our case, for instance, a certain API call could only be activated once in one sample. In the case of a large and varied Feature collection, no function in the algorithm will be played by this special API call and, Therefore, replacing it would in no way impact precision.

After the features are extracted and interpreted as a mixture matrix, we ended up with 70518 features. For collection and processing, this number is too high for the exact forecasts. With such a large feature set, for example, it takes to load the dataset, preprocess it and run the knearest neighbor's algorithm on an x64 8GB RAM machine for approximately two-three hours. This quantity of the resources is unacceptable, and

irrelevant functionality need to be disabled. Filtering methods are three general types of methods of function filtering, Methods with wrappers and embedded methods.

Results and Discussion

CHAPTER 4 RESULTS AND DISCUSSION

This chapter involves the analysis of results done after simulation. Proposed method is used to upgrade the performance of Ransomware Classification. The results of all the steps of proposed method are shown. Implementation is performed using Weka 3.8.5 simulation tool. Test results of intermediate phase show the accuracy and efficiency of applied algorithms.

This chapter addresses the outcomes of the review of the application of the Methods in machine learning. The result is based on various parameters such as True Positive Rate, False Positive Rate, Precision, Recall, F-Measure, MCC, ROC Area, PR Area. The identification accuracy is calculated as the Percentage of instances listed correctly. Accuracy can be calculated as-

$$Accuracy = count(Correctly\ identified\ samples) / count(Total\ samples);$$

At last, this chapter summarize result in tabular form and shows the comparison between existing work and proposed work with the help of table.

4.1 Naïve Bayes Classifier:

Out of 1156 instances 968 instances are classified accurately while 168 instances classified incorrectly. Hence accuracy is 0. 8373.The result in TP Rate, FPRate, Precision, Recall, F-Measure, MCC, ROCArea, PRCArea and their class of malware or benign is shown below in figure 4.1.

Figure 4.1 shows the confusion matrix and TP Rate, FPRate, Precision, Recall, FMeasure, MCC, ROCArea, PRCArea and their class of malware or benign by Naïve Bayes Classifier.

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.647	0.022	0.836	0.647	0.730	0.696	0.974	0.823	Benign
	0.904	0.028	0.796	0.904	0.846	0.829	0.969	0.852	Dridex
	0.937	0.002	0.978	0.937	0.957	0.953	0.991	0.954	Locky
	0.973	0.003	0.973	0.973	0.973	0.971	0.997	0.974	Teslacrypt
	0.622	0.020	0.676	0.622	0.648	0.626	0.935	0.589	Vawlrak
	0.707	0.013	0.863	0.707	0.777	0.760	0.957	0.773	Zeus
	0.977	0.012	0.914	0.977	0.945	0.938	0.979	0.909	DarkComet
	0.930	0.005	0.960	0.930	0.945	0.938	0.975	0.954	CyberGate
	0.926	0.011	0.911	0.926	0.918	0.908	0.976	0.943	Xtreme
	0.962	0.045	0.608	0.962	0.745	0.745	0.986	0.837	CTB-Locker
Weighted Avg.	0.855	0.015	0.864	0.855	0.853	0.840	0.975	0.870	

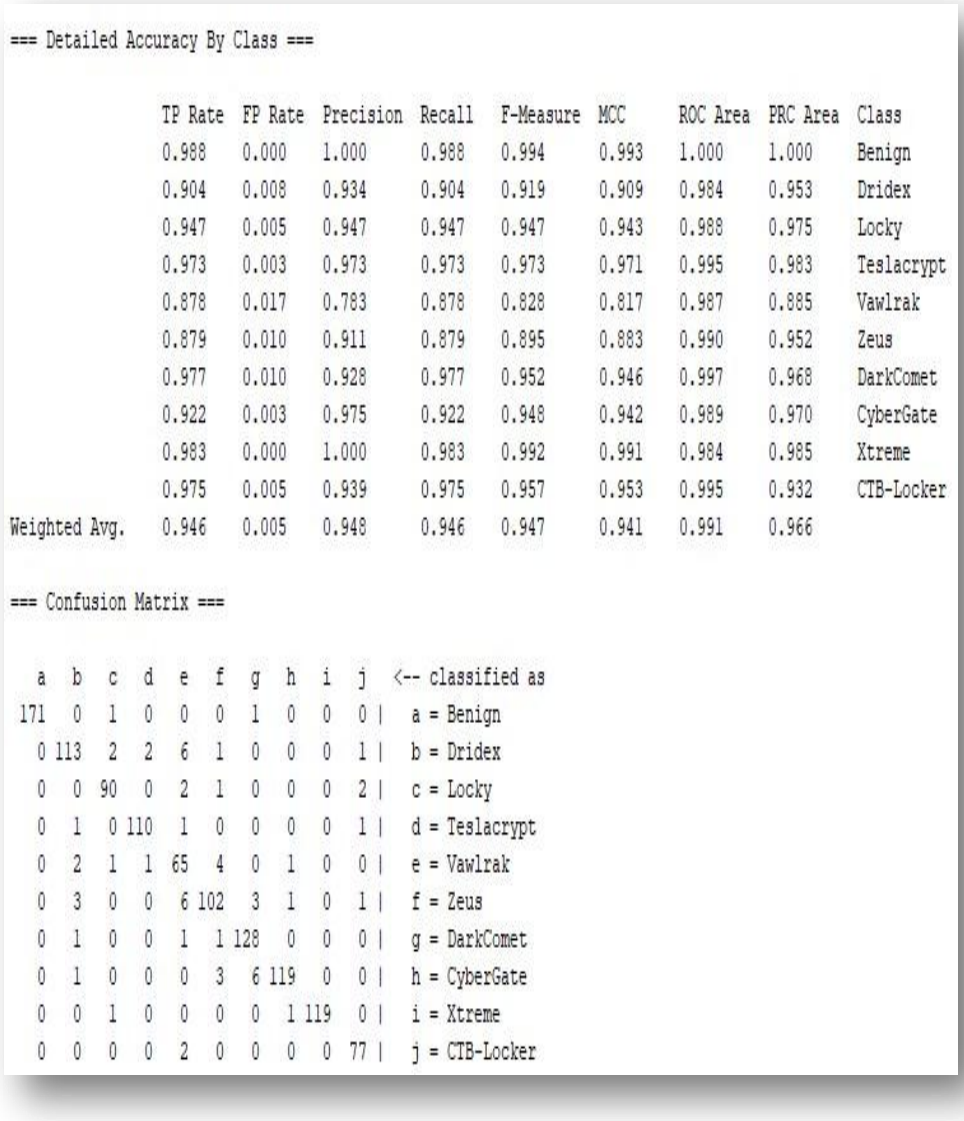
=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	<-- classified as
112	12	0	0	18	4	0	3	6	18	a = Benign
2	113	1	0	0	2	1	0	0	6	b = Dridex
2	0	89	0	0	1	0	0	1	2	c = Locky
0	1	0	110	0	1	0	0	0	1	d = Teslacrypt
5	11	0	1	46	1	0	2	2	6	e = Vawlrak
7	4	0	1	4	82	1	0	2	15	f = Zeus
1	0	0	0	0	2	128	0	0	0	g = DarkComet
0	1	0	0	0	0	7	120	0	1	h = CyberGate
3	0	0	1	0	2	3	0	112	0	i = Xtreme
2	0	1	0	0	0	0	0	0	76	j = CTB-Locker

Figure 4.1: Results obtained from Naïve Bayes Classifier

4.2 Regression Classifier:

Out of 1156 instances 1094 instances are classified accurately while 62 instances classified incorrectly. Hence accuracy is 0.9463.



```
=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.988    0.000    1.000      0.988    0.994      0.993    1.000     1.000     Benign
               0.904    0.008    0.934      0.904    0.919      0.909    0.984     0.953     Dridex
               0.947    0.005    0.947      0.947    0.947      0.943    0.988     0.975     Locky
               0.973    0.003    0.973      0.973    0.973      0.971    0.995     0.983     Teslacrypt
               0.878    0.017    0.783      0.878    0.828      0.817    0.987     0.885     Vawlrak
               0.879    0.010    0.911      0.879    0.895      0.883    0.990     0.952     Zeus
               0.977    0.010    0.928      0.977    0.952      0.946    0.997     0.968     DarkComet
               0.922    0.003    0.975      0.922    0.948      0.942    0.989     0.970     CyberGate
               0.983    0.000    1.000      0.983    0.992      0.991    0.984     0.985     Xtreme
               0.975    0.005    0.939      0.975    0.957      0.953    0.995     0.932     CTB-Locker
Weighted Avg.  0.946    0.005    0.948      0.946    0.947      0.941    0.991     0.966

=== Confusion Matrix ===

   a   b   c   d   e   f   g   h   i   j   <-- classified as
 171   0   1   0   0   0   1   0   0   0 |   a = Benign
   0 113   2   2   6   1   0   0   0   1 |   b = Dridex
   0   0  90   0   2   1   0   0   0   2 |   c = Locky
   0   1   0 110   1   0   0   0   0   1 |   d = Teslacrypt
   0   2   1   1  65   4   0   1   0   0 |   e = Vawlrak
   0   3   0   0   6 102   3   1   0   1 |   f = Zeus
   0   1   0   0   1   1 128   0   0   0 |   g = DarkComet
   0   1   0   0   0   3   6 119   0   0 |   h = CyberGate
   0   0   1   0   0   0   0   1 119   0 |   i = Xtreme
   0   0   0   0   2   0   0   0   0  77 |   j = CTB-Locker
```

Figure 4.2: Results obtained from Regression on weka tool

Figure 4.2 shows the confusion matrix and TP Rate, FPRate, Precision, Recall, FMeasure, MCC, ROCArea, PRCArea and their class of malware or benign by Regression Classifier.

4.3 J48 DecisionTree Classifier:

Out of 1156 instances 1086 instances are classified accurately while 70 instances classified incorrectly. Hence accuracy is 0.9394.

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Benign
	0.912	0.013	0.898	0.912	0.905	0.893	0.957	0.894	Dridex
	0.916	0.004	0.956	0.916	0.935	0.930	0.980	0.930	Locky
	0.982	0.005	0.957	0.982	0.969	0.966	0.988	0.931	Teslacrypt
	0.838	0.012	0.827	0.838	0.832	0.821	0.933	0.829	Vawlrak
	0.819	0.013	0.872	0.819	0.844	0.828	0.925	0.794	Zeus
	0.969	0.006	0.955	0.969	0.962	0.957	0.980	0.916	DarkComet
	0.938	0.009	0.931	0.938	0.934	0.926	0.977	0.871	CyberGate
	0.992	0.003	0.976	0.992	0.984	0.982	0.999	0.987	Xtreme
	0.962	0.003	0.962	0.962	0.962	0.959	0.992	0.955	CTB-Locker
Weighted Avg.	0.939	0.006	0.939	0.939	0.939	0.933	0.975	0.916	

=== Confusion Matrix ===

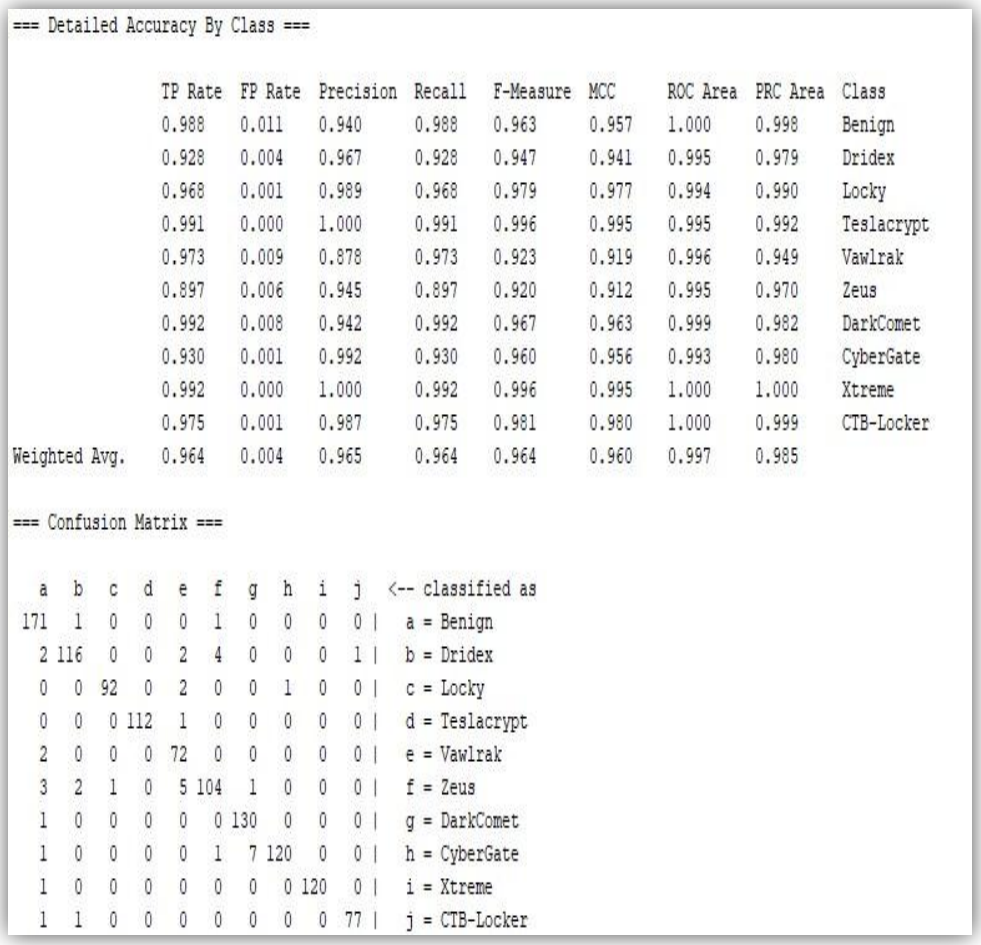
a	b	c	d	e	f	g	h	i	j	<-- classified as
173	0	0	0	0	0	0	0	0	0	a = Benign
0	114	0	0	3	7	0	1	0	0	b = Dridex
0	1	87	1	2	1	0	1	2	0	c = Locky
0	0	0	111	1	0	0	1	0	0	d = Teslacrypt
0	4	1	1	62	4	0	1	0	1	e = Vawlrak
0	6	3	2	4	95	1	2	1	2	f = Zeus
0	0	0	1	0	0	127	3	0	0	g = DarkComet
0	2	0	0	0	1	5	121	0	0	h = CyberGate
0	0	0	0	1	0	0	0	120	0	i = Xtreme
0	0	0	0	2	1	0	0	0	76	j = CTB-Locker

Figure 4.3: Results obtained from J48 on weka tool

Figure 4.3 shows the confusion matrix and TP Rate, FPRate, Precision, Recall, FMeasure, MCC, ROCArea, PRCArea and their class of malware or benign by J48 Classifier.

4.4 Random Forest Classifier:

Out of 1156 instances 1114 instances are classified accurately while 42 instances classified incorrectly. Hence accuracy is 0.9636.



=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.988	0.011	0.940	0.988	0.963	0.957	1.000	0.998	Benign
	0.928	0.004	0.967	0.928	0.947	0.941	0.995	0.979	Dridex
	0.968	0.001	0.989	0.968	0.979	0.977	0.994	0.990	Locky
	0.991	0.000	1.000	0.991	0.996	0.995	0.995	0.992	Teslacrypt
	0.973	0.009	0.878	0.973	0.923	0.919	0.996	0.949	Vawlrak
	0.897	0.006	0.945	0.897	0.920	0.912	0.995	0.970	Zeus
	0.992	0.008	0.942	0.992	0.967	0.963	0.999	0.982	DarkComet
	0.930	0.001	0.992	0.930	0.960	0.956	0.993	0.980	CyberGate
	0.992	0.000	1.000	0.992	0.996	0.995	1.000	1.000	Xtreme
	0.975	0.001	0.987	0.975	0.981	0.980	1.000	0.999	CTB-Locker
Weighted Avg.	0.964	0.004	0.965	0.964	0.964	0.960	0.997	0.985	

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	<-- classified as
171	1	0	0	0	1	0	0	0	0	a = Benign
2	116	0	0	2	4	0	0	0	1	b = Dridex
0	0	92	0	2	0	0	1	0	0	c = Locky
0	0	0	112	1	0	0	0	0	0	d = Teslacrypt
2	0	0	0	72	0	0	0	0	0	e = Vawlrak
3	2	1	0	5	104	1	0	0	0	f = Zeus
1	0	0	0	0	0	130	0	0	0	g = DarkComet
1	0	0	0	0	1	7	120	0	0	h = CyberGate
1	0	0	0	0	0	0	0	120	0	i = Xtreme
1	1	0	0	0	0	0	0	0	77	j = CTB-Locker

Figure 4.4: Results obtained from Random Forest Classifier

Figure 4.4 shows the confusion matrix and TP Rate, FPRate, Precision, Recall, FMeasure, MCC, ROCArea, PRCArea and their class of malware or benign by Random Forest Classifier.

4.5 Comparison of Existing Work to Proposed Work:

According to all result discussed above we can summarize them in following table in terms of average TPR, average FPR, F-measure and accuracy. We sort them according to highest accuracy and we can see that Random Forest has highest accuracy in our proposed work.

Table 4.1: Comparision of all algorithm in accuracy for proposed work

Classifier	Avg TPR	Avg FPR	F-measure	Accuracy
Random Forest	0.964	0.004	0.964	0.9636
Regression	0.946	0.005	0.947	0.9463
J48	0.939	0.006	0.939	0.9394
Naive Bayes	0.855	0.015	0.853	0.8373

Table 4.1 shows Random forest classifier have highest accuracy of 0.9636 and Naïve Bayes classifier have lowest accuracy of 0.8373 in comparison to all algorithms.

Comparison of Proposed Work to Existing Work:

Table 4.2 shows the comparison between proposed work and existing work.

Table 4.2: Comparison of proposed work to existing work

Classifier	Accuracy(existing)	Accuracy (proposed)
Random Forest	0.966	0.9636
J48	0.902	0.9394
Naive Bayes	0.796	0.8373

Hence our proposed work find higher accuracy in J48 and Naive Bayes algorithm whereas find almost same accuracy in Random Forest algorithms.

Summary and Conclusion

CHAPTER 5 SUMMARY AND CONCLUSIONS

In this chapter, the goals which we have achieved and some suggestions for future work are provided. First section includes summary of each, and everything covered in our approach. Second section describes what all conclusions can be derived from our work and last section discusses about future scope of the proposed work.

5.1 Summary:

For detection of Ransomware, we observe other malicious websites and we try whether is there any similarity in those website and we notice that through common feature we can recognize that malicious activity. We took raw data of malicious website with the help of Virus Total and applied Machine Learning Algorithm using Weka. We extract some important feature from raw data with the help of cuckoo sandbox. At last, we become able to classify that data.

5.2 Conclusion:

We have proposed to develop the technique for detection of ransomware, to determine the classification of ransomware and develop the method for increasing the detection rate of malware. We have used four classification algorithms in our technique. We identified ransomware with improved rate. Random Forest, Naïve Bayes, Regression and J48 classifier are used in our proposed work which shows accuracy of 0.9636,0.8373,0.9463 and 0.9394 respectively. Thus, Random Forest show highest accuracy of 0.9636. J48 and Naïve Bayes classifier shows higher accuracy than previous work and Random Forest find almost same accuracy. We have determined the different classification of ransomwares and malware which are dridex, locky, teslacrypt, vawtrak, zeus, darkcomet, cybergate, xtreme and CTB-locker. The proposed work has been simulated using Weka tool and simulation results are presented by confusion matrix and class of malwares.

5.3 Future Scope:

1. For future scope we may look into other ransomware families such as Cerber, Chimera Vipasana Zero Locker etc.
2. Feature extraction methods may also can be improved for more accuracy and more realibelity.

3. For new ransomware which digital signature is totally different we may select some feature from volatile memory and operating system.

Literature Cited

LITERATURE CITED

Alazab, R. V. M., KP, S., Poornachandran, P., Nemrat, A. and Venkatraman, S. 2019. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access*, 7: 41525-41550.

Alhawi, O. M. K., Baldwin, J. and Dehghantanha, A. 2018. Leveraging Machine Learning Techniques for Windows Ransomware Network Traffic Detection. *Cyber Threat Intelligence*, 70:93–106.

Amin, K., Robertson, Balzarotti, B. and Kirda. 2015. A Look Under the Hood of Ransomware Attacks. Detection of Intrusions and Malware and Vulnerability Assessment. DIMVA 2015. Lecture Notes in Computer Science, vol 9148. Springer, Cham. 12p.

Andronio, N., Zanero, S. and Maggi, F. 2015. HelDroid: Dissecting and Detecting Mobile Ransomware. *'In: International Symposium on Recent Advances in Intrusion Detection'* at Switzerland, during. December 12-15. pp.382–404.

Brewer, R. and Logrhythm. 2016. Ransomware attacks: detection, prevention and cure. *Netw. Secur.*, 9:5-9.

Cabaj, K., Gregorczyk, M. and Mazurczyk, W. 2018. Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics. *Int. J. Comput. Electr. Eng.*, 66(8): 353–368.

Carlin, D., Cowan, A., Kane, P. and Sezer, S. 2017. The effects of traditional anti-virus labels on malware detection using dynamic runtime opcodes. *IEEE Access*, 5: 17742– 17752.

Chen, J., Wang, C., Zhao, Z., Chen, K., Du, R. and GJ. 2018. Uncovering the face of android ransomware: characterization and real-time detection. *IEEE Trans Inf Forens. Secur.*, 13(5):286–300.

Cusack, G., Michel, O. and Keller, E. 2018. Machine learning-based detection of ransomware using SDN. *'In: ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization'* at USA, during. August 12-15. pp. 1–6.

Haddadi, F., Khanchi, S., Shetabi, M., and Derhami, V. 2010. Intrusion detection and attack classification using feed-forward neural network. *'In: 2010 Second International Conference on Computer and Network Technology'* at USA, during. July 8-12. pp. 262–266

Manoun, A., Robert, L. and Paul, W. 2011. Malware Detection Based on Structural and Behavioral Features of API Calls. '*In:* International Cyber Resilience Conference' at Perth (Western Australia), during. August 20-23. pp. 13-19.

Michael H. L., Andrew, C. and Jamie L. 2014. The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory. *Int. J. Comput. Secur.*, 21:78–84.

Pundir, S. L. 2013. Feature Selection Using Random Forest in Intrusion Detection System. *Int. j. adv. Eng***.,** 6 (3):13-19.

Rai, M. and Mandoria, H. L. 2019. A Study on Cyber Crimes, Cyber Criminals and major Security Breaches. *Int. Res. J. Eng. Technol.*, 6(7): 233-240.

Rhode, M., Burnap, P. and Jones, K. **2018.** Early-stage malware prediction using recurrent neural networks. *Int. J. Comput. Secur.*, 77:578–594.

Salman, T., Bhamare, D., Erbad, A., Jain, R. and Samaka, M. 2017. Machine learning for anomaly detection and categorization in multi-cloud environments. '*In: 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing'*, at New York (USA), during. June 26-28. pp. 97-103.

Scaife, N., Carter, H., Traynor, P. and Butler, K. R. B. 2016. CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data. '*In: 2016 IEEE 36th International Conference on Distributed Computing Systems*' at Nara (Japan), during. June 27-30. pp. 303-312.

Scalas, M., Maiorca, D., Mercaldo, F., Visaggio, C. A., Martinelli, F. and Giacinto, G. 2019. On the Effectiveness of System API-Related Information for Android Ransomware Detection. *Comput. Secur.*, 86: 168-182.

Sgandurra, D., Munoz, L., Mohsen, R. and Lupu, E. C. 2016. Automated dynamic analysis of ransomware: Benefits, limitations and use for detection. *IEEE Access*, 9: 42– 45.

Shankarapani, M. K., Ramamoorthy, S., Movva, R. S. and Mukkamala, S. 2010. Malware detection using assembly and API call sequences. *J. Comput. Virol.*, 7(2): pp. 107–119.

Shaukat, S. K. and Ribeiro, V. J. 2018. RansomWall- A layered defense system against cryptographic ransomware attacks using machine learning. *'In: 2018 10th International Conference on Communication Systems & Networks'* at Bengaluru, during. January 3-7. pp. 356-363.

Shukla, M., Mondal, S. and Lodha, S. 2016. Virtualized environment for mitigating ransomware threats. *'In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security'* at Vienna (Austria), during. October 24 - 28. pp. 1784-1786.

Song, S., Kim, B. and Lee S. 2016. The Effective Ransomware Prevention Technique Using Process Monitoring on Android Platform. *Mob. Inf. Syst.*, 2016(11): 1–9.

Takeuchi, Y., Sakai, K. and Fukumoto, S. 2018. Detecting ransomware using support vector machines. *'In: Proceedings of the 47th International Conference on Parallel Processing Companion'* at USA, during. August 13-16. pp. 1–6.

Vinayakumar, R., Soman, K. P., Velan, K. K. S. and Ganorkar, S. 2017. Evaluating shallow and deep networks for ransomware detection and classification. *'In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)'* at Udupi (India), during. September 13-16. pp. 259-265.

Zhao, M., Zhang, T. and Yuan, Z. 2011. Antimaldroid: An efficient svm-based malware detection framework for android. '*In*: *International Conference on Information Computing and Applications*' at Berlin, during. September 12-15. pp. 158–166.

Appendices

APPENDIX

```
#import libraries
library(RWeka)
library(kernlab)
 library(Boruta)
library(class)
 library(dplyr)
 library(lubridate)
library(gmodels)
 library(ggvis)
library(e1071)
 library(randomForest)
 #set directory
 home_dir <- "C:\\Users\\Kateryna\\Desktop\\Thesisstuff\\selectedfeatures.csv"
 #load data from .csv file
 selected_apis <- read.csv(home_dir, header=TRUE)
 #define the normalization function
normalize <- function(x) {
 num <- x - min(x)
 denom <- max(x) - min(x)
 return (num/denom)
 }coln = ncol(selected_apis)
coln1=coln+1
#J48
 fit <- J48(as.factor(Class)~., data=selected_apis.train1)
 # summarize the fit summary(fit)
 # make predictions
 predictions <- predict(fit, selected_apis.test1)
 # summarize accuracy
 CrossTable(selected_apis.testlabels, predictions, type="Classification")
 #two-class
 fit2 <- ksvm(as.factor(Malware)~., data=selected_apis.train2)
```

```
 # summarize the fit
summary(fit2)
 # make predictions
predictions <- predict(fit2, selected_apis.test2)
 # summarize accuracy
CrossTable(selected_apis.testlabels.twoway, predictions, type="Classification")
```

#Naive Bayes

```
 fit <- naiveBayes(as.factor(Class)~.,
data=selected_apis.train1)
 # summarize the fit
summary(fit)
# make predictions
 predictions <- predict(fit, selected_apis.test1)
# summarize accuracy CrossTable(selected_apis.testlabels, predictions,
type="Classification")
 #two-way fit
<- naiveBayes(as.factor(Malware)~., data=selected_apis.train2)
# summarize the fit summary
(fit)
 # make predictions
 predictions <- predict(fit, selected_apis.test2)
# summarize accuracy
 CrossTable(selected_apis.testlabels, predictions, type="Classification")
```

#RandomForest

```
 fit <- randomForest(as.factor(Class)~.,
data=selected_apis.train1)
 # summarize the fit summary
(fit)
# make predictions
predictions <- predict(fit, selected_apis.test1)
 # summarize accuracy
```

```
 CrossTable(selected_apis.testlabels,
 predictions, type="CClassification")
 #two-way fit <- randomForest(as.factor(Malware)~.,
data=selected_apis.train2)
# summarize the fit summary
(fit)
 # make predictions
 predictions <- predict(fit, selected_apis.test2)
 # summarize accuracy
 CrossTable(selected_apis.testlabels, predictions, type="Classification")
```

#Regression classification

```
model_pred <- reg(train = selected_apis.train1, test = selected_apis.test1,
 cl = selected_apis.trainlabels, k=1)
CrossTable(x = selected_apis.testlabels,
 y = model_pred, prop.chisq=FALSE)
 #two-way
 model_twoway<-knn(train = selected_apis.train2,
test = selected_apis.test2,
cl = selected_apis.trainlabels.twoway, k=1)
 prob <- attr(model_pred, "prob")
CrossTable(x = selected_apis.testlabels.twoway,
y = model_twoway, prop.chisq=FALSE)
```

CURRICULUM-VITAE

Name : **Shivam Kumar Pujari**

Phone Number : **8077070648**

Mailing Address : Vikas Nagar Sec03
Haripur Nayak Kusumkhera
Haldwani
Pin – 263139
District – Nainital
Uttarakhand (India)

Permanent Address : Vikas Nagar Sec03 Haripur Nayak Haldwani
Pin – 263139
District – Nainital
Uttarakhand (India)

E-mail : shivampj@gmail.com

Career Objective: To be a Researcher in the field of information technology.

Educational Qualification: (should be written in reverse chronological order)

S. No.	Examination Passed	Institution	Year	Percentage/ CGPA
1.	M. Tech. (Information Technology)	G.B.P.U.A&T, Pantnagar	2021	7.598 CGPA
2.	B. Tech. (Computer Science Engineering)	Uttrakhand Technical University.	2017	59.9%
3.	Class-12th	Uttrakhand State Board	2013	63.8%
4	High school	Uttrakhand State Board	2011	73%

- **Specialization: Major:** Information Technology **Minor**: NIL
- **Software Skills:** Software Skills, Creative Thinking, Ambitious and focused.
- **Professional Skills:**
  - **Coding Languages:** C, Java.
  - **Operating System**: WINDOWS, UNIX/LINUX.
  - **Tools**: Weka.

Place: Haldwani, Uttarakhand
Date: February, 2021

Shivam Pujari

(Shivam Kumar Pujari)

Name	**:**	**Shivam Kr Pujari**	**Id. No.**	**:** 54104
Semester & year of admission	**:**	1st, 2018-2019	**Degree**	**:** Master of Technology
Major	**:**	Information Technology	**Department**	**:** Information Technology
Thesis Title	**:**	**Detection of Ransomware using Machine Learning**		
Advisor	**:**	**Dr. H. L. Mandoria , Professor & Head IT**		
No. of Pages	**:**	40		**: Dr. H. L. Mandoria , Professor & Head IT**

ABSTRACT

Today's world depends on the cyber world, since it is very useful for collecting information, data and transporting them. Since anyone can use it, for security purpose of data a technique called encryption is made. Unfortunately, this strong technique of encryption for security is also useful for hackers to lock any file or system by encryption by infecting malware. This type of malware which encrypt data is called Ransomware.

In the digital world there are various types of attacks for a different aspect of motive such as economic benefits, personal issues, religious issues, political benefits, or special propaganda, etc. Ransomware attacks are for financial benefits and most popular in today's world.

We purpose a method in which we can classify and detect ransomware and some other malware also.

(H L Mandoria)
Advisor

(Shivam Kr Pujari)
Author

नाम	:	शिवम कुमार पुजारी	**परिचयांकसंख्या**	:	54104
सत्र एवं प्रवेश वर्ष	:	प्रथम 2018–19	**उपाधि**	:	प्रोधोगिकी में स्नातकोत्तर
विभाग	:	सूचना प्रौद्योगिकी			
मुख्य विषय	:	सूचना प्रौद्योगिकी			
शोध शीर्षक	:	**"मशीन लर्निंग का उपयोग कर रैंसमवेयर का पता लगाना"**			
पृष्ठ संख्या	:	40	**सलाहकार**	:	डॉ0 एच0 एल0 मंडोरिया, (प्रोफेसर हेड आईटी)

सारांश

आज की दुनिया साइबर दुनिया पर निर्भर करती है, क्योंकि यह जानकारी एकत्र करने, डेटा और उन्हें परिवहन करने के लिए बहुत उपयोगी है। चूंकि कोई भी इसका उपयोग कर सकता है, डेटा के सुरक्षा उद्देश्य के लिए एन्क्रिप्शन नामक तकनीक बनाई जाती है। दुर्भाग्य से, सुरक्षा के लिए एन्क्रिप्शन की यह मजबूत तकनीक हैकर्स के लिए भी उपयोगी है कि वे मैलवेयर को संक्रमित करके एन्क्रिप्शन द्वारा किसी भी फाइल या सिस्टम को लॉक कर सकते हैं। इस प्रकार के मैलवेयर जो डेटा एन्क्रिप्ट करते हैं, उन्हें रैंसमवेयर कहा जाता है।

डिजिटल दुनिया में मकसद के एक अलग पहलू के लिए विभिन्न प्रकार के हमले होते हैं जैसे कि आर्थिक लाभ, व्यक्तिगत मुद्दे, धार्मिक मुद्दे, राजनीतिक लाभ, या विशेष प्रचार, आदि। रैंसमवेयर हमले वित्तीय लाभ के लिए होते हैं और आज की दुनिया में सबसे लोकप्रिय हैं।

हमारा एक तरीका है जिसमें हम रैंसमवेयर और कुछ अन्य मैलवेयर का भी वर्गीकरण और पता लगा सकते हैं।

(एच0 एल0 मंडोरिया)
सलाहकार

(शिवम कुमार पुजारी)
लेखक