# A Malware Detection Method for Health Sensor Data Based on Machine Learning

**Mr. A. Sanjeev Kumar**

[1]*Assistant Professor, Dept. of Master of Computer Applications, Narayana Engineering College, Gudur.*

**S. Sivaprasad**

[2] *PG Scholar, Dept. of Master of Computer Applications, Narayana Engineering College, Gudur.*

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract** - Traditional signature-based malware detection approaches are sensitive to small changes in the malware code. Currently, most malware programs are adapted from existing programs. Hence, they share some common patterns but have different signatures. To health sensor data, it is necessary to identify the malware pattern rather than only detect the small changes. However, to detect these health sensor data in malware programs timely, we propose a fast detection strategy to detect the patterns in the code with machine learning-based approaches. [3]In particular, XG Boost, Light GBM and Random Forests will be exploited in order to analyze the code from health sensor data Terabytes of program with labels, including benign and malware programs, have been collected. The challenges of this task are to select and get the features, modify the three models in order to train and test the dataset, which consists of health sensor data, and evaluate the features and models. When a malware program is detected by one model, its pattern will be broadcast to the other models, which will prevent malware program from intrusion effectively.[1]

***Key Words*::**

- Malware Detection,
- Machine Learning,
- Health Sensor Data.

## 1.INTRODUCTION

With the advent of the Internet of Things Era, all kinds of sensors are applied to collect health sensor data. Inevitably, some malware or malicious codes concealed in health sensor data, which are considered as intrusion in the target host computer, are executed according to the logic prescribed by a hacker. The categories of malicious codes in health sensor data include computer viruses, worms, trojans, botnets, ransomware and so on [1]. Malware attacks can steal core data and sensitive information and damage computer systems and networks. It is one of the greatest threats to today's computer security [2, 3]. The method of performing malware analysis is usually one of two types [4-7]. [4]

studying each component. Binary files can also be disassembled (or redesign) using a disassembler (such as IDA). Machine code can sometimes be interpreted into assembly code, and humans can read and understand assembly code. Malware analysts can understand assembly instructions and get an image of what the program should execute. Some modern malware is created using ambiguous techniques to defeat this type of analysis, such as embedding grammatical code errors. These errors can confuse the disassembler, but they still work in the actual execution[7].

(2) Dynamic analysis is performed by observing how the malware actually behaves when it runs on the host 1 This work was supported by the Qatar National Research Fund

(a member of the Qatar Foundation) under Grant NPRP10-1205-160012.[7]The statements made herein are solely the responsibility of the authors. system. while the broad-spectrum scanning scans the feature code and uses masked bytes to divide the sections that need to be compared and those that do not need to be compared. Furthermore, with the development of malware technology, malware begins to deform in the transmission process in order to avoid being found and killed, and there is a sudden increase in the number of malware variants. The shape of the variants changes a lot so that it is difficult to extract a piece of code as a malware signature[3].

## 2. RELATED WORK

Based on this situation, a natural idea is to apply machine learning-based methods that use existing experience and knowledge to perform static code analysis on unknown binary code and automatically classify malware. According to the guidance, this paper uses the related technologies of machine learning based methods and explores the application of this method in the classification of malware [11-14]. The essence of malware detection is a classification problem, which distinguishes the samples to be detected into malware or legitimate software.[9]Therefore, the host malware detection technology is driven by a machine learning algorithm's core steps, and the main research steps of this paper are as follows: □ Collect sufficient malware code samples and legitimate software samples. □ Perform effective data processing on the sample and extract the features. □ Further select the main features for classification. □ Combine the training using machine learning algorithms and establish a classification model.

□ Detect unknown samples using the trained classification model. The ultimate goal is to find the most effective features and models in this practical task. This chapter introduces the main research questions and basic ideas[10].

In the era of the fourth industrial revolution, there is a growing trend to deploy sensors on industrial equipment, and analyze the industrial equipment's running status according to the sensor data. Thanks to the rapid for industrial control at the edge of the network or at data centers. Due to the considerable development of deep learning in recent years, a common practice of such analysis is to conduct deep learning [2,3,4]. Such methods select a subset of all fetched sensor data stream as the input features, and generate equipment predictions. As a result, the performance of the learning model was seriously impacted by the features selected, thus feature selection plays a critical role for such methods[6].

## 3. EXISTING SYSTEM

Based on this situation, a natural idea is to apply machine learning-based methods that use existing experience and knowledge to perform static code analysis on unknown binary code and automatically classify malware. According to the guidance, this paper uses the related technologies of machine learning based methods and explores the application of this method in the classification of malware[11].

### DISADVANTAGES OF EXISTING SYSTEM:

Must need basic knowledge to perform static code analysis on unknown binary code and automatically

classify malware.

## 4. PROPOSED SYSTEM

In my project , we mainly focus on static code analysis. The early static code analysis methods mainly include feature matching or broad-spectrum signature scanning. Feature matching simply uses feature string matching to complete the detection, while the broad-spectrum scanning scans the feature code and uses masked bytes to divide the sections that need to be compared and those that do not need to be compared. Since both methods need to get malware samples and extract features before they can be detected, the hysteresis problem is serious[12].

Furthermore, with the development of malware technology, malware begins to deform in the transmission process in order to avoid being found and killed, and there is a sudden increase in the number of malware variants. The shape of the variants changes a lot so that it is difficult to extract a piece of code as a malware signature[14].

## ADVANTAGES OF PROPOSED SYSTEM:

simply uses feature string matching to complete the detection, while the broad-spectrum scanning done both comparison and un-comparison

## 5.SYSTEM ARCHITECTURE

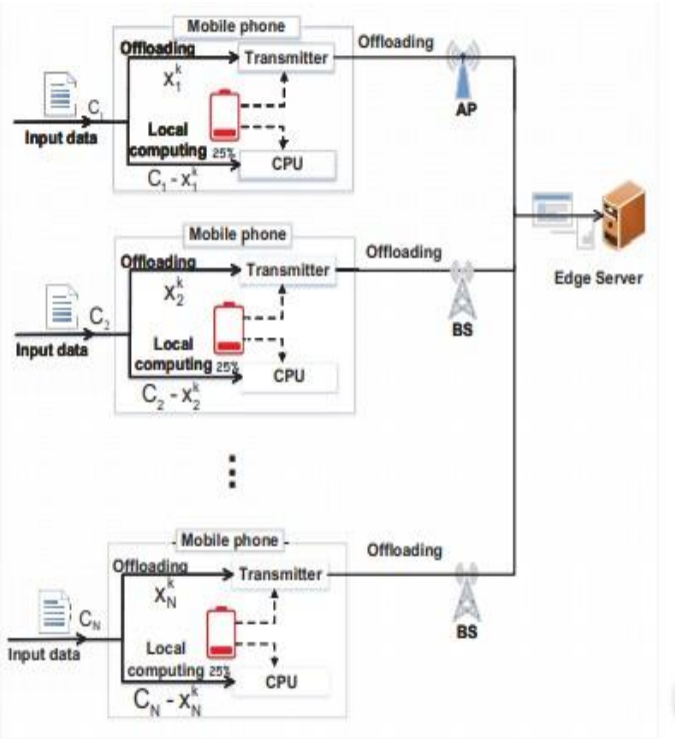


Fig. 1: Multi-user computation offloading with $N$ users in an edge computing environment.

## 6. SYSTEM IMPLEMENTATION

### 6.1 MODULE DESCRIPTION:

**1.Upload dataset**

we get the sensor data from the Kaggle website.

**2.Data Preprocessing** : after getting the data we have to remove the unwanted data from thedataset like null values, unwanted rows, unwanted columns.

**3.Train and Test Model :** after preprocessing the data we have to split the data into two part straining data and testing data.

**4.Run Algorithm :** we have to test machine learning algorithms to find the best algorithm with best accuracy by training the data and testing the data to algorithms.

**5.Accuracy graph :** Finally we plot the graph of all the algorithms with accuracy.

## 7.TESTING

### Testing

Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free. If testing is done successfully it will remove all the errors from the software.

if some words generate wrong trapdoor when there is no back ground knowledge, then the system generates false positive ratio.

### White Box Testing

Testing technique based on knowledge of the internal logic of an application's code and includes tests like coverage of code statements, branches, paths, conditions. It is performed by software developers

### Black Box Testing

A method of software testing that verifies the functionality of an application without having specific knowledge of the application's code/internal structure. Tests are based on requirements and functionality.

### Unit Testing

Software verification and validation method in which a programmer tests if individual unitsof source code are fit for use. It is usually conducted by the development team.

### Integration Testing

The phase in software testing in which individual software modules are combined and tested as a group. It is usually conducted by testing teams.

### Black Box Testing

Blackbox testing is testing the functionality of an application without knowing the details of its implementation including internal program structure, data structures etc. Test cases for black box testing are created based on the requirement specifications. Therefore, it is also called as specification-based testing. Fig.4.1 represents the black box testing:

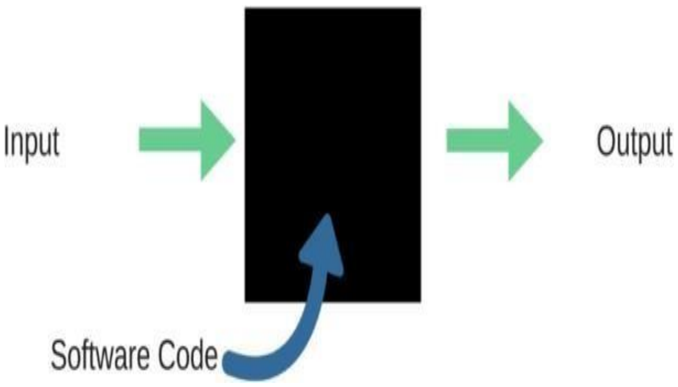


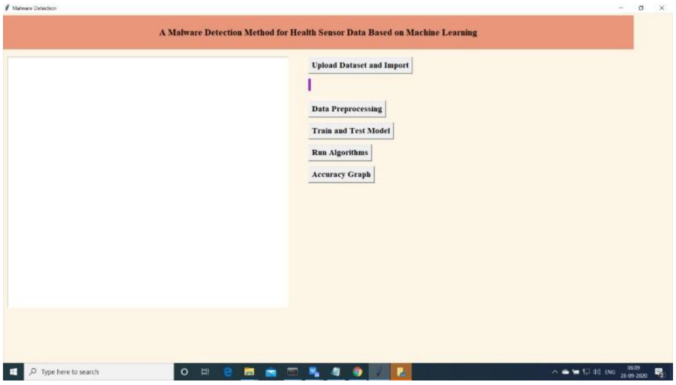
**Fig.:** Black Box Testing

When applied to machine learning models, black box testing would mean testing machine learning models without knowing the internal details such as features of the machine learning model, the algorithm used to create the model etc. The challenge, however, is to verify the test outcome against the expected values that are known beforehand.

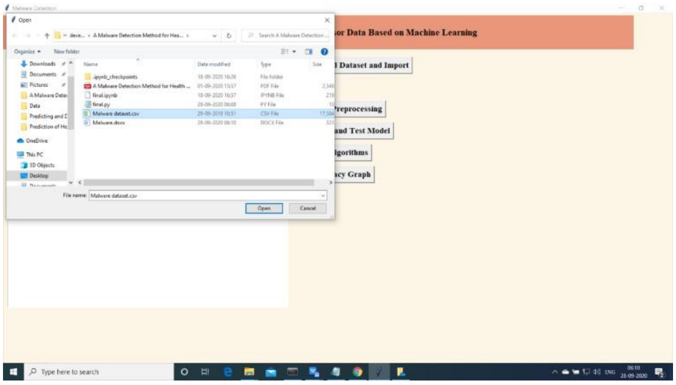
password. Once Login is successful user will do some operations like View Profile, Request Key, View Access Control, View Clinical Reports, and View Patient Details.

## 8. RESULTS AND DISCUSSION

**1.** Now click on "Upload data and import"



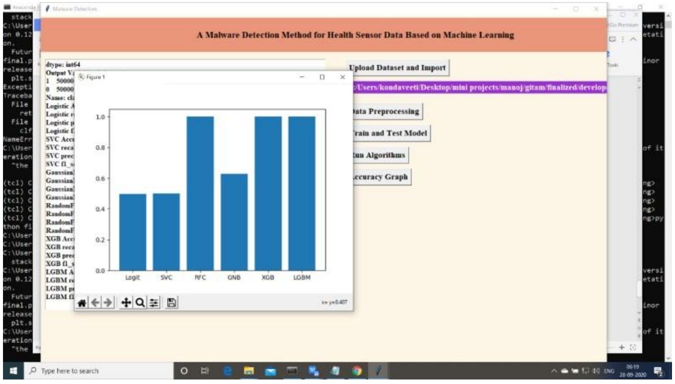
2.Upload the data and read the basic data information will be shown on the screen



| Name | Date modified | Type | Size |
|---|---|---|---|
| .ipynb_checkpoints | 18-09-2020 16:26 | File folder | |
| A Malware Detection Method for Health ... | 01-09-2020 13:57 | PDF File | 2,348 |
| final.ipynb | 18-09-2020 16:37 | IPYNB File | 219 |
| final.py | 28-09-2020 06:08 | PY File | 10 |
| Malware dataset.csv | 29-09-2019 10:31 | CSV File | 17,384 |
| Malware.docx | 28-09-2020 06:10 | DOCX File | 321 |

File name: Malware dataset.csv

3. Accuracy Comparison for all the models


Malware Detection
A Malware Detection Method for Health Sensor Data Based on Machine Learning
dtype: int64
Upload Dataset and Import
:/Users/kondaveeti/Desktop/mini projects/manoj/gitam/finalized/develop
Data Preprocessing
Train and Test Model
Run Algorithms
Accuracy Graph
Figure 1
1.0
0.8
0.6
0.4
0.2
0.0
Logit
SVC
RFC
GNB
XGB
LGBM
Type here to search

## 8. CONCLUSION

With the increasing complexity of malware codes concealed in health sensor data [27-30, 38, 40], the application of machine learning algorithms in the detection of malicious code has been increasingly valued by the academic community and numerous security vendors. Based on the theory of machine learning, this paper combines the advantages of different models [31-33, 36-37] and discusses the static code analysis based on different machine learning algorithms and different code features. This work can provide referential value for the future design and implementation of malware detection technology for machine learning [34]. However, this area still belongs to the developmental stage. There are still many future tasks and challenges and they are summarized below. 1. Lack of valuable data: A machine learning algorithm often requires tens of thousands of data [35] to be trained in order to get an effective model. The acquisition of these basic data often requires manual operations and the speed cannot be guaranteed [36, 37]. [2].

## 9.REFERENCES

[1] L. Wu, X. Du, W. Wang, B. Lin, “An Out-of-band Authentication Scheme for Internet of Things Using Blockchain Technology,” in Proc. of IEEE ICNC 2018, Maui, Hawaii, USA, March 2018.

[2] M. Shen, B. Ma, L. Zhu, R. Mijumbi, X. Du, and J. Hu, “Cloud-Based Approximate Constrained Shortest Distance Queries over Encrypted Graphs with Privacy Protection”, IEEE Transactions on Information Forensics & Security, Volume: 13, Issue: 4, Page(s): 940 – 953, April 2018, DOI: 10.1109/TIFS.2017.2774451.

[3] P. Dong, X. Du, H. Zhang, and T. Xu, “A Detection Method for a Novel DDoS Attack against SDN Controllers by Vast New Low-Traffic Flows,” in Proc. of the IEEE ICC 2016, Kuala Lumpur, Malaysia, 2016.

[4] Z. Tian, Y. Cui, L. An, S. Su, X. Yin, L. Yin and X. Cui. A Real-Time Correlation of Host-Level Events in Cyber Range Service for Smart Campus. IEEE Access. vol. 6, pp. 35355-35364, 2018. DOI: 10.1109/ACCESS.2018.2846590.

[5] Q. Tan, Y. Gao, J. Shi, X. Wang, B. Fang, and Z. Tian. Towards a Comprehensive Insight into the Eclipse Attacks of Tor Hidden Services. IEEE Internet of Things Journal. 2018. DOI: 10.1109/JIOT.2018.2846624.

[6] Z. Wang, C. Liu, J. Qiu, Z. Tian, C., Y. Dong, S. Su Automatically Traceback RDP-based Targeted Ransomware Attacks. Wireless Communications and Mobile Computing. 2018. https://doi.org/10.1155/2018/7943586.

[7] L. Xiao, Y. Li, X. Huang, X. Du, “Cloud-based Malware Detection Game for Mobile Devices with Offloading”, IEEE Transactions on Mobile Computing, Volume: 16, Issue: 10, Pages: 2742 – 2750, Oct. 2017. DOI: 10.1109/TMC.2017.2687918.

[8] https://en.wikipedia.org/wiki/Malware_analysis

[9] Z. Tian, W. Shi, Y. Wang, C. Zhu, X. Du, et al., “Real-Time Lateral Movement Detection Based on Evidence Reasoning Network for Edge Computing Environment”, IEEE Transactions on Industrial Informatics, Volume: 15, Issue: 7, Page(s): 4285 – 4294, March 2019.