PROJECT REPORT ON

# A Machine Learning Framework for DomainGeneration Algorithm (DGA)-Based Malware Detection

SUBMITTED BY

Harsh Dobariya

Akshay Kalapgar

Mohit Kamble

Siddesh Parab

GUIDED BY

Prof. Abhay E. Patil

MCT
MANJARA CHARITABLE TRUST
RAJIV GANDHI INSTITUTE OF TECHNOLOGY, MUMBAI
(Permanently Affiliated to University of Mumbai)
Juhu Versova Link Road, Andheri (West), Mumbai-53

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**UNIVERSITY OF MUMBAI**
**2021**

MCT
MANJARA CHARITABLE TRUST
RAJIV GANDHI INSTITUTE OF TECHNOLOGY, MUMBAI
(Permanently Affiliated to University of Mumbai)
Juhu Versova Link Road, Andheri (West), Mumbai-53

## DEPARTMENT OF INFORMATION TECHNOLOGY

## ***CERTIFICATE***

Date: ____________

This is to certify that, the project work embodied in this report entitled, *"**A Machine Learning Framework for Domain Generation Algorithm (DGA)-Based Malware Detection**"* submitted by "***Harsh Dobariya*** bearing Roll No. 814", "***Akshay Kalapgar*** bearing Roll No. 831", "***Mohit Kamble*** bearing Roll No. 832", "***Siddesh Parab*** bearing Roll No. 850"for the award of ***Bachelor ofEngineering*****(B.E.)** degree in the subject of ***Information Technology***, is a work carried out by them under my guidance and supervision within the institute. The work described in this project report is carried out by the concerned students and has not been submitted for the award of anyother degree of the University of Mumbai.

Further, it is certifying that the students were regular during the academic year 2020-20 and have worked under the guidance of concerned faculty until the submission of this project work at ***Rajiv Gandhi Institute of Technology, Mumbai.***

| | |
|---|---|
| Prof. Abhay E. Patil | Prof. Swapnil Gharat |
| **Project Guide** | **Project Coordinator** |

| | |
|---|---|
| Dr. Sunil B. Wankhade | Dr. Sanjay U. Bokade |
| **Head of Department** | **Principal** |

# CERTIFICATE OF APPROVAL

This project report entitled

***A Machine Learning Framework for Domain Generation Algorithm(DGA)-Based Malware Detection***

Submitted by:

| | |
|---|---|
| ***HARSH DOBARIYA*** | ***814*** |
| ***AKSHAY KALAPGAR*** | ***831*** |
| ***MOHIT KAMBLE*** | ***832*** |
| ***SIDDESH PARAB*** | ***850*** |

In partial fulfillment of the requirements of the degree of **Bachelor of Engineering** in **Information Technology** is approved.

**Internal Examiner**

______________________

SEAL OF
INSTITUTE

**External Examiner**

______________________

Date:

Place:

# Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

| **ROLL NO.** | **NAME** | **SIGNATURE** |
|---|---|---|
| 814 | HARSH DOBARIYA | |
| 831 | AKSHAY KALAPGAR | Akshay |
| 832 | MOHIT KAMBLE | Mohit |
| 850 | SIDDESH PARAB | Siddesh |

Date:

Place:

# Acknowledgement

With all reverence, we take the opportunity to express our deep sense of gratitude and wholehearted indebtedness to our respected guide, **Prof. Abhay E. Patil**, Department of Information Technology, Rajiv Gandhi Institute of Technology, Mumbai. From the day of conception of this project his active involvement and motivating guidance on day-to-day basis has made it possible for us to complete this challenging work in time.

We would like to express a deep sense of gratitude to our respected **Head of the Department**, **Dr. Sunil B. Wankhade** who went all the way out to help us in all genuine cases during the course of doing this project. We wish to express our sincere thanks to **Dr. Sanjay U. Bokade**, **Principal**, Rajiv Gandhi Institute of Technology, Mumbai and would to like to acknowledge specifically for giving guidance, encouragement and inspiration throughout the academics.

We would like to thank all the staff of Information Technology Department who continuously supported and motivated during our work. Also, we would like to thank our colleagues for their continuous support and motivation during the project work. Finally, we would like to express our gratitude to our family for their eternal belief in us. We would not be where we are today without their support and encouragement.

HARSH DOBARIYA

AKSHAY KALAPGAR

MOHIT KAMBLE

SIDDESH PARAB

Date:

Place:

# ABSTRACT

Attackers usually use a Command and Control (C2) server to manipulate the communication. In order to perform an attack, threat actors often employ a Domain Generation Algorithm (DGA), which can allow malware to communicate with C2 by generating a variety of network locations. Traditional malware control methods, such as blacklisting, are insufficient to handle DGA threats.In this paper, we propose a machine learning framework for identifying and detecting DGA domains to alleviate the threat. We collect real-time threat data from the real-life traffic over a one- year period. We also propose a deep learning model to classify a large number of DGA domains. The proposed machine learning framework consists of a two level model and a prediction model. In the two-level model, we first classify the DGA domains apart from normal domains and then use the clustering method to identify the algorithms that generate those DGA domains. To differentiate DGA domain names from normal domain names, researchers have discovered that DGA-generated domain names contain significant features. Therefore, many studies aim to target blocking those DGA domain names as a defense approach. The DGA that generates the domain fluxing botnet needs to be known so that we can take countermeasures. Several studies have looked at understanding and reverse engineering the inner workings of botnets. The study focused on domain fluxing malware and relied on the binary extraction for DGA. Their approach is only effective for certain types of malware. To be precise, we achieve an accuracy of 95.89% for the classification in the framework and 97.79% in the DNN model, 92.45% for the second-level clustering, and 95.21% for the prediction in the framework.

# TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER 1
# INTRODUCTION

## 1.1 Introduction to Project

Malware attackers attempt to infiltrate layers of protection and defensive solutions, resulting in threats on a computer network and its assets. Anti-malware software have been widely used in enterprises for a long time since they can provide some level of security on computer networks and systems to detect and mitigate malware attacks. However, many anti-malware solutions typically utilize static string matching approaches, hashing schemes, or network communication white listing. These solutions are too simple to resolve sophisticate malware attacks, which can hide communication channels to bypass most detection schemes by purposely integrating evasive techniques. The issue has posed a serious threat to the security of an enterprise and it is also a grand challenge that needs to be addressed.

In this project, we first propose a machine learning framework to classify and detect DGA malware and develop a DNN model to classify the large datasets of DGA domains that we gradually collected. We then experimentally evaluate the proposed framework through a comparison of various machine learning approaches and a deep learning model. The general goal of our machine learning framework is to determine which algorithm is employed so that our proposed framework can prevent future communications from the C2.

The comparison results provide us a useful guideline for our future study in DGA detection and prediction. In our future research, we will also apply deep learning in clustering and predictions that are out of the scope of this paper.

## 1.2 Background of the study

Some of the sophisticate malware attackers use either a static or dynamic method to communicate with a centralized server to service a Command and Control (C2). In a static method, everything is fixed. For example, the malware has both a fixed IP address and a fixed domain name permanently (i.e., its domain name will not change throughout its lifespan). Thus, as long this malware has been identified as a threat, a simple rule can be applied to resolve this malware threat issue.

In a dynamic method, Domain Generation Algorithm (DGA) has been commonly used to communicate back to a variety of servers. The DGA is a sequencing algorithm that is used to periodically generate a large number of domain names, which are often used by malware to evade domain-based firewall controls.

The generated domain names give malicious actors the opportunity to hide their C2 servers so that it is hard for the enterprise to identify the DGA. The domains generated by DGAs are short-lived registered domains and they are easier for human to identify but harder for machines to detect automatically.

## 1.3 Statement of the Problem

Multiple conditions for a DGA to function in a network environment where filtering results in a firewall that protects the communication and an empty cell in an Internet domain that result in an NXDOMAIN error.

Firewall blacklisting constantly expands as the multiple sources of inputs expand filtering rules. However, sequences in a DGA may not be known to these inputs promptly. Moreover, for the malware that communicates with an appropriate domain correctly, a threat actor must register each respective domain name in the sequence to maintain the C2 or risk the loss of a node in the distribution.

Our research problem is to accurately identify and cluster domains that originate from known DGA-based techniques where we target to develop a security approach that autonomously mitigates network communications to unknown threats in a sequence

## 1.4 Objectives of the Project

The main objective of our project is to create a malware detection system, which can test analyzeand verify each and every domain using a training data set, and provide an outcome whether the domain contains malware or not. This system will help the networking institutions take effective and timely decisions, replacing the existing time taking, long procedures and paper work required to check the credibility of the domains. The system aims to achieve:

- In DBSCAN algorithm, we use the features described above to calculate the domain distance and to group the domains that are generated by the same DGA together according to their domain feature difference.
- Distinguish the model from training and prediction stages.
- The nodes in each layer are fully connected to the nodes in the next will not miss any local minima, but it will take a long time to converge.

## 1.5 Purpose of study

The purpose of our work is to present a data analysis model for effective classification among domains, and enhanced decision making in malware detection to the applicants. For users, malware risk is a major challenge, which directly or indirectly affects the reliability of the network.

The model allows storage of the huge volume of domain data, which is then cleaned and features are extracted and reduced.

## 1.6 Scope of the work

The networking industry is among many industries which have massive and useful data about their customers but very few are utilizing this set of information to enhance the user experience and using the data information to prevent fraud. The challenge is not about dealing with trillions of bytes of data; it is about getting started with a quantitative approach so that you can drive value from your data, whatever size that data is. They are very well aware of the fact that if the data can be used effectively they can fulfill the needs of user accurately.

Major scope of malware detection in networking industry in the present and near future includes:

- **User Segmentation**

  User segmentation is classifying the customer on the basis of the age, gender, behavior, habits etc. The networking industry has agreed that customer retention is a keyto company's success and is becoming more user-centric.

  The analysis helps the networking services to analyze the spending pattern of an individual user which help them to offer services time to time to their users. The analysis also helps in identifying a valuable user, one who spent the most money. And through this data analysis, they can provide best to the user. This ultimately will lead to increased user satisfaction. Additionally, it will also help the networking industry understand the spending patterns of the users, domain usage, and consequently cross-selling of various domains.

- **Malware Detection**

  This is one of the biggest problems that every networking industry has been facing. With the increase in an online transaction, the incidents of fraud have increased too. To avoid such fraud the networking industry is using the malware detecting technology which helps them to understand the domain history and using pattern of users and increase security on every unusual transaction. This will help them to mitigate any fraudulent activities before it grows bigger.

- **Offering Personalized Services**

  Offering Personalized Services to a user is nothing but the next level of marketing where they offer product and services to as per users' interest and requirement. Yes, it is possible with the help of data analysis. Networking industry collects data from e-commerce website and malware detecting technology analyze the buying habit, interest and requirements of individual user by doing data analysis. This data analysis helps to offer services and products to the user time to time as per their interest and requirements which help them to retain the present user and attract the new one.

- **Risk Management**

  Risk Management is an important factor in every industry and risk in the networking industry can come in any form like an unrecoverable fees after malware detection, and fraudulent activities. It cannot be stopped completely but the early detection of risk can be helpful in preventing huge losses. With the help of data analysis, they can perform the risk management analysis and can minimize the user's risk.

- **Addressing Compliance Requirements**

  Networking services are required to do regular compliance, audit and maintain certain regulations for their data, privacy and security measures. They now have access to billions of user's needs. They can use data to cater to serve the user more effectively. Cloud-based analytics packages can sync in real time with your data systems, creating actionable insight dynamically.

## 1.7 Limitation of the Project

The limitation of the project is based on the training of dataset and accuracy achieved credit analysis.

- For training of large dataset, high computation system is required which can handle large scale processing.
- Domain generation algorithms are not intended to replace the data entities; they just try to show the malware-detecting process.
- It not possible to determine in advance the accuracy of the developed algorithms.
- Resistance to face new challenges related with privacy and deal with many users, technologies and data sources.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Technical Paper Review

The amount of research carried out and successfully implemented in the field of Machine Learning is currently limited due to novelty of the technology. This section elaborates an extensive study of the available research in order to gather enough information to meet our objectives.

### 2.1.1 "A Machine Learning Framework for Domain Generation AlgorithmBased Malware Detection." by Yi Li, Kaiqi Xiong, Tommy Chin, Chengbin Hu.

This paper gives an idea for the domain generation algorithm and group of other algorithms and machine learning models. The goal is to detect malware if any domain contains it on basis of the domain generation algorithm. This paper has summarized various algorithms and machine learning models and also a framework has proposed. The goal is to distinguish between malicious and no malicious behaviors. The promise of such systems is great. Theoretically, this type of solution can deal with all attacks, both known and unknown. Moreover, it promises to free the user from having to keep the system updated, since there is no use of attack signatures.

**Advantages:** In the secondlevel clustering we apply the DBSCAN algorithm. Only the DGA domains obtained from the first-level classification will be used for clustering.

**Disadvantages:** Research problem is to accurately identify and cluster domains that originate from known DGA-based techniques where we target to develop a security approach that autonomously mitigates network communications to unknown threats in a Sequence.

### 2.1.2 “Learning and Classification of Malware Behavior.” by Learning and Classification of Malware Behavior.

This paper is about the classification of the malware behavior and it gives a brief idea about how each of the various malware is generated and how it affects the system of the user. A in depth research is done here which was helpful to understand the malware behavior. Various questions about the malware behavior are answered in this paper. Behavior-based malware detection evaluates an object based on its intended actions before it can actually execute that behavior. An object’s behavior, or in some cases its potential behavior, is analyzed for suspicious activities. Attempts to perform actions that are clearly abnormal or unauthorized would indicate the object is malicious, or at least suspicious.

**Advantages:** Results show that 70% of malware instances not identified by anti-virus software can be correctly classified by our approach.

**Disadvantages:** Proposed machine learning framework aims to solve the problem of detecting DGA sequences using machine learning techniques derived from observations in a network. A further weakness of the proposed supervised classification approach is its inability to find structure in new malware families not present in a training corpus. The presence of unknown malware families can be detected by the rejection mechanism used in our classifiers, yet no further distinction among rejected instances is possible. Whether this is a serious disadvantage in comparison to clustering methods is to be seen in practice.

### 2.1.3 “An SDN based framework for guaranteeing security and performance in informationcentric cloud networks.” by Chatterjee, D. Tosh, S. Shetty, K. Xiong, and C. Kamhou.

This paper gives a glimpse of the Software Defined Networking technology. Here the main goal was to detect any security threats which could affect the cloud networks. The performance in the information gathering is also measured along with the security threats. All different aspects are tested regarding the possibilities of a breach in the network. The traditional algorithms are studied in brief and are implemented especially where there is a very long pattern in length. The basic purpose of SDN is to allow users to virtualize their hardware. A software-defined network attempts to build a computer network by separating it into two segments. The control plane can provide performance and fault management of NetFlow, IPFIX and SNMP protocols. This plane is generally used to manage configurations of devices connected to the SDN on a remote access basis.

**Advantages:** The attempting times were less than of the traditional algorithms especially in the case of a very long minimum pattern length. One of the reasons why SDN has risen to prominence has been the number of problems inherent in maintaining a traditional legacy network.

**Disadvantages:** The lengthy processing time when directly extended to the multiple string patterns matching. SDN systems are still a new technology. Being a new technology, there are still areas that could use improvement. If a large number of undeclared routes are brought into the network at the same time, they will request a specific route. Unfortunately, this influx in requests can make it difficult for the network to respond to actual requests. One of the disadvantages of a SDN network is that since you are eliminating use of the physical routers and switches, you won't have the security that comes with them. The main one that you will be missing is the firewall. This can leave your network more vulnerable if you're not careful.

### 2.1.4 “A two-hashing table multiple string pattern matching algorithm.” by C. Khancom e, V. Boonjing, and P. Chanvarasuth.

This paper is about the traditional algorithms which were used when there was a long pattern in a string. The two-hashing table was used to solve the problem of multiple string patterns. The hashing method just described above is also known as open **or** external hashing which allows data to be stored in chained lists with theoretically unlimited storage space. This does not require more keys, and the chaining allows larger amounts of data to be handled. The word “open” refers to open addressing. With closed hashing**,** the number of available keys is limited by the table’s capacity. If you try to store more data than its capacity allows for, overflow will occur. When running through the table again, it will be checked for available locations where the overflows can be placed.

**Advantages:** The biggest advantage of using a hash table is being able **to** search through large amounts of data quickly**.** However, this poses a challenge to the database’s architects who must estimate the required size well in advance to keep the risk of collision low. Many data types can be used in hash tables as long as hash values can be calculated from them.

**Disadvantages:** The disadvantages of hash tables include the fact that databases can degrade if they go through a large number of collisions**.** The probability that a collision will occur increases with the amount of data. A large number of hash functions do not have the ability to move to the next or previous data set. Main advantage is synchronization. In many situations, hash tables turn out to be more efficient than search trees or any other table lookup structure. For this reason, they are widely used in many kinds of computer software, particularly for associative arrays, database indexing, caches and sets. Hash collisions are practically unavoidable. When hashing a random subset of a large set of possible keys. Hash tables become quite inefficient when there are many collisions. Hash table does not allow null values, like hash map.

## 2.2 Existing System

Threat models: Multiple conditions for a DGA to function in a network environment where filtering results in a firewall that protects the communication and an empty cell in an Internet domain those results in NXDOMAIN error.

Each HMM date record represents a series of domain observations. First sequences of domain name are processed by a feature extractor and each of these feature vectors is used as a training record.

Then, similar sequences are clustered as a group of DGA domain names with certain outcomes. After the training process, if a sequence does not have an HMM sequence representation (or it is not presented in the training data but the test data), the HMM model then generates the future predicted results. Otherwise, we will use an existing HMM sequence presentation.

**Disadvantages of Existing System:**

1. Firewall protects the communication and an empty cell in an internet domain that results inno domain error.
2. Queries not matching the knowledge are stored in a backlog of the software.

# CHAPTER 3

# METHODOLOGY

## 3.1 Proposed System

In our proposed system, Domains extracted from DGAs. Machine learning framework that encompasses multiple feature extraction techniques and the models to classify the DGA domains from normal domains, cluster the DGA domains, and predict a DGA domain.

A deep learning model to handle large datasets multiple on- line sources from simple Google searching provide example codes for a DGA construction.

Online threat intelligence feeds give an approach to examining current and live threats in real- world environment.

Using real-time active malicious domains derived from DGAs on the public Internet measures the accuracy of the proposed approach. The structure of the data is presented in a CSV format of domain names, originating malware, and DGA membership with the daily file size of approximate 110MB.

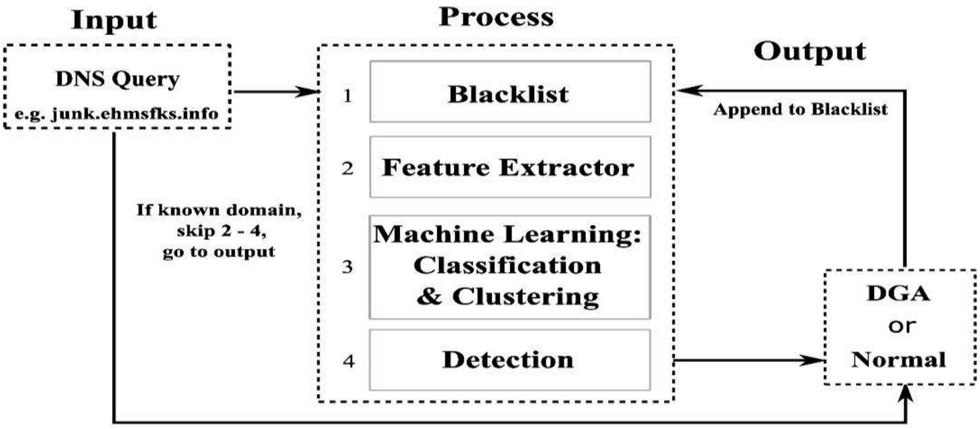


**Fig. 3.1:** Diagrammatical Representation of working of the system

### Flowchart of the System

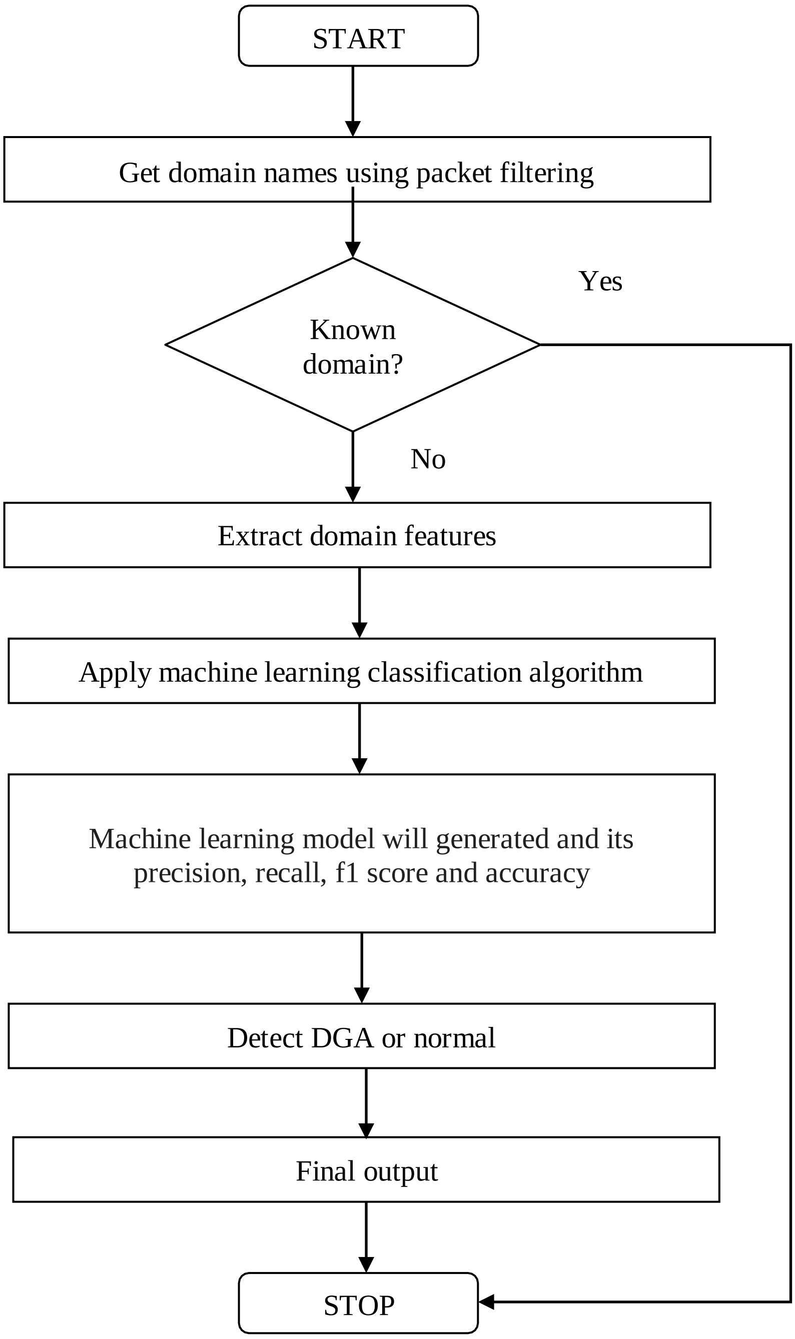

START
Get domain names using packet filtering
Known domain?
Yes
No
Extract domain features
Apply machine learning classification algorithm
Machine learning model will generated and its precision, recall, f1 score and accuracy
Detect DGA or normal
Final output
STOP

**Fig. 3.2 -** Flowchart of the System

## Class diagram of the System

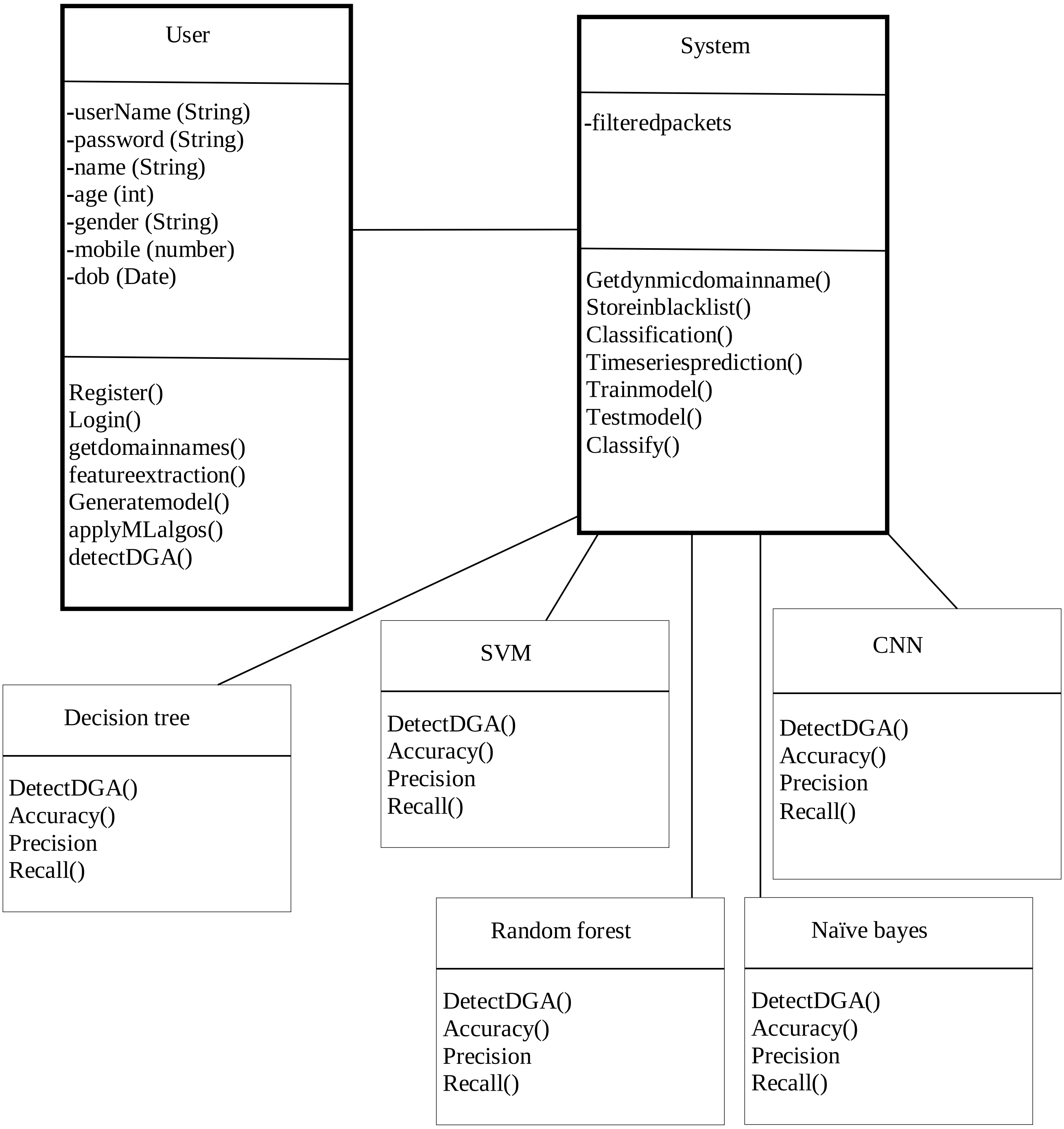


**Fig. 3.3** – Class diagram of the System

In the project, we have assumed that DGA domains have groups of very significant characters from normal domains. By grouping domains according to their features, the authors applied a machine learning classifier to distinguish DGA domains from normal domains easily.

We propose a machine learning framework that consists of three important steps, as shownin Figure below. We first have the DNS queries with the payload as the input. In order to classify DGA domain names.

**Advantages of Proposed System:**

1. Domain Generation Algorithm (DGA), which allows malware to generate numerous domainnames until it finds its corresponding C&C server.
2. It is highly resilient to detection systems and reverse engineering, while allowing the C&Cserver to have several redundant domain names.

## 3.2 Principle of working

We first have the DNS queries with the payload as the input. Then, the DNS queries will be passed to our process step, which consists of 4 important components:

1. We first use a domain-request packet filter to get domain names and then store them in a dynamic blacklist. If the input is a known domain, we will skip (2) - (4) and directly go to the output; otherwise, we will proceed to the next component.
2. Then, a feature extractor is used to extract domain features.
3. Next, we apply the first-level classification to distinguish DGA domains from non-DGA domains and the second-level clustering to group similar DGA domains.
4. Finally, we use a time-series model to predict the features of a domain. After the domain name goes through the process step, we will append this domain to the dynamic blacklist. The rest of this section discusses the four components of the process step in details.

## 3.3 Filtering packet data Feature extraction work:-

- To filter packet data we are using pyshark which captures network packets.We will store this packet information in pcap format
- By reading packet we will filter the data and obtain domain name.Packet flow also obtained from this.
- If domain name extracted in this found in blacklist we will stop further steps.
- With the python coding we will calculate the following feature
- Length- length of domain name.
- Meaningful Word Ratio: dictionary will be maintained of meaningful word and output willbe taken by dividing with length of domain name
- Percentage of Numerical Characters: numeric character involved in domain name system.
- Pronounce ability Score frequency of text in domain calculated.
- Percentage of the Length of the Longest Meaningful String (LMS): dividing the meaningfulword with the length of domain.
- Levenshtein Edit Distance: It measures the minimum number of single-character edits between a current domain and its previous domain in a stream of DNS queries received by the server. The Levenshtein distance is calculated based on a domain and its predecessor

## 3.4 Machine learning classification:-

Following algorithms will be applied on feature obtained above.

- **Decision Tree:** It calculates entropy and information gain and output generated but has problem of over fitting. We will generate module with the selected feature.

- **ANN:** It's Artificial Neural Networks. Here we give input layer, hidden layer and output layer. Then with the feature we calculate the output.

- **Multiple Logistic regression:** the **logistic model** (or **logit model**) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick.
- **Naive Bayes:** it calculates probability of certain class. Model will be generated using pickle and stored
- **Random Forest:** Random forest avoids over fitting problem and model will be generated, stored into pickle.

All this machine learning model will generated and its

i. precision
ii. recall
iii. f1 score
iv. Accuracy will be calculated.

- **Clustering:-**

Dbscan used for outlier's detection

Outliers are specific entries in dataset that are different than other point and don't play vitalrole in classification.

In statistics, an **outlier** is an observation point that is distant from other observations.

In this domain name will be clustered based on:

i. Cryptolockereg. nxgbdtnvrfker.ru
ii. TOVAReg.:- gppwkpxyremp.net
iii. Dyreeg:- q2aa41a5b31294e5e6f28d1adcf48a54b.tk
iv. normalDomaineg:- easypdfcombine.com

## 3.5 System  Requirements

### Software Requirements

- Spyder
- Python 3

### Hardware Requirements

- Minimum RAM required: 4GB (Suggested: 8GB)
- Minimum Free Disk Space: 25GB
- Minimum Processor i3 or above
- Operating System of 64bit

# CHAPTER 4

# WORKING OF THE SYSTEM

In this section, we explain the working of our proposed system illustrated with the help of screenshots shown below.

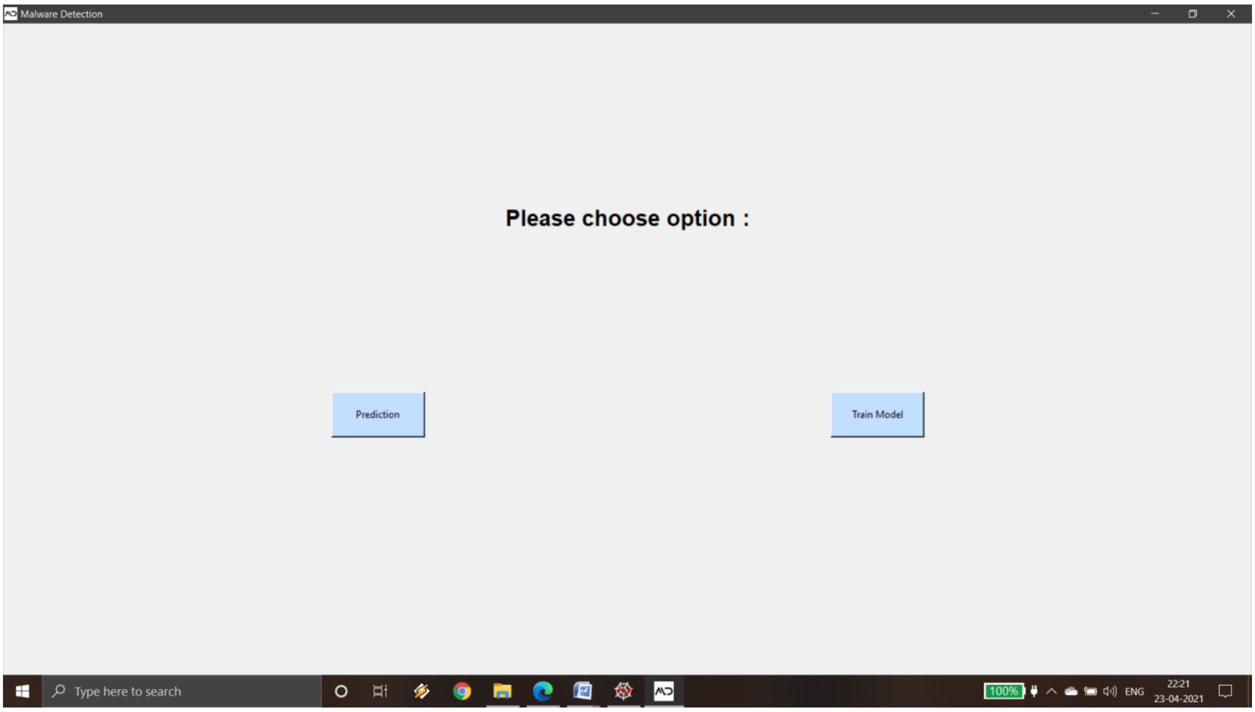


**Figure 4.1:** Basic System Application

## 4.1 Beginning with basic system application

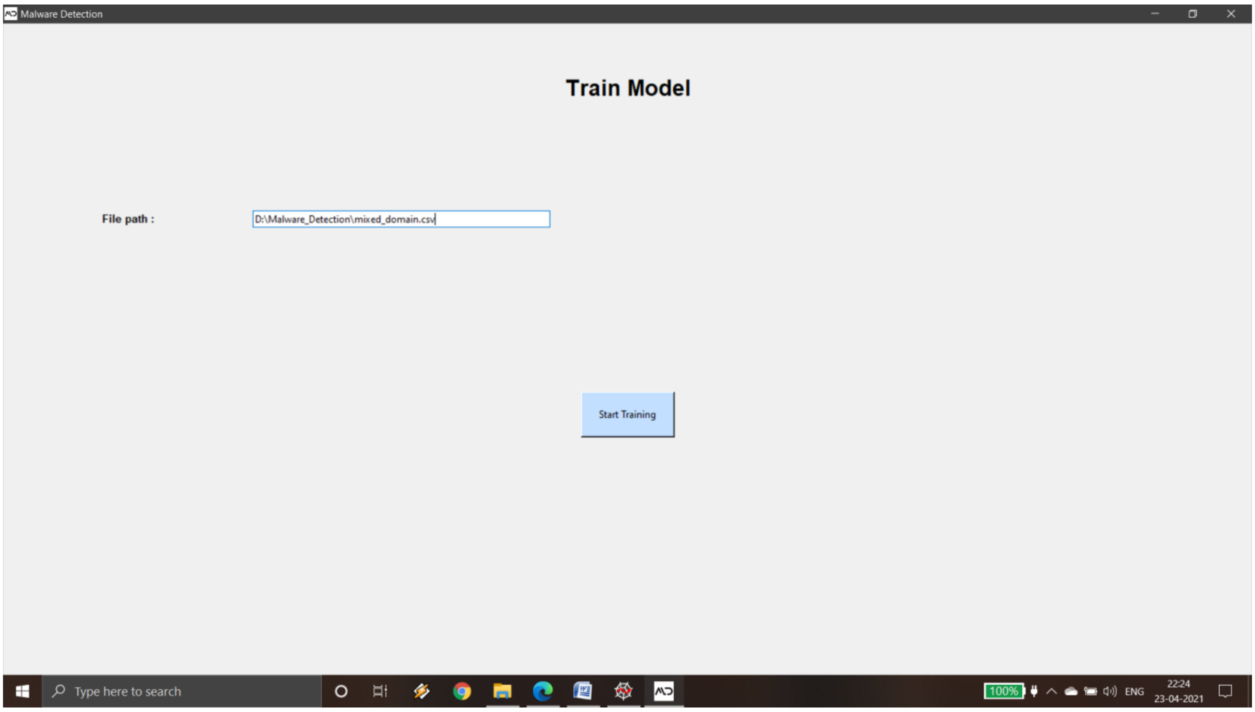
It is a basic system application made with the help of Tkinter GUI.

Here there are two buttons where the user can choose to train the model and get the confusion matrix and classification report or to predict the domain(s) whether they contain malware or not.

## 4.2 Training the model

By clicking on the “Train Model” button the user is directed towards the next page of the application where the user can input the dataset file’s path and start the prediction.

Artificial Neural networks (ANN) or neural networks are computational algorithms. It intended to simulate the behavior of biological systems composed of “neurons”.  A neural network is a machine learning algorithm based on the model of a human neuron. The ANN algorithm after comparing with the other mentioned algorithms showed the best accuracy and so we used it for the UI.



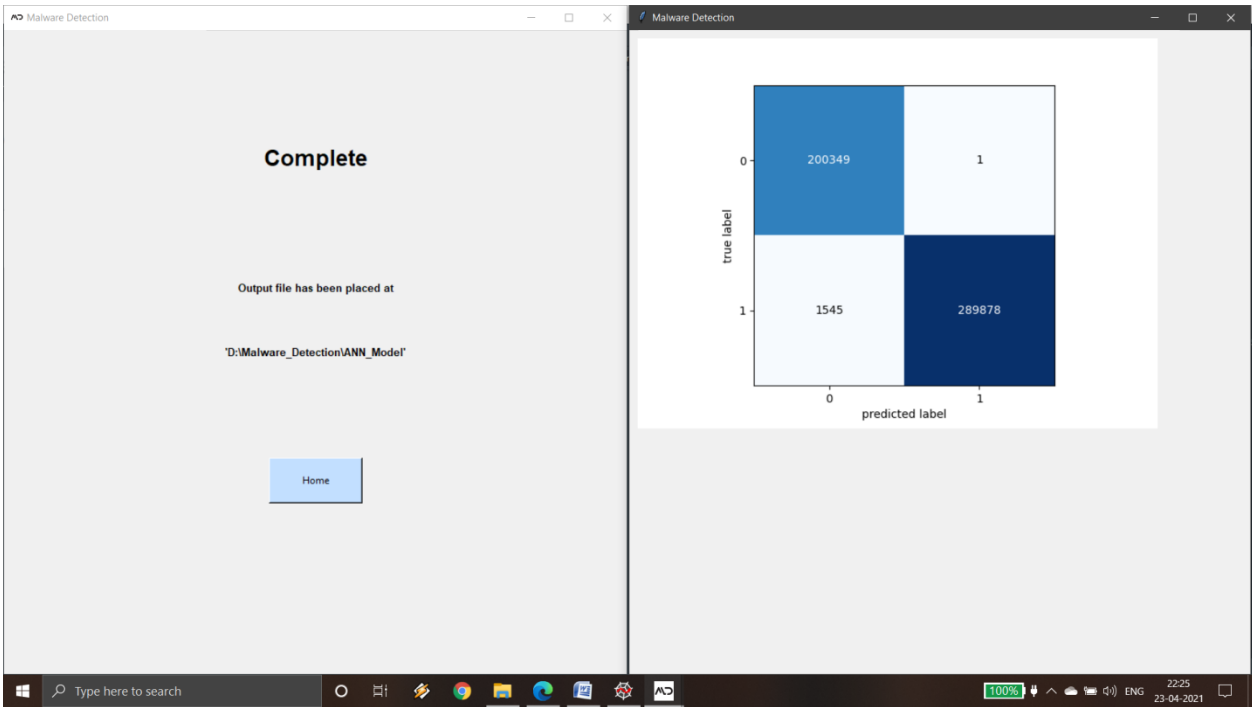
Train Model

File path : D:\Malware_Detection\mixed_domain.csv

**Figure 4.2:** Training the model.

## 4.3 Plotting the confusion matrix

The classification reports are present in the logs section of Spyder and the confusion matrix is obtained in the system application. Also the whole model gets store in a sub folder where the main program file is present.



Complete

Output file has been placed at

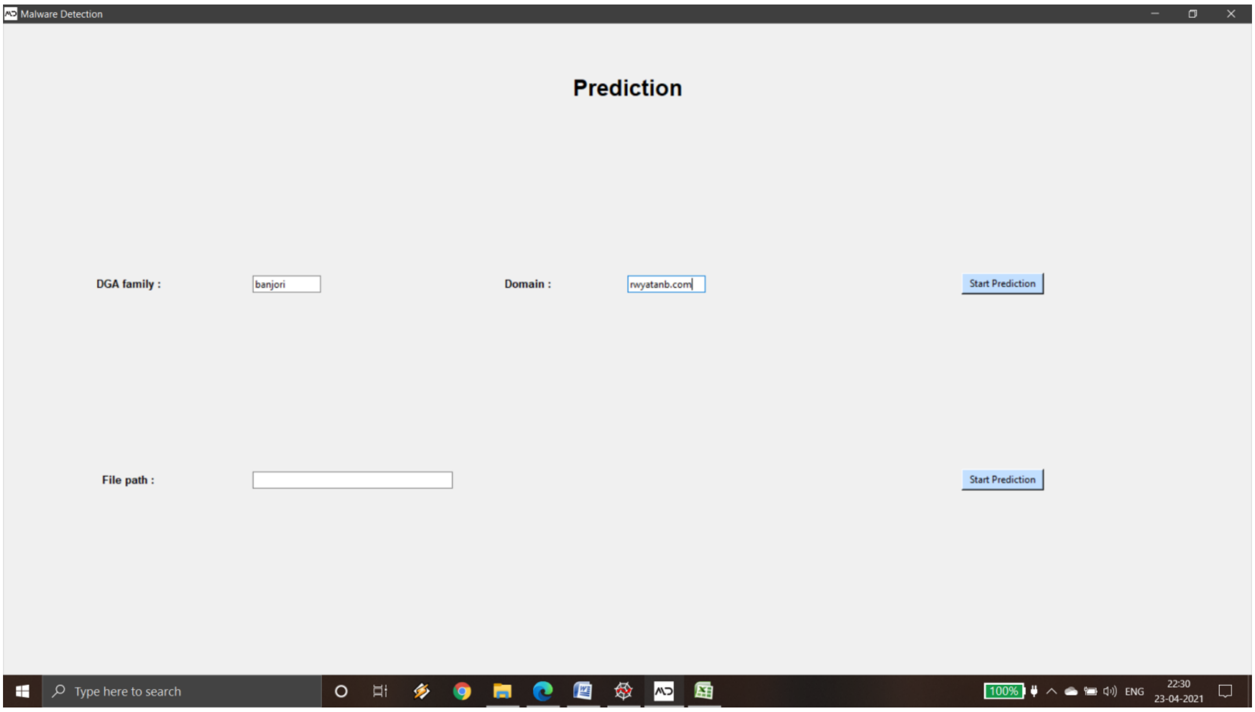
'D:\Malware_Detection\ANN_Model'

**Figure 4.3:** Confusion Matrix

## 4.4 First test prediction for a single domain

If the user wants to predict a single domain, the family of the domain (if any) should be entered in the textbox aside “DGA Family” label.

After that enter the domain in the textbox aside the “Domain” label and start the prediction.



Prediction

DGA family : banjori

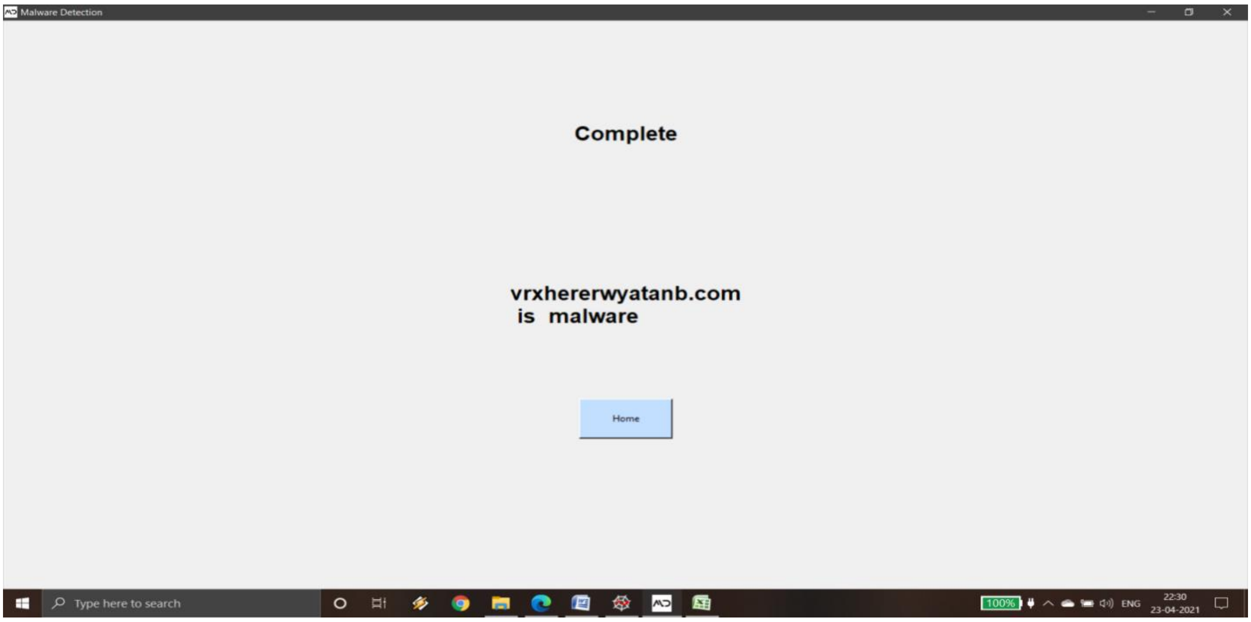
Domain : rwyatanb.com

File path :

**Figure 4.4:** First Test Prediction for a single domain.

## 4.5 Result for first test single domain prediction.

Here the first test shows that the domain contains malware.

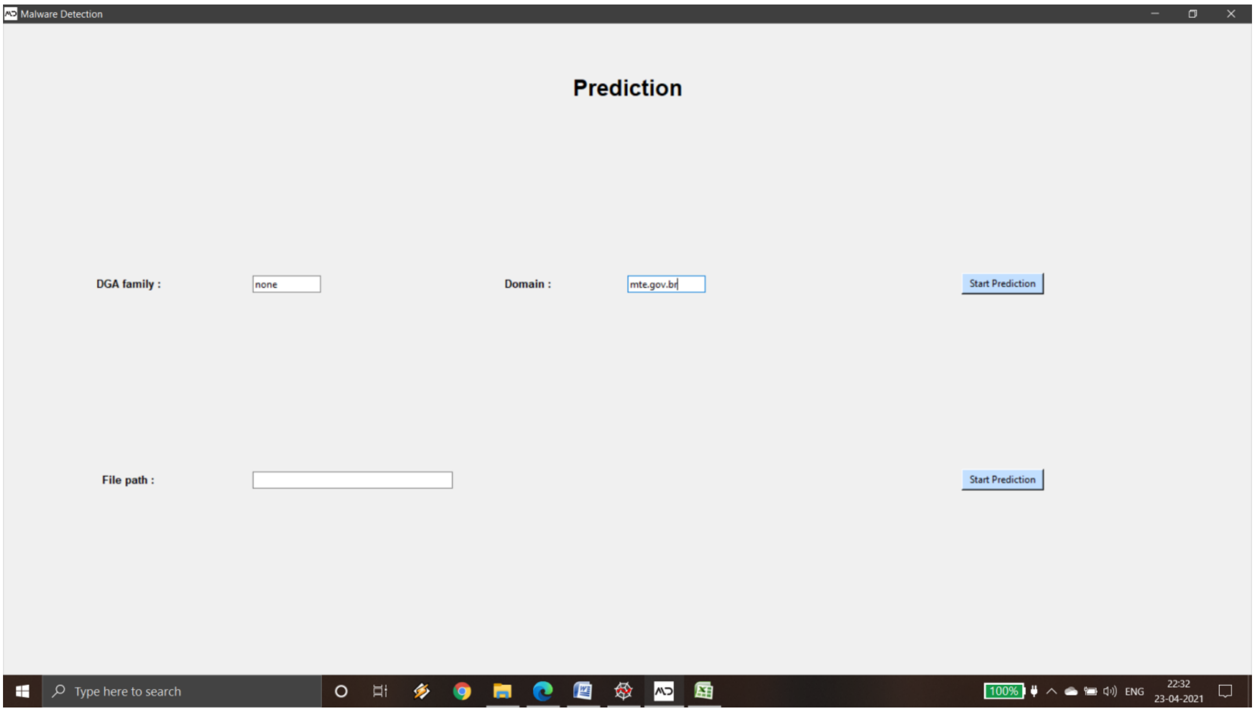


Complete

vrxhererwyatanb.com
is malware

**Figure 4.5:** First test result

## 4.6 Second test for a single domain prediction



Prediction

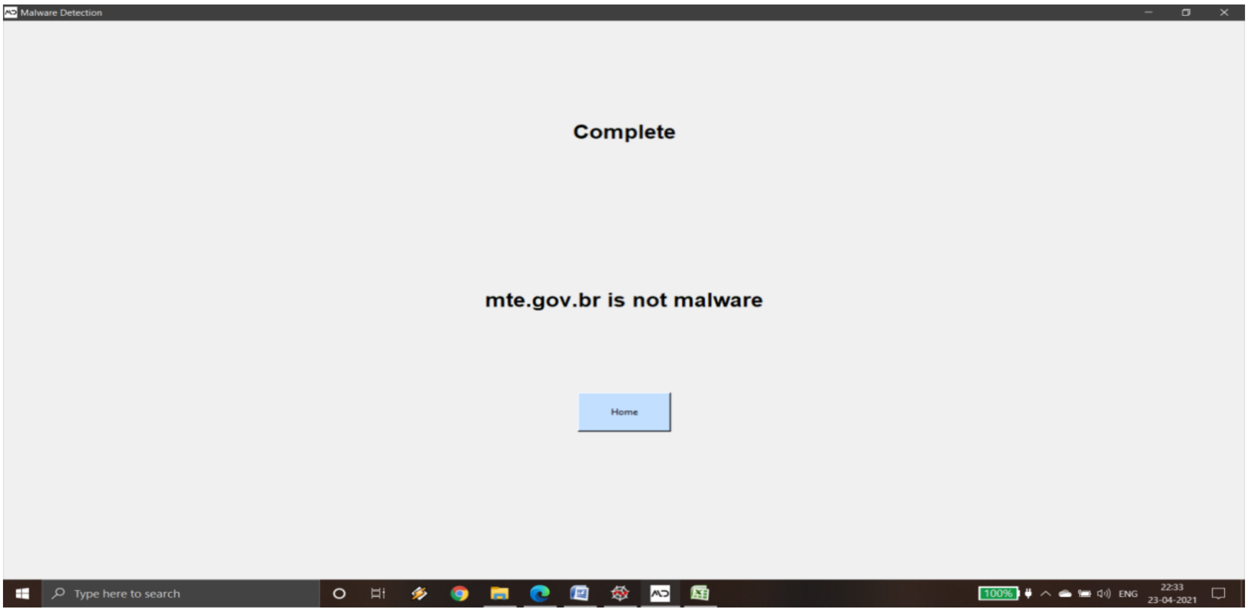
DGA family : none

Domain : mte.gov.br

File path :

**Figure 4.6:** Second test for a single domain prediction.

## 4.7 Result for second test for a single domain prediction.



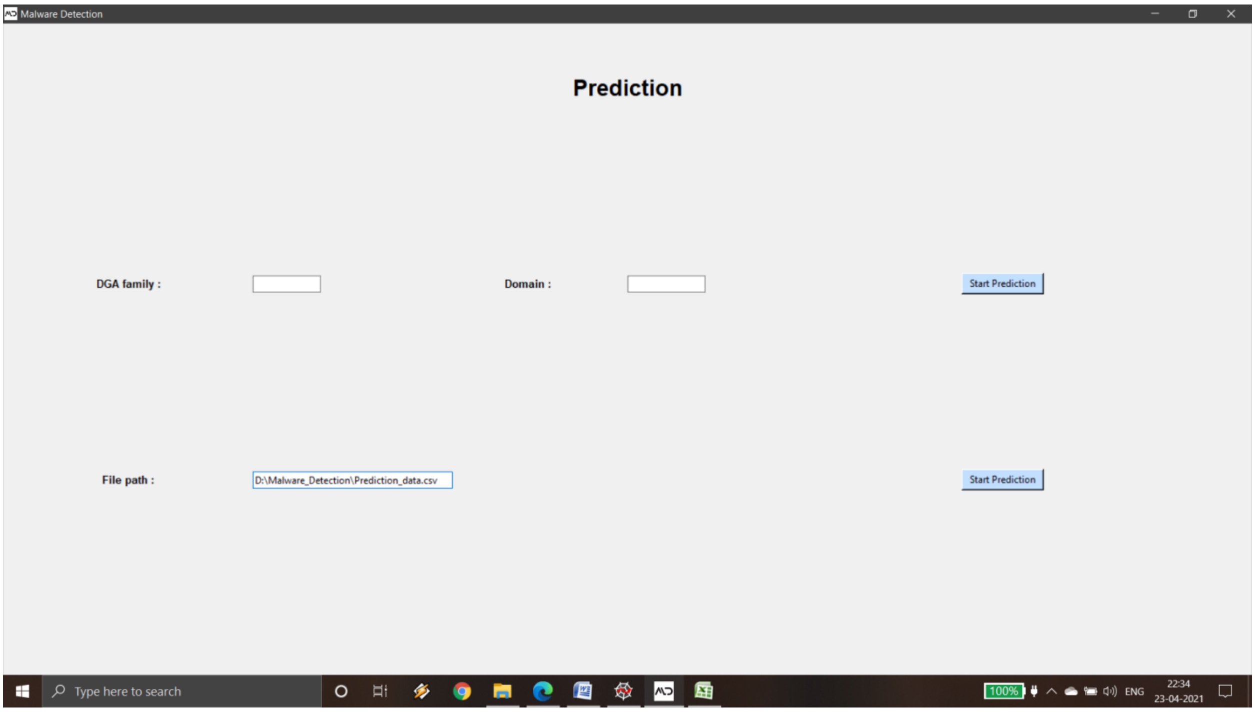
Complete

mte.gov.br is not malware

**Figure 4.7:** Second test result

## 4.8 Prediction for a list of domains

Here if the user wants to predict a list of domains, the user can input the file's path containing the list of domains in the textbox aside "File path" label and start the prediction by clicking the "Start Prediction" aside the textbox.



File path : D:\Malware_Detection\Prediction_data.csv

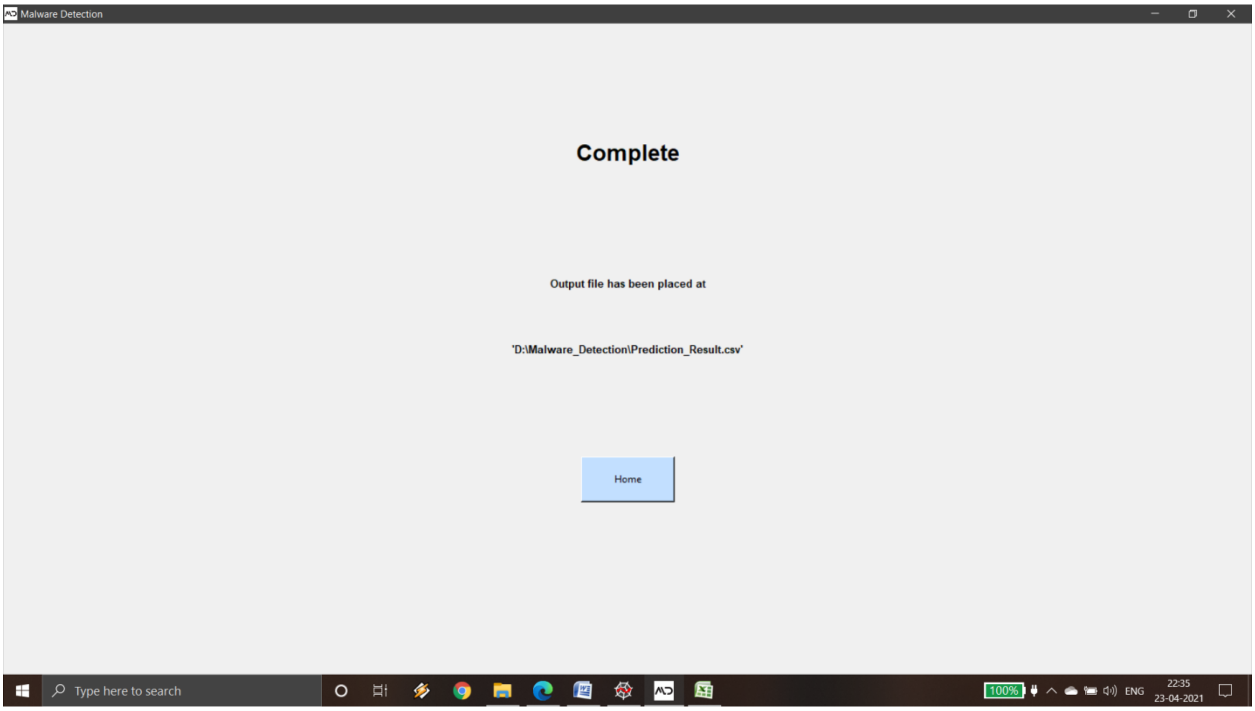
**Figure 4.8:** Prediction for a list of domains

## 4.9 Result for a list of domains

Here the user would get a text message that the result of the prediction for the given list of domains.

The message would contain the file's path created by the system in the folder where the main program is stored.

The prediction of the list would be stored in a CSV file which the user could open and fetch the results.



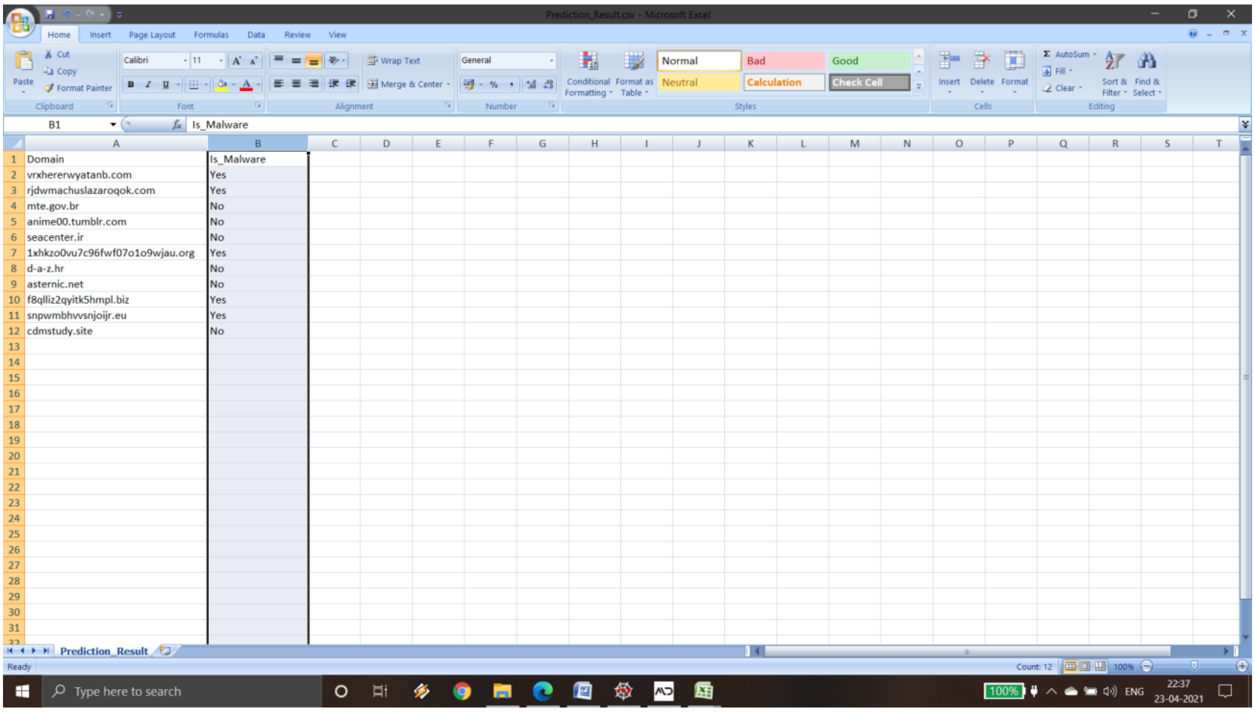
Complete

Output file has been placed at

'D:\Malware_Detection\Prediction_Result.csv'

Home

**Figure 4.9:** File location of result for a list of domains



B1 Is_Malware

| | A | B |
|---|---|---|
| 1 | Domain | Is_Malware |
| 2 | vrxhererwyatanb.com | Yes |
| 3 | rjdwmachuslazaroqok.com | Yes |
| 4 | mte.gov.br | No |
| 5 | anime00.tumblr.com | No |
| 6 | seacenter.ir | No |
| 7 | 1xhkzo0vu7c96fwf07o1o9wjau.org | Yes |
| 8 | d-a-z.hr | No |
| 9 | asternic.net | No |
| 10 | f8qlliz2qyitk5hmpl.biz | Yes |
| 11 | snpwmbhvvsnjoijr.eu | Yes |
| 12 | cdmstudy.site | No |

**Figure 4.10:** CSV file of result for a list of domains.

# CHAPTER 5

# CONCLUSION & FUTURE WORK

## 5.1 Conclusion

Detecting DGAs is a grand challenge in security areas. Blacklisting is good for handling static methods. However, DGAs are usually used by an attacker to communicate with variety of servers. They are dynamic, so simply using the blacklisting is not sufficient for detecting a DGA. In this research, we have proposed the machine learning framework with the development of a deep learning model to handle DGA threats. The proposed machine learning framework consists of a dynamic blacklist, a feature extractor, a two level machine learning model for classification and clustering, and a prediction model.

We have collected a real-time threat intelligence feed over a one-year period where all domains live threats on the Internet. As the size of the data we collected becomes larger and larger, we have built a deep learning model to perform the classification, which has a better performance than the machine learning algorithms. Based on our extensive experiments on the real-world feed, we have shown that the proposed framework can effectively extract domain name features as well as classify, cluster and detect domain names. We have further used ANN model to improve our classification.

## 5.2 Future Work

In the future, we will further explore deep learning algorithms for domain name clustering and predictions for this research and evaluate them on a real-world. The most common method to detect malicious URLs deployed by many antivirus groups is the blacklist method. Blacklists are essentially a database of URLs that have been confirmed to be malicious inthe past. Scope of this project is useful for it helps to prevent malicious activities in cyber world. It is intended to improve the system performance on the based on dataset. Also use new techniques to get accurate result.

# REFERENCES

[1] Yi Li, Kaiqi Xiong, Tommy Chin, Chengbin Hu, “A Machine Learning Framework for Domain Generation Algorithm-Based Malware Detection” IEEE Access (Volume: 7), 31 January, 2019

[2] T. Chin, K. Xiong and M. Rahouti, "SDN-based kernel modular countermeasure for intrusion detection", Proc. 13rd EAI Int. Conf. Secur. Privacy Commun. Netw., pp. 270- 290, September, 2019.

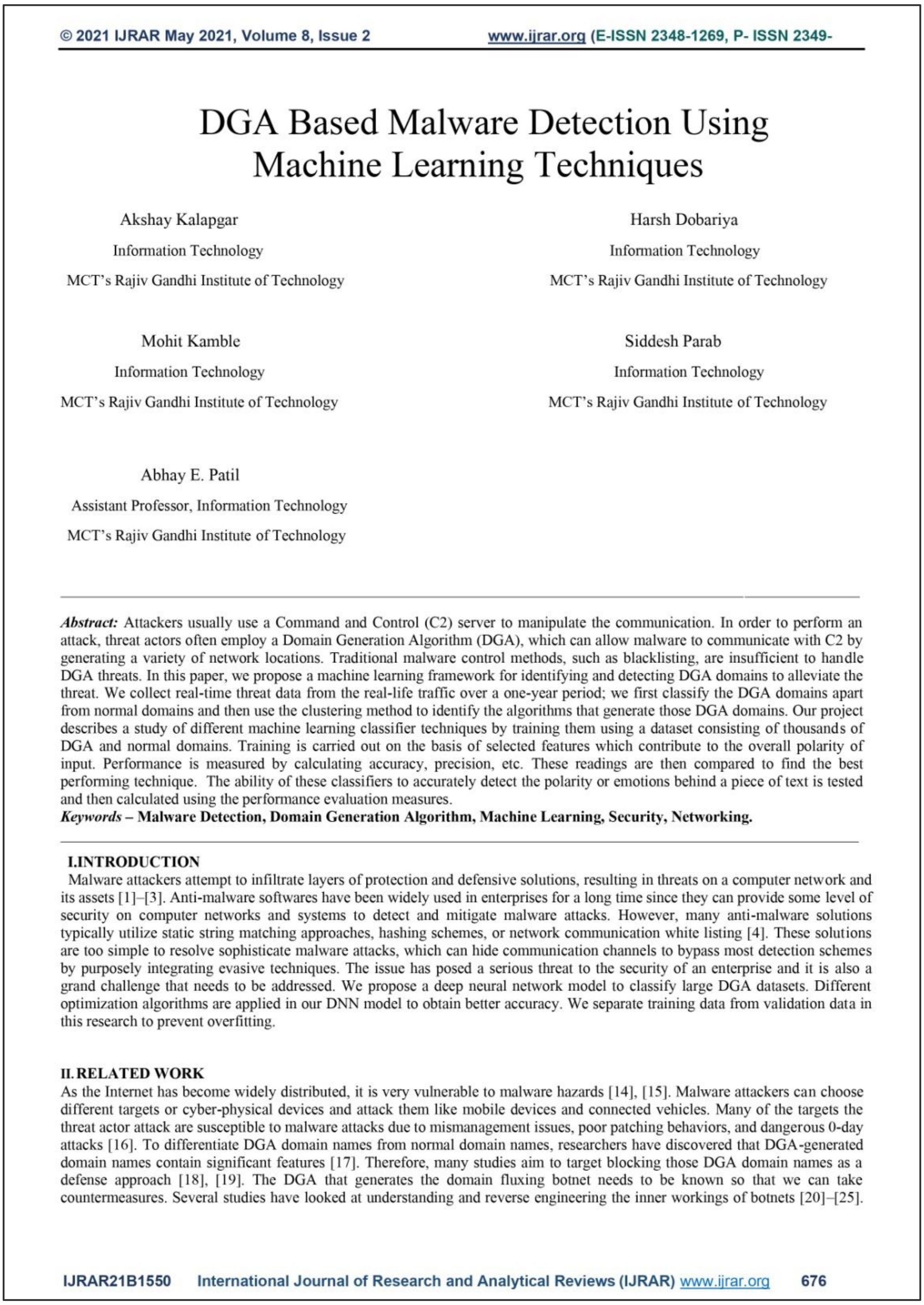
[3] S.Mammadli, December 2016, “An SDN based framework for guaranteeing security and performance in information-centric cloud networks,” Procedia Computer Science, pp.495-499.

[4] Xiaojie and D.Huailin, W.Qingfeng, July 2009, “A two-hashing table multiple string pattern matching algorithm,” Tenth International Conference on Information Technology: New Generations (ITNG), IEEE, January 2013.

[5] Jong Young Lee, Jun Young Chang and Eul Gyu Im September 2019, “DNS analysis based malware detection system,” Knowledge Engineering and Applications (ICKEA), IEEE International Conference on IEEE , pp. 193-196.

# PUBLICATION AND CERTIFICATES

[1] Akshay Kalapgar, Harsh Dobariya, Mohit Kamble, SiddeshParab, Abhay E. Patil, "DGA BASED MALWARE DETECTION USING MACHINE LEARNING TECHNIQUES", IJRAR - International Journal of Research and Analytical Reviews (IJRAR), E-ISSN 2348-1269, P- ISSN 2349-5138, Volume.8, Issue 2, Page No pp.676-681, May 2021, Available here http://www.ijrar.org/IJRAR21B1550.pdf



## DGA Based Malware Detection Using Machine Learning Techniques

Akshay Kalapgar
Information Technology
MCT's Rajiv Gandhi Institute of Technology

Harsh Dobariya
Information Technology
MCT's Rajiv Gandhi Institute of Technology

Mohit Kamble
Information Technology
MCT's Rajiv Gandhi Institute of Technology

Siddesh Parab
Information Technology
MCT's Rajiv Gandhi Institute of Technology

Abhay E. Patil
Assistant Professor, Information Technology
MCT's Rajiv Gandhi Institute of Technology

***Abstract:*** Attackers usually use a Command and Control (C2) server to manipulate the communication. In order to perform an attack, threat actors often employ a Domain Generation Algorithm (DGA), which can allow malware to communicate with C2 by generating a variety of network locations. Traditional malware control methods, such as blacklisting, are insufficient to handle DGA threats. In this paper, we propose a machine learning framework for identifying and detecting DGA domains to alleviate the threat. We collect real-time threat data from the real-life traffic over a one-year period; we first classify the DGA domains apart from normal domains and then use the clustering method to identify the algorithms that generate those DGA domains. Our project describes a study of different machine learning classifier techniques by training them using a dataset consisting of thousands of DGA and normal domains. Training is carried out on the basis of selected features which contribute to the overall polarity of input. Performance is measured by calculating accuracy, precision, etc. These readings are then compared to find the best performing technique. The ability of these classifiers to accurately detect the polarity or emotions behind a piece of text is tested and then calculated using the performance evaluation measures.

***Keywords*** **– Malware Detection, Domain Generation Algorithm, Machine Learning, Security, Networking.**

### I.INTRODUCTION

Malware attackers attempt to infiltrate layers of protection and defensive solutions, resulting in threats on a computer network and its assets [1]–[3]. Anti-malware softwares have been widely used in enterprises for a long time since they can provide some level of security on computer networks and systems to detect and mitigate malware attacks. However, many anti-malware solutions typically utilize static string matching approaches, hashing schemes, or network communication white listing [4]. These solutions are too simple to resolve sophisticate malware attacks, which can hide communication channels to bypass most detection schemes by purposely integrating evasive techniques. The issue has posed a serious threat to the security of an enterprise and it is also a grand challenge that needs to be addressed. We propose a deep neural network model to classify large DGA datasets. Different optimization algorithms are applied in our DNN model to obtain better accuracy. We separate training data from validation data in this research to prevent overfitting.

### II. RELATED WORK

As the Internet has become widely distributed, it is very vulnerable to malware hazards [14], [15]. Malware attackers can choose different targets or cyber-physical devices and attack them like mobile devices and connected vehicles. Many of the targets the threat actor attack are susceptible to malware attacks due to mismanagement issues, poor patching behaviors, and dangerous 0-day attacks [16]. To differentiate DGA domain names from normal domain names, researchers have discovered that DGA-generated domain names contain significant features [17]. Therefore, many studies aim to target blocking those DGA domain names as a defense approach [18], [19]. The DGA that generates the domain fluxing botnet needs to be known so that we can take countermeasures. Several studies have looked at understanding and reverse engineering the inner workings of botnets [20]–[25].

Since the DGA domain names are usually randomly generated, the lengths of DGA domains are very long. Such a feature can be used to detect DGA domains.

### III. PROPOSED SYSTEM

In our proposed system, Domain extracted from DGAs. Machine learning framework that encompasses multiple feature extraction techniques and the models to classify the DGA domains from normal domains, cluster the DGA domains, and predict a DGA domain. A deep learning model to handle large datasets multiple on-line sources from simple Google searching provide example codes for a DGA construction. Online threat intelligence feeds give an approach to examining current and live threats in real-world environment. Using real-time active malicious domains derived from DGAs on the public internet measures the accuracy of the proposed approach.

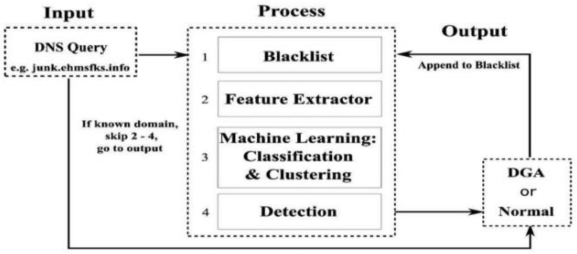


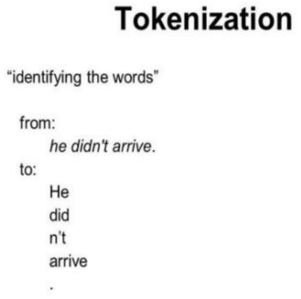
Fig 1 Proposed System

### IV. IMPLEMENTATION

In this section, the detailed steps of how the works if being implemented are elaborated by stating the various algorithms being used. The main steps involve pre-processing, feature extraction, classification.

(1) Pre-processing: The set of domains has to undergo several pre-processing techniques to ensure the highest efficiency in thelater stages. It is important to ensure that the data has structural integrity. We start by performing tokenization, Stop-words removalis also an important step. This is followed by performing lemmatization and stemming. Finally, we obtain training and testing datasets by splitting the main dataset.

(A) Tokenization: Tokenization splits a review into meaningful and smaller parts to help reduce the overall complexity. Data cleaning removes any detected errors and outliers in the domains such as spelling mistakes. This is necessary to avoid any confusion during feature extraction.

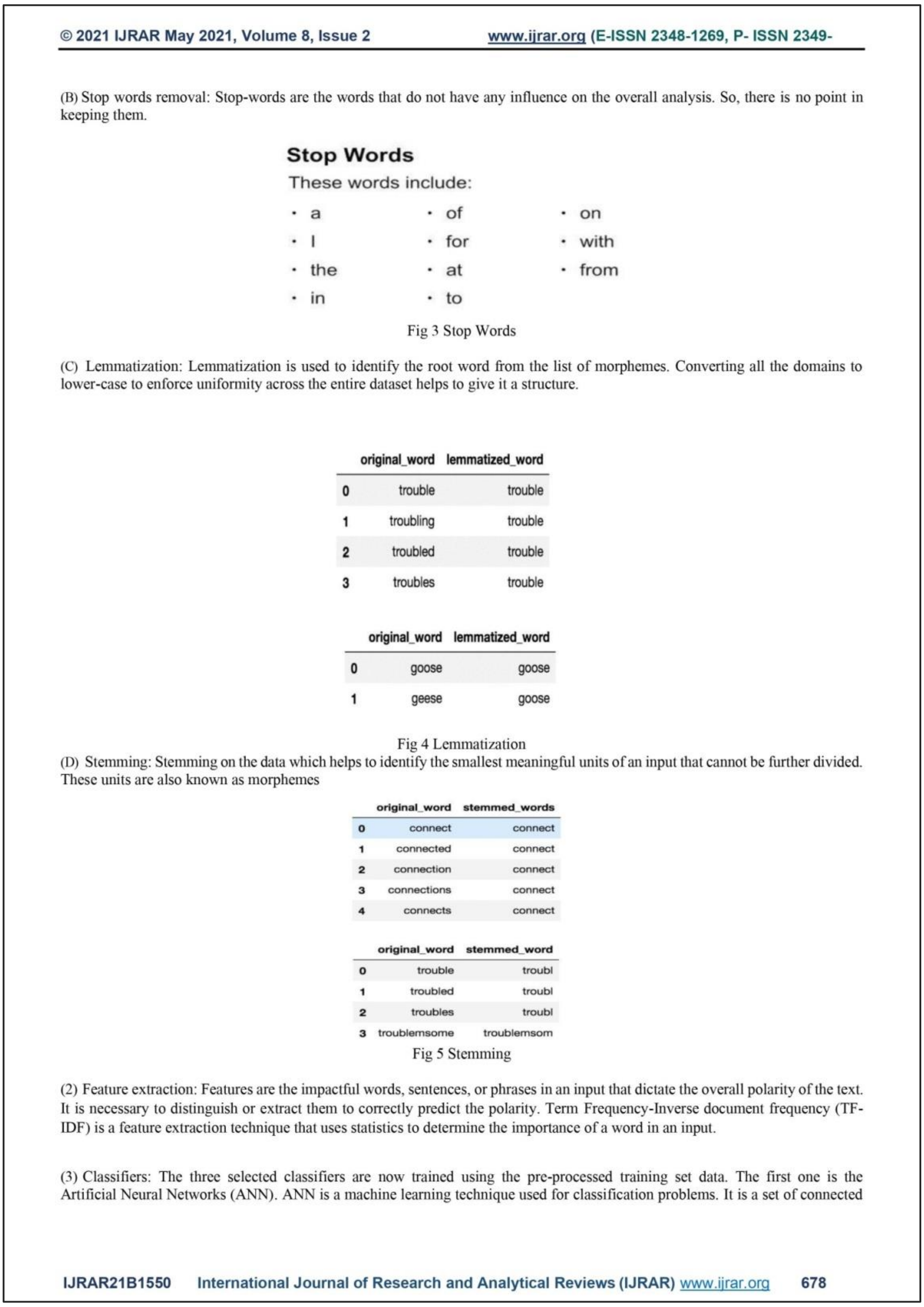


Tokenization

"identifying the words"

from:
*he didn't arrive.*
to:
He
did
n't
arrive
.

Fig 2 Tokenization



(B) Stop words removal: Stop-words are the words that do not have any influence on the overall analysis. So, there is no point in keeping them.

**Stop Words**

These words include:

- a
- I
- the
- in
- of
- for
- at
- to
- on
- with
- from

Fig 3 Stop Words

(C) Lemmatization: Lemmatization is used to identify the root word from the list of morphemes. Converting all the domains to lower-case to enforce uniformity across the entire dataset helps to give it a structure.

| | original_word | lemmatized_word |
|---|---|---|
| 0 | trouble | trouble |
| 1 | troubling | trouble |
| 2 | troubled | trouble |
| 3 | troubles | trouble |

| | original_word | lemmatized_word |
|---|---|---|
| 0 | goose | goose |
| 1 | geese | goose |

Fig 4 Lemmatization

(D) Stemming: Stemming on the data which helps to identify the smallest meaningful units of an input that cannot be further divided. These units are also known as morphemes

| | original_word | stemmed_words |
|---|---|---|
| 0 | connect | connect |
| 1 | connected | connect |
| 2 | connection | connect |
| 3 | connections | connect |
| 4 | connects | connect |

| | original_word | stemmed_word |
|---|---|---|
| 0 | trouble | troubl |
| 1 | troubled | troubl |
| 2 | troubles | troubl |
| 3 | troublemsome | troublemsom |

Fig 5 Stemming

(2) Feature extraction: Features are the impactful words, sentences, or phrases in an input that dictate the overall polarity of the text. It is necessary to distinguish or extract them to correctly predict the polarity. Term Frequency-Inverse document frequency (TF-IDF) is a feature extraction technique that uses statistics to determine the importance of a word in an input.

(3) Classifiers: The three selected classifiers are now trained using the pre-processed training set data. The first one is the Artificial Neural Networks (ANN). ANN is a machine learning technique used for classification problems. It is a set of connected

Input output network in which weight is associated with each connection. It consists of one input layer, one or more intermediate layer and one output layer. Learning of neural network is performed by adjusting the weight of connection. By updating the weight iteratively performance of network is improved. The next classifier is Naïve Bayes, based on Bayes theorem. It is a probabilistic model that assumes that all the features are independent of each other. Its simplistic nature enables it to be efficient even as the sizeof the data increases. Random forest classifier, as the name suggests, consists of a large number of decision trees. These trees are capable of predicting the polarity of a review independently. Their predictions can then be combined the output with the highest occurrence is concluded to be the final output.

## V. RESULTS

We observed the accuracy and precision of the classifiers by testing them twice on the same unclassified dataset with differing sizes. Accuracy is a performance evaluation measure that displays the percentage of inputs that were labeled correctly from the total number of inputs. Precision measures the ratio of true positives to the total number of positives labeled by the system.

For the first test with 5,000 domains, Random Forest provided the highest accuracy at 86.2%. It was closely followed by ANN at 84.5% and Naïve Bayes at 81.4%.

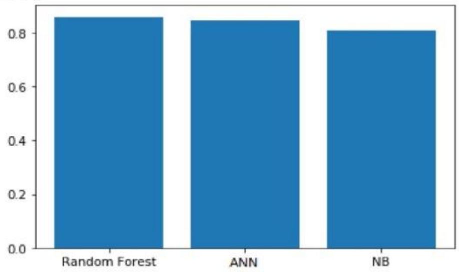


Fig. 6 Accuracy of test-1

The precision scores followed the same pattern with Random Forest leading with a precision of 77.1%, with ANN at 73.3% and Naïve Bayes at 71%.

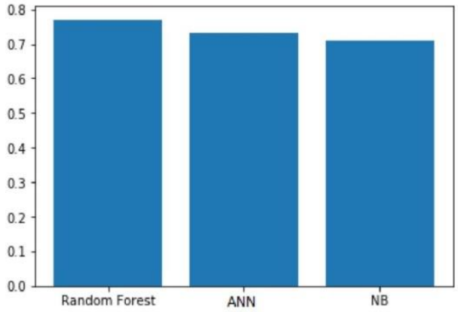


Fig. 7 Precision of test-1

For the second test using the same dataset but with only 50,000 domains, ANN yielded the highest accuracy at 81.5%, with Random Forest providing a yield of 80.1% and Naïve Bayes of 76.7%.

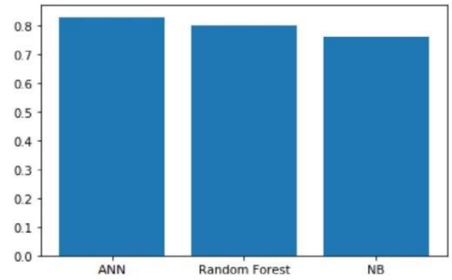


Fig. 8 Accuracy of test-2

For precision, ANN again led with a score of 72.5%. However, unlike accuracy, Naïve Bayes recorded a higher precision than Random Forest. Naïve Bayes gave 69% and Random Forest gave 67.8%.

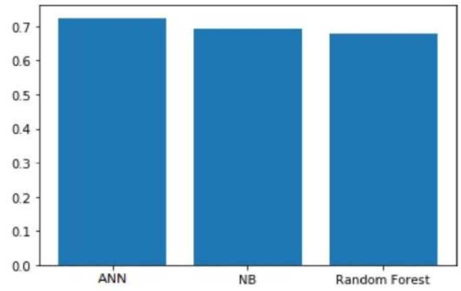


Fig. 9 Precision of test-2

| | Random Forest | | Artificial Neural Networks | | Naïve Bayes | |
|---|---|---|---|---|---|---|
| | Test-1 | Test-2 | Test-1 | Test-2 | Test-1 | Test-2 |
| Accuracy | 86.2% | 80.1% | 84.5% | 83% | 81.4% | 76.7% |
| Precision | 77.1% | 67.8% | 73.3% | 72.5% | 71% | 69% |

Table 1

## VI. CONCLUSION AND FUTURE WORK

Domain generation algorithm helps us to determine the threats of the malware behind a piece of text. Our work consisted of training three different classifiers and comparing their test results. The dataset which was used for training and testing was a compilation of DGA domains and normal domains. The tests showed that the performance changed as the number of domains changed. For a higher number of domains, Random Forest gave the best performance. For the lower number of domains, ANN proved to be the best performer. In the future, even more, classifiers can be tested and the number of performance evaluation measures used can be increased. Additionally, performance can be further increased by ensembling several classifiers with the highest yields.

## VII. ACKNOWLEDGEMENTS

We would like to thank MCT's Rajiv Gandhi Institute of Technology for providing research facilities and to Prof. Abhay E Patil (Information Technology) department for their support and encouragement.

## REFERENCES

[1] K. Rieck, T. Holz, C. Willems, P. Düssel, and P. Laskov, "Learning and classification of malware behavior" in International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, 2008, pp. 108–125.

[2] T. Chin, K. Xiong, and M. Rahouti, "SDN-based kernel modular countermeasure for intrusion detection," in Proceedings of 13rd EAI International Conference on Security and Privacy in Communication Networks. Springer, 2017.

[3] U. Ghosh, P. Chatterjee, D. Tosh, S. Shetty, K. Xiong, and C. Kamhoua, "An SDN based framework for guaranteeing security and performance in information-centric cloud networks," in Proceedings of the 11th IEEE International Conference on Cloud Computing (IEEE Cloud), 2017.

[4] C. Khancome, V. Boonjing, and P. Chanvarasuth, "A two-hashing table multiple string pattern matching algorithm," in Tenth International Conference on Information Technology: New Generations (ITNG). IEEE, 2013, pp. 696–701.

[5] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, "Phoenix: DGA-based botnet tracking and intelligence," in International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, 2014, pp. 192–211.

[6] A. K. Sood and S. Zeadally, "A taxonomy of domain-generation algorithms," IEEE Security & Privacy, vol. 14, no. 4, pp. 46–53, 2016.

[7] K. Xiong, "Multiple priority customer service guarantees in cluster computing," in Proceedings of the IEEE International Symposium on Parallel & Distributed Processing (IPDPS). IEEE, 2009, pp. 1–12.

[8] Xiong, "Resource optimization and security for cloud services." WileyISTE, 2014.

[9] T. Chin, K. Xiong, C. Hu, and Y. Li, "A machine learning framework for studying domain generation algorithm (dga)-based malware," in SecureComm, 2018.

[10] K. Xiong and X. Chen, "Ensuring cloud service guarantees via service level agreement (sla)-based resource allocation," in Proceedings of the IEEE 35th International Conference on Distributed Computing Systems Workshops, ICDCS Workshops. IEEE, 2015, pp. 35–41. K. Sornalakshmi, "Detection of DoS attack and zero day threat with siem," in International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2017, pp. 1–7.

[11] S. Yadav and A. N. Reddy, "Winning with DNS failures: Strategies for faster botnet detection," in International Conference on Security and Privacy in Communication Systems. Springer, 2011, pp. 446–459.

[12] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan, "Detecting algorithmically generated domain-flux attacks with DNS traffic analysis," IEEE/Acm Transactions on Networking, vol. 20, no. 5, pp. 1663–1677, 2012.

[13] F. Guo, P. Ferrie, and T.-C. Chiueh, "A study of the packer problem and its solutions," in International Workshop on Recent Advances in Intrusion Detection. Springer, 2008, pp. 98–115.

[14] T. Holz, M. Steiner, F. Dahl, E. Biersack, F. C. Freiling et al., "Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm." LEET, vol. 8, no. 1, pp. 1–9, 2008.

[15] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," IEEE Security & Privacy, vol. 9, no. 3, pp. 49–51, 2011. [22] J. Stewart, "Inside the storm: Protocols and encryption of the storm botnet," 2009.

[16] H. S. Phillip Porras and V. Yegneswaran, "Conficker C P2P protocol and implementation," 2009.

[17] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: analysis of a botnet takeover," in Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009, pp. 635–647.

[18] L. Zhang, S. Yu, D. Wu, and P. Watters, "A survey on latest botnet attack and defense," in IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2011, pp. 53–60.

[19] T. Barabosch, A. Wichmann, F. Leder, and E. Gerhards-Padilla, "Automatic extraction of domain name generation algorithms from current malware," in Procceedings of NATO Symposium IST-111 on Information Assurance and Cyber Defense, Koblenz, Germany, 2012.

[20] J. Gardiner and S. Nagaraja, "On the security of machine learning in malware c&c detection: A survey," ACM Computing Surveys (CSUR), vol. 49, no. 3, p. 59, 2016.

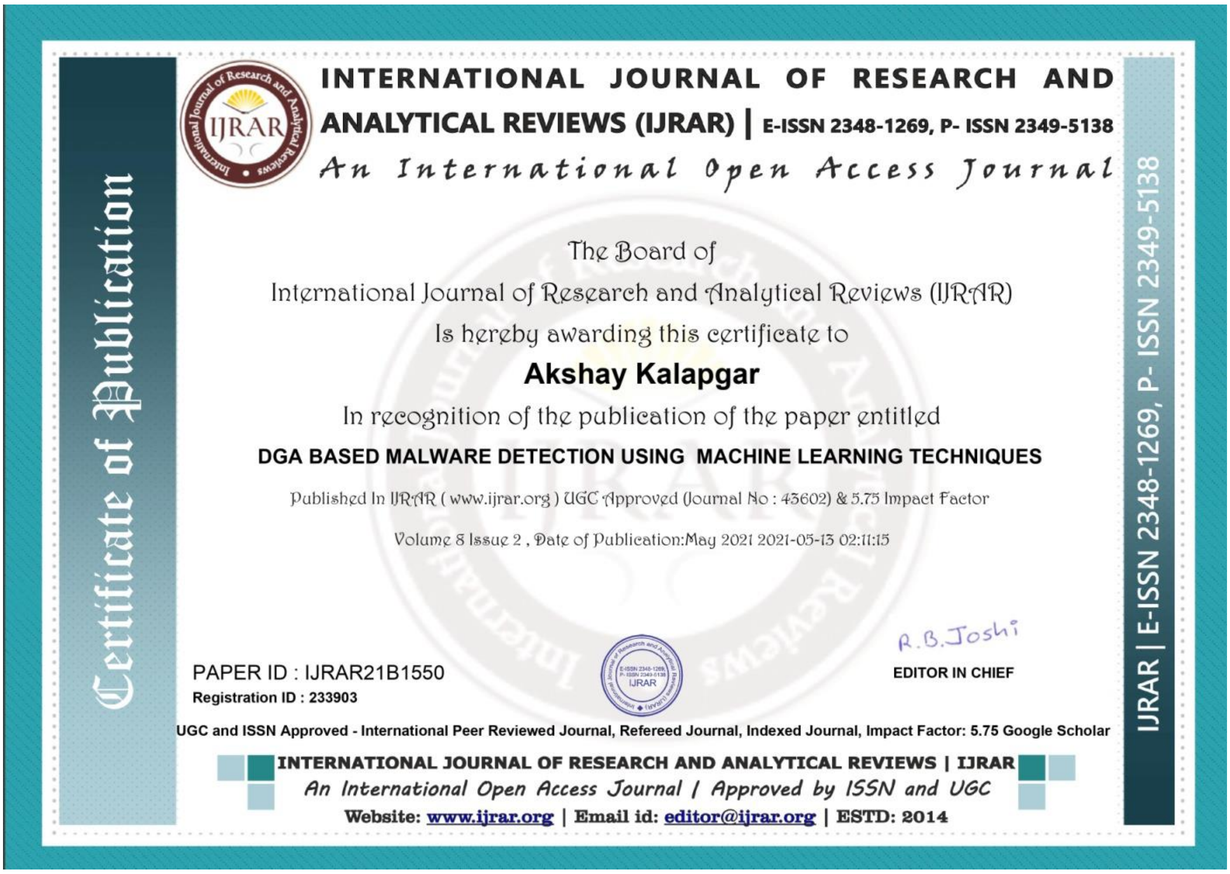
[21] M. Mowbray and J. Hagen, "Finding domain-generation algorithms by looking at length distribution," in IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). IEEE, 2014, pp. 395–400.

[22] A. Ahluwalia, I. Traore, K. Ganame, and N. Agarwal, "Detecting broad length algorithmically generated domains," in International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. Springer, 2017, pp. 19–34.

[23] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious URLs," in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009, pp. 1245–1254.

[24] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From throw-away traffic to Bots: Detecting the rise of DGA-based malware." in USENIX security symposium, vol. 12, 2012.

[25] W. Wang and K. Shirley, "Breaking bad: Detecting malicious domains using word segmentation," arXiv preprint arXiv:1506.04111, 2015.



Certificate of Publication

**INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | E-ISSN 2348-1269, P- ISSN 2349-5138**

*An International Open Access Journal*

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

**Akshay Kalapgar**

In recognition of the publication of the paper entitled

**DGA BASED MALWARE DETECTION USING MACHINE LEARNING TECHNIQUES**

Published In IJRAR ( www.ijrar.org ) UGC Approved (Journal No : 43602) & 5.75 Impact Factor

Volume 8 Issue 2 , Date of Publication:May 2021 2021-05-13 02:11:15

PAPER ID : IJRAR21B1550

**Registration ID : 233903**

R.B.Joshi

**EDITOR IN CHIEF**

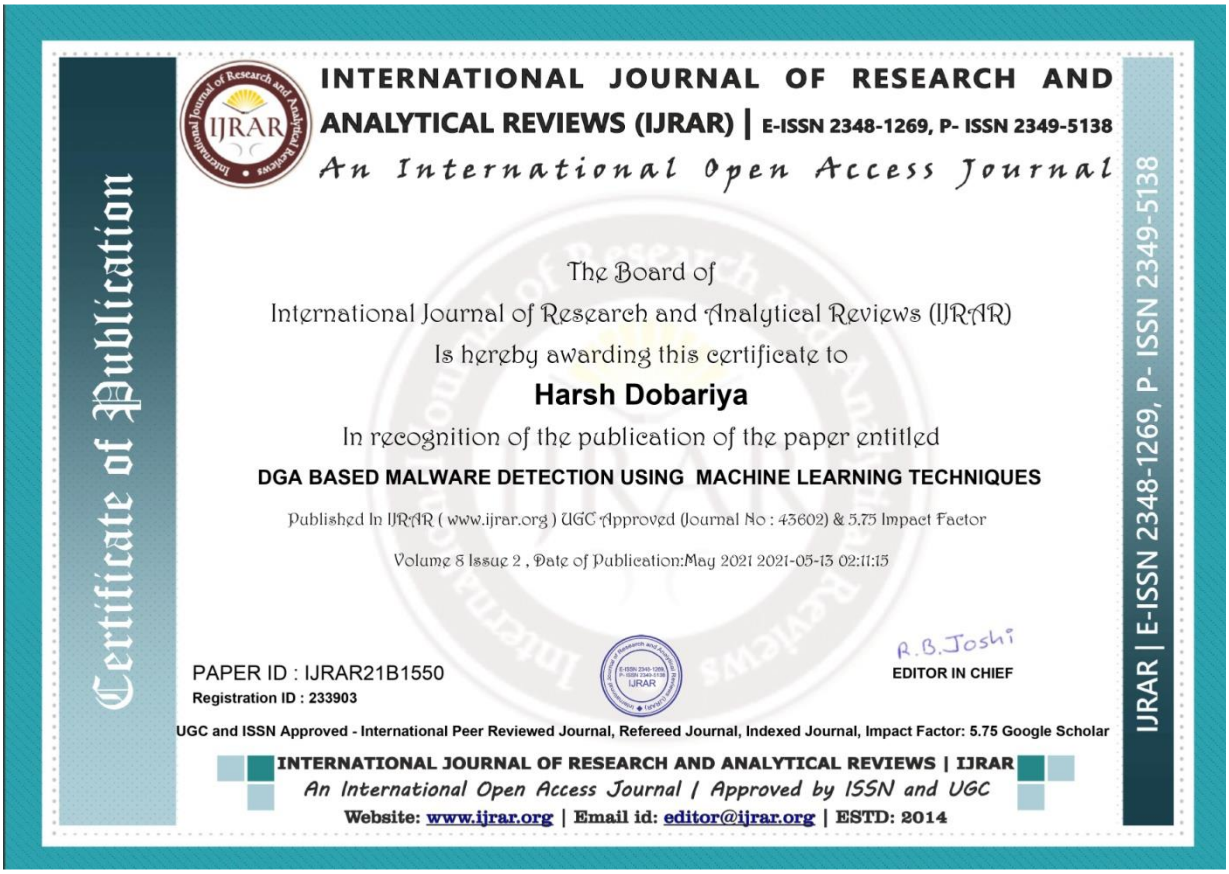
**UGC and ISSN Approved - International Peer Reviewed Journal, Refereed Journal, Indexed Journal, Impact Factor: 5.75 Google Scholar**

**INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR**

*An International Open Access Journal / Approved by ISSN and UGC*

**Website: www.ijrar.org | Email id: editor@ijrar.org | ESTD: 2014**

IJRAR | E-ISSN 2348-1269, P- ISSN 2349-5138



Certificate of Publication

**INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | E-ISSN 2348-1269, P- ISSN 2349-5138**

*An International Open Access Journal*

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

**Harsh Dobariya**

In recognition of the publication of the paper entitled

**DGA BASED MALWARE DETECTION USING MACHINE LEARNING TECHNIQUES**

Published In IJRAR ( www.ijrar.org ) UGC Approved (Journal No : 43602) & 5.75 Impact Factor

Volume 8 Issue 2 , Date of Publication:May 2021 2021-05-13 02:11:15

PAPER ID : IJRAR21B1550

**Registration ID : 233903**

R.B.Joshi

**EDITOR IN CHIEF**

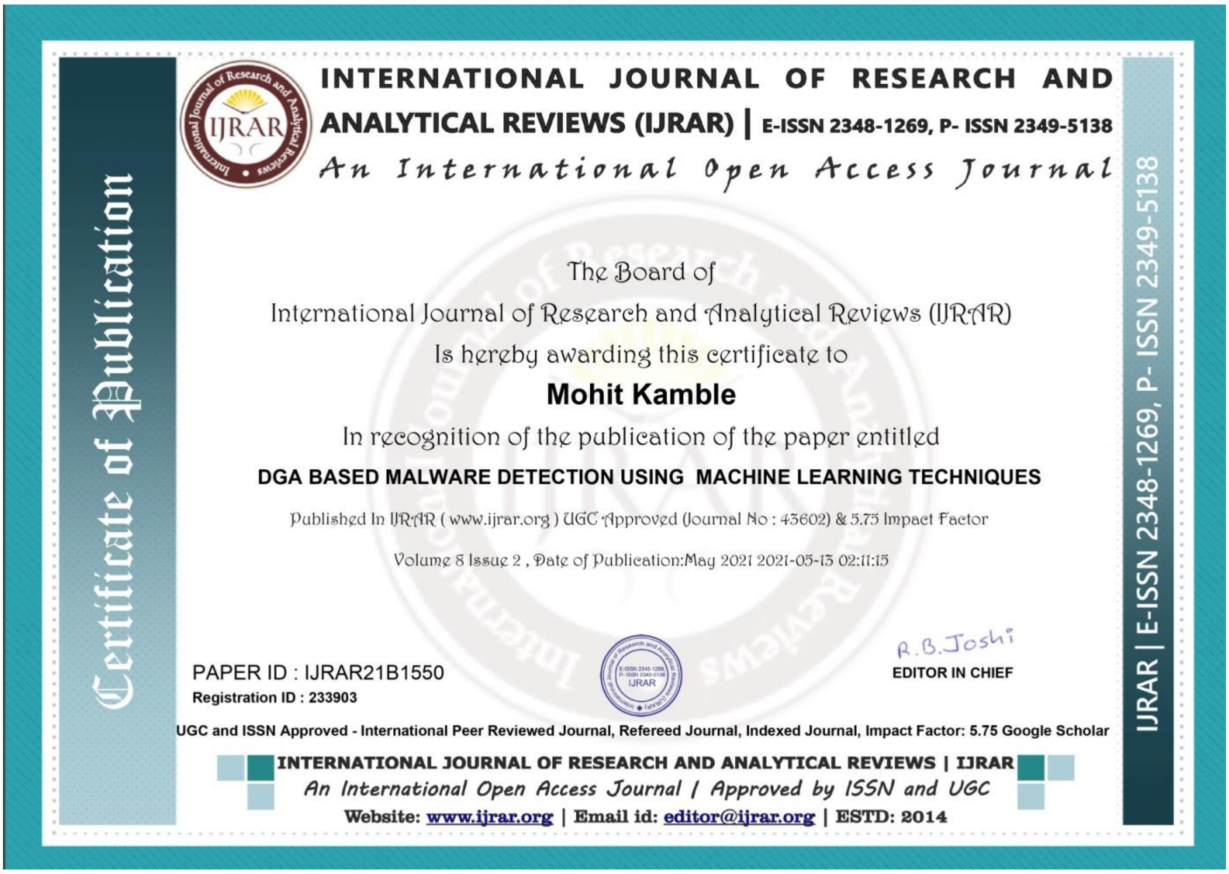
**UGC and ISSN Approved - International Peer Reviewed Journal, Refereed Journal, Indexed Journal, Impact Factor: 5.75 Google Scholar**

**INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR**

*An International Open Access Journal / Approved by ISSN and UGC*

**Website: www.ijrar.org | Email id: editor@ijrar.org | ESTD: 2014**

IJRAR | E-ISSN 2348-1269, P- ISSN 2349-5138



Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | E-ISSN 2348-1269, P- ISSN 2349-5138

An International Open Access Journal

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

**Mohit Kamble**

In recognition of the publication of the paper entitled

**DGA BASED MALWARE DETECTION USING MACHINE LEARNING TECHNIQUES**

Published In IJRAR ( www.ijrar.org ) UGC Approved (Journal No : 43602) & 5.75 Impact Factor

Volume 8 Issue 2 , Date of Publication:May 2021 2021-05-13 02:11:15

PAPER ID : IJRAR21B1550

Registration ID : 233903

R.B.Joshi

EDITOR IN CHIEF

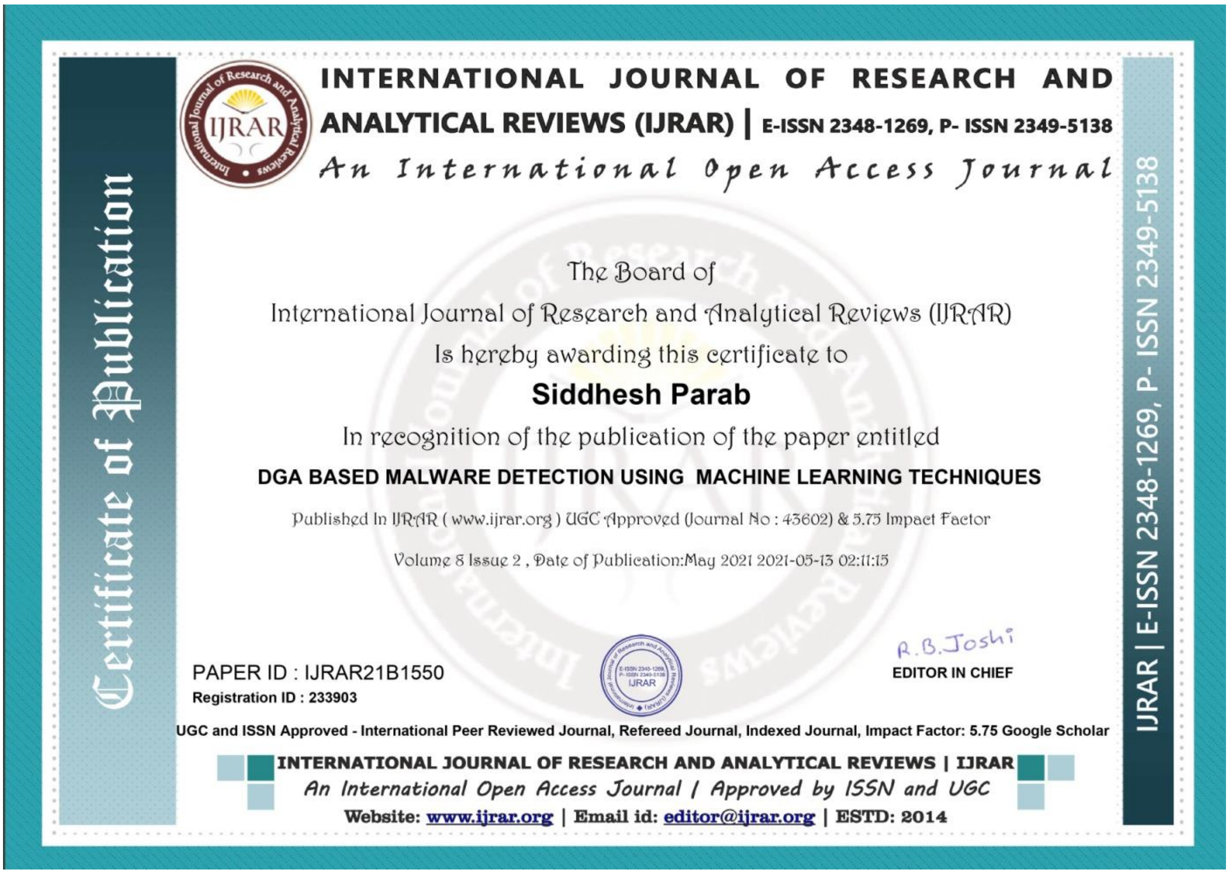
UGC and ISSN Approved - International Peer Reviewed Journal, Refereed Journal, Indexed Journal, Impact Factor: 5.75 Google Scholar

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Open Access Journal / Approved by ISSN and UGC

Website: www.ijrar.org | Email id: editor@ijrar.org | ESTD: 2014

IJRAR | E-ISSN 2348-1269, P- ISSN 2349-5138



Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | E-ISSN 2348-1269, P- ISSN 2349-5138

An International Open Access Journal

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

**Siddhesh Parab**

In recognition of the publication of the paper entitled

**DGA BASED MALWARE DETECTION USING MACHINE LEARNING TECHNIQUES**

Published In IJRAR ( www.ijrar.org ) UGC Approved (Journal No : 43602) & 5.75 Impact Factor

Volume 8 Issue 2 , Date of Publication:May 2021 2021-05-13 02:11:15

PAPER ID : IJRAR21B1550

Registration ID : 233903

R.B.Joshi

EDITOR IN CHIEF

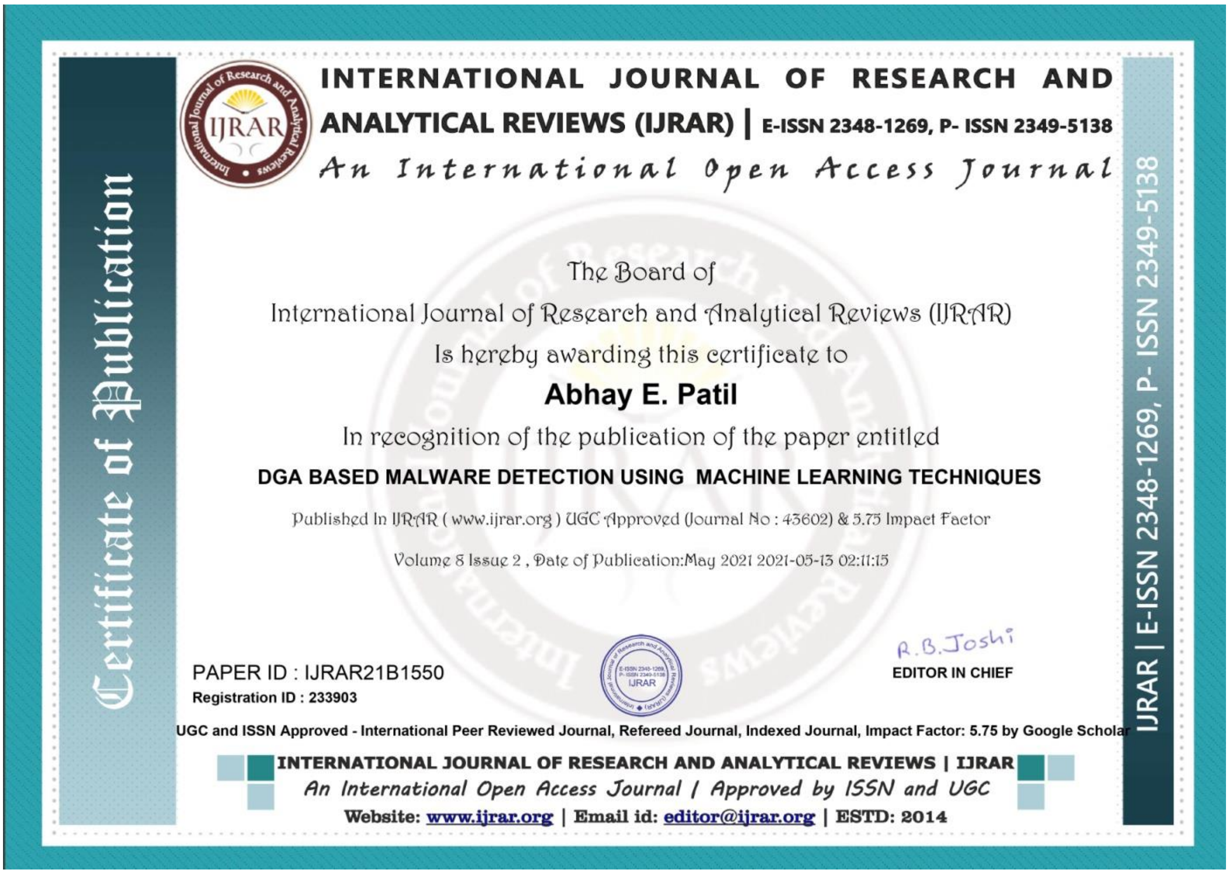
UGC and ISSN Approved - International Peer Reviewed Journal, Refereed Journal, Indexed Journal, Impact Factor: 5.75 Google Scholar

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Open Access Journal / Approved by ISSN and UGC

Website: www.ijrar.org | Email id: editor@ijrar.org | ESTD: 2014

IJRAR | E-ISSN 2348-1269, P- ISSN 2349-5138



Certificate of Publication

**INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | E-ISSN 2348-1269, P- ISSN 2349-5138**

*An International Open Access Journal*

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

**Abhay E. Patil**

In recognition of the publication of the paper entitled

**DGA BASED MALWARE DETECTION USING MACHINE LEARNING TECHNIQUES**

Published In IJRAR ( www.ijrar.org ) UGC Approved (Journal No : 43602) & 5.75 Impact Factor

Volume 8 Issue 2 , Date of Publication:May 2021 2021-05-13 02:11:15

PAPER ID : IJRAR21B1550

**Registration ID : 233903**

E-ISSN 2348-1269 P-ISSN 2349-5138 IJRAR

R.B.Joshi

**EDITOR IN CHIEF**

**UGC and ISSN Approved - International Peer Reviewed Journal, Refereed Journal, Indexed Journal, Impact Factor: 5.75 by Google Scholar**

**INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR**

*An International Open Access Journal / Approved by ISSN and UGC*

**Website: www.ijrar.org | Email id: editor@ijrar.org | ESTD: 2014**

IJRAR | E-ISSN 2348-1269, P- ISSN 2349-5138