iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VI **Month of publication:** June 2022

DOI: https://doi.org/10.22214/ijraset.2022.44668

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

Malware Analysis with Machine Learning: Classifying Malware based on PE Header

Sundeep Varma[1], Jonnadula Narasimharao[2]
[1]Student, Department of Computer Science and Engineering CMR Technical Campus.
[2]Associate Professor, Computer Science And Engineering, CMR Technical Campus.

Abstract: ***If we study the adventures of malware, it has been since the evolution of the computer itself. Since then, they have been a fascinating tool for hackers to exploit computers. Due to the advancement of technology, the malware also has been becoming more complex and hard to detect which in turn creates a cat and mouse chase between security researchers and hackers trying to outsmart each other. While traditional anti-virus softwares are failing to detect these malware variants, in consequence, new technologies have to be adapted to detect them. This paper presents an approach to detect malware using machine learning techniques. This paper explores different malware detection techniques and how ML can be integrated to make it more reliable. ML models are trained with PE features of malware and benign executables. It is observed that the model predicts with 99.4% accuracy. This approach would help researchers to understand and develop more sophisticated antiviruses to detect even complex malware.***

Keywords: Malware, Malware analysis,Machine Learning, PE headers,Random Forest Classifier, Gradient boosting classifier, XGB classifier.

I. INTRODUCTION

Malware, a short name for malicious software is just code like any other program but designed for harmful purposes. As the industries are moving online, the data on cyberspace is rapidly increasing, creating a larger scope for hackers to perform malicious hacks. Though there are many techniques of hacking, using malware and the compromising system seems the most effective. These adventures of malware date back to the early 1970s. The first-ever malware reported was called “Creeper worm”.Creeper was a worm that reproduced itself. It crept through the ARPANET and infected computers, leaving the message "I'm creeper, catch me if you can"[9]. Fred Cohen introduced the word "virus" in the mid-eighties, which he defined as “A program that can infect other programs by modifying them to include a, possibly evolved, version of itself” [9]. Back then viruses spread via floppy disks. As the Internet started, the malware took advantage of this communication medium. With the advancement of technology, viruses have adapted these technologies and have become more and more dangerous.

With a short analysis of statistics on malware attacks, we can estimate how dangerous these tiny chunks of code can be. There are more than 1 billion malware programs out there. Over 560,000 new pieces of malware are detected every day [12], Viruses are mostly spread via .exe files, targeting the windows systems mostly. The trend of malware attacks has been increasing lately, in 2016 it was 580.40 million. In 2017 it reached up to 702.06 million. In 2018 it peaked at 812.67 million [11]. Last year, it was observed that a first ever total of 442,151 unprecedented malware variants have been identified, up 65% year-on-year to an average of 1,211 per day.[10] These numbers show the threat that malware poses. Although the number of malware attacks is decreasing lately , the effectiveness of the malware has been increasing.

To tackle such problems, security researchers have been creating sophisticated anti-virus software. The anti-virus software too began alongside viruses. When the creeper was active, in order to combat it Ray Thomson developed a worm called reaper. Although it wasn't an antivirus, the reaper was the same as creeper. which spread among the devices and deleted the creeper from the infected system[14]. Since then to combat the viruses the anti-viruses systems are being updated. Traditional antivirus works on systems that detect malware based on signatures. Security researchers analyze the malware variants and record signatures that are used as a reference by these antiviruses [3]. If there is any file having the signature same as of the viruses, they classify them as malicious. These systems worked for a while, but attackers choose more complex methods to evade antiviruses such as obfuscating the code or faking the signatures. As a result of the advancement in the evolution of malware, more effective techniques and advanced technologies should be considered and AI seems to be the perfect solution. Being a rapidly advancing technology, it can be implemented to mine features and discover patterns in malware in order to detect new variants.

This paper proposes an approach as mentioned above of integrating machine learning with traditional anti-virus systems. It makes these systems smart and capable of identifying even complex malware. As traditional antiviruses fail.This solution can be implemented with other security measures to prevent malware attacks.

II. LITERATURE REVIEW

In 2019, Daniel Gilbert, CarlesMateu, JordiPlanes presented a paper that explored different approaches to detect malware which inspired my project. In this paper, they explored how machine learning can be applied for malware analysis, history of malware, different types of malware, the structure of a PE file, traditional malware analysis techniques and In-depth analysis of how static features can be implemented using different methods were explained.[3]

Tzu-Yen Wang, Chin-Hsiung Wu, Chu-Cheng Hsieh proposed a system where they explored features to classify malicious executables. Using Dumpbin they created a custom dataset by obtaining PE entries of 1908 benign and 7863 malware files. Priority was given to most occurred PE headers upon calculating the info gain and gain ratio they finalized the optimum set of features which was trained on an SVM model for classification.[2]

Mohamed Belaoued and Smaine Mazouzi in their research paper proposed a system to detect malware in real-time which was based on an examination of the data in the PE-Optional Header fields. To choose features and classify them into subsets, the Chi-square technique and the Phi coefficient were used. These features were used to train several machine learning algorithms. In the proposed system only the technical PE headers were considered as features that might not be enough to generate an efficient model.[4]

Vyas, in 2017, investigated malware detection using PE features and proposed a system. He selected 20 PE features and classified them under 4 categories: metadata, file packing, DLL imports and Function imports. These were trained on 4 different machine learning models SVM, Decision trees, KNN, RF achieved a detection rate of 98%.7% [5]

In June 2019 S. L. Shiva Darshan and C. D. Jaidhar designed a system to detect windows malware based on a hybrid feature set. They integrated both the static features obtained from PE headers, optional headers, file header and dos header with dynamic features like API calls, which are invoked by windows executable during the runtime. LSCV was used for selecting the best features. Their model achieved 99.7% of accuracy This is a great metric but to obtain dynamic features, a program needs to be executed. which leads to more time and computation[6]

Based on a careful investigation of static information in the PE files, J. Bai, J. Wang, and G. Zou proposed a system that uses DLL, API function calls as features, ignoring the DLL and API calls that appeared less than 100 times. Upon calculating the information gain, 30 different DLL and API call features were selected. CfsSubsetEval and WrapperSubsetEval feature selection algorithms were applied to select the most effective features. Several classification techniques were applied to get the stable accuracy of 97.6% [7]

Schultz and Zadok in their paper have proposed a system that uses PE features. These features included the dynamically linked libraries used and DLL function calls made by an executable. They even included the number of different function calls made within each DLL alongside Strings and byte sequences. They employed Ripper and Nb algorithms on the obtained feature set to classify the malware [8]

In 2018 Joshua Saxe, Hillary Sanders published a book called malware data science which explores the integration of malware analysis with machine learning. This book provided the reference for my project. It had all the basic to advance topics explaining PE headers, Anatomy of PE headers, malware analysis, static malware analysis, extracting features, classification algorithms, training and testing with different machine learning algorithms [1]

III. BACKGROUND

A. Malware

Malicious software, in short Malware, is intrusive software that is designed to damage and destroy computers. Mostly used by attackers to damage and dispute the computers[20]. Malware is very much similar to any other normal program but the actions it performs make them dangerous. Malware is a general notation for the domain of harmful software like viruses, worms, Trojan viruses, spyware, adware, and ransomware. This classification is based on the action they perform.

B. Types of Malware

1) *Ransomware* - one of the most famous and effective malware types. It is a type of software that installs itself on a computer, encrypts all the user files and demands a ransom in turn of decrypting the files. [3]
2) *Backdoor* - It's a type of software that installs itself on the computer and allows attackers to access the computer remotely, often allowing the attacker to establish a connection to the victim's machine at any point of time and perform unauthorized actions.
3) *Spyware:* This program gets installed on pc without the knowledge of the user and steals the data. This includes tasks performed, logs, browsing history, app usage etc. Which are monitored remotely. It's often used by government agencies or people in infosec to monitor sensitive environments.

4) *Trojan*: A trojan is software that might seem harmless but upon running or opening it, it downloads other malicious software like a backdoor or spyware and gives access to the system to attackers.
5) *Worm*: Is a type of malware that exploits the vulnerability of the system and can even spread or self replicate onto other machines without any human interaction.
6) *Virus*: A Virus, the most popular amongst other types, is a type of malware that performs malicious action and can propagate to other devices and infect it. It is similar to a worm but viruses spread depending on human activity and can not replicate without them.

C. Malware Analysis

This is the procedure for examining and identifying the intentions and objectives of the malware, extracting the possible amount of information. The data we extract assists us in defining the scope of the Malware's functionality. Often done by security researchers. This includes different procedures like investigating the code and dissecting the way of behavior. Based on the type of analysis it is classified into 2 types static and dynamic. [3][15]

D. Static Malware Analysis

This is the technique of examining malware without actually running it. Is done manually, This analysis is done by analyzing different parameters such as source code, data structures, strings, Pe headers etc. It is straightforward and fast compared to other techniques as it's simple and quick. Below are the techniques used in static analysis [3].

1) *Strings*: This type of analysis is done to find any suspicious strings. Information such as IP addresses,error messages, comments, URLs, attack commands, registry keys etc might be stored in strings and are specially checked in this process to obtain hints regarding the behavior of the program. [3]
2) *Imports*: Contains valuable information as it will help us understand the capabilities and functionalities since it uses a few APIs to import and call the functions. [17]
3) *PE Headers*: Analyzing the PE file headers which contain a lot of information regarding the program such as metadata , checksum, Sections etc which will be a lot of help understanding what the functionality of the code is.
4) *Examining the Source Code*: Disassembling the code and examining, which is just reading and understanding what code does, this takes a lot of time but will help to understand the objective of malware better.

E. Dynamic Malware Analysis

This process involves the actual execution of code and monitoring its behavior in order to understand the functionality and objective of the malware. When static analysis fails or is least effective this process is done. As the actual execution of code is done, a safe and controlled environment should be set up to minimize the risk. A dedicated setup has to be made which might be a complex task.[15]

1) *Function call Analysis*: A program itself cannot contain everything it needs.It has to rely on functions and other libraries to execute some part of code or run a task. Function call analysis analyzes these calls made by malware and determines the behavior of the malware.
2) *Execution control*: In this process the debugging is done, i.e. the CPUs trap flag allows us to generate an interrupt at the required point. This way breakpoints can be set at any opcodes, then both malware and OSes state can be examined to deduce the behavior of the code.
3) *Tracing*: The infected environment is analyzed such as memory and network picking up the traces which provide insight of the behavior of malware.

F. PE File.

Portable executable in-short PE, is a Windows-specific executable file format. Many binary files like object code, DLL, core dumps use this format. It's just a data structure that contains the information about the file itself. It has 2 sections a pe header and section[16][3]

1) *Headers of PE:* This has all the information OS requires to run the executable file. It contains information such as required libraries, import table, code, metadata etc. All these are divided into sections and some of them are mentioned below[16]

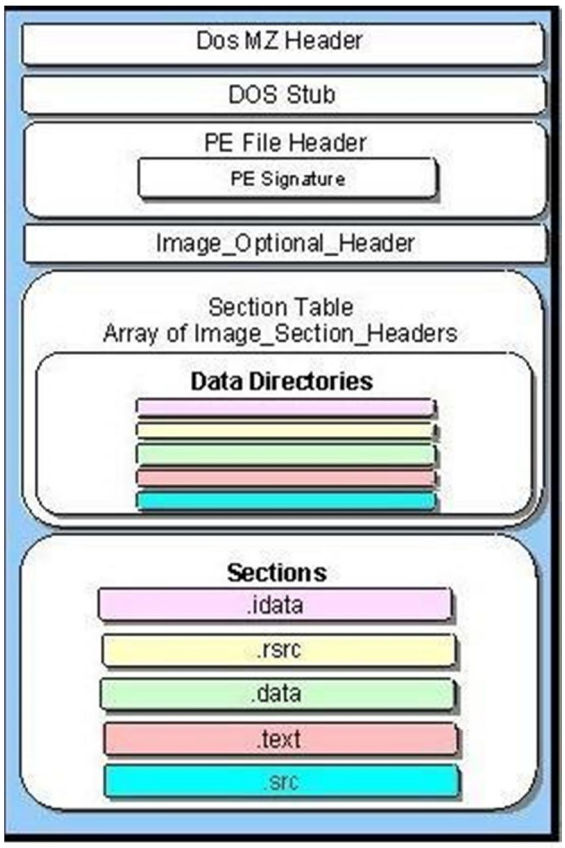


Fig 1: Sections of PE Header

2) *Mz dos* - It defines that the given type is an executable file. It has a magic number. All MS-DOS-compatible executable files have this value set to 0x54AD.
3) *Dos stub*- Does nothing more than printing “program cannot run in dos”, used to check compatibility.
4) *PE file heade*: This contains the signatures of the file like md5 and sha256, Machine information, NumberOfSections, SizeOfOptionalHeader.
5) *Imageheader:* This section stores information about the executable file like subsystems and entry points.
6) *Section Table*: Contains information related to how executable files can be loaded into the memory Containers, sections of virtual size, the size of raw data, a pointer to raw data, characteristics of the section.
7) *data*: This section stores the initialized data such as strings or constants which will be used by the application later in execution.
8) *rdata:* Contains debug directory which stores the size, type and location of different types of debug information related to file.
9) *.idata:* Section contains import table also the data and function which will be imported from DLLs
10) *.edata:* Sections contains export table also data and function which will be exported to DLLS
11) *.rsrc:* Resources such as images and other assets are stored in this section
12) *.bss*: This represents the uninitialized data for the application.

G. Machine Learning

Arthur Samuel is regarded as a forerunner in the field of machine learning. He defined machine learning as "a field of study that gives computers the ability to learn without being explicitly programmed."[18]

Machine learning, in layman's terms, is a field of artificial intelligence that learns from provided data and then uses that knowledge to complete tasks without the need for human interaction. This is of three types: supervised, unsupervised, and reinforcement.

For our experiment, we use support vector machine, logistic regression, random forest, extreme gradient descent, decision trees, and Adaboost. We explore these algorithms below!

1) Linear Regression

Linear regression is a supervised machine learning technique that seeks for the optimal linear relationship between independent and dependent variables. It predicts if the value gives parameters belonging to a specific trained category or not. under sklearn module Linear regression is used to predict the nature of the file.

2) Random Forest

Random forest algorithm is an ensemble classifier Decision trees make up this classifier. This uses the method of bagging. The output is based on the decisions made by these trees at different levels, for our experiment, we take 50 decision trees, This algorithm is under sklearn module

3) Extreme Gradient boosting

In xgboost boosting algorithm, Ensemble trees are made up of decision tree models that are introduced to the ensemble one by one and fitted using the gradient descent optimization algorithm to fix the prediction error of the decision tree models that had previously been built.This type of learning model is called boosting. we use 50 trees for the classification

4) Decision Trees

Decision tree algorithm, as the name suggests, are the tree-like structure of nodes and leaves. TAt the same time that the dataset is broken down into smaller subgroups, an accompanying decision tree is constructed sequentially. While a decision is made at every node which forms the conditions for the given features and classifications of data is made on these rules. Decision trees of depth 10 are generated for our model.

5) AdaBoost

Adaptive boosting, This too is a boosting algorithm. which, as xgboost ensemble trees are constructed with only 1. split is only at a single level and weights are assigned to each instance. It reassigns higher weights to wrongly classified points which are given more importance in the next model. This happens until the error is low, Boosting is used to reduce bias as well as variance. As trees are constructed, decisions will be made and classification will be done. we 100 decision trees for our project

6) Gaussian naïve bias

This algorithm is a variant of naive bias which follows Gaussian distribution. Naive Bayes is a supervised classification technique based on the Bayes theorem.

IV. METHODOLOGY

A. Dataset

Dataset is obtained from Kaggle [19] which was lastly updated 8 months ago. This dataset has 57 columns which have different PE attributes and around 138047 rows.Each row represents an executable file and each row was 57 columns i.e. each executable sample had 57 features of PE headers. This dataset had 41323 benign and 96724 malicious files which are divided into 80:20 ratios for training and testing.

B. Feature Selection

As the curse of dimensionality states, the more number of features the more no of configurations grows exponentially, which in turn increases the area of error probability, slowing down the training time. Our dataset has 54 features of the PE file which might slow down the learning process. An article explores the different approaches to select the important features which might solve the problem of overfitting[21] . This was further explored in another article that has the practically implemented code to select the important features [22]. Different selection methods were implemented but extra-tree-classifiers generated the best features. We base our feature selection on this proposed approach.

As the paper explores the idea of using tree estimators to select the important features.According to the data set, an extra tree classifier is implemented to select the important features which choose 13 Important features out of 54. These features are trained on different machine learning models

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

V. RESULTS

A. Training Time

The below table is a comparison of the time taken by all the implemented machine learning models.

Table 1 : Comparison of training time

Algorithm	Training Time
Decision Trees	0.48s
Random Forest	5.34s
GradientBoosting	8.38s
AdaBoost	10.1s
Gaussian Naive Bayes	0.04s
Linear Regression	0.08s

B. Evaluation Metrics

The effectiveness and performance of the system was estimated using 6 evaluation metrics such as True positive rate, False positive rate, precision, recall. F-measures and accuracy. The calculations are shown below.

TRP=TP/(TP + FN)
FPR=FP/(FP+TN)
Precision= TP/(TP +FP)
Recall= TP/(TP+FN)
f1_score=2*(precision*recall)/(precision + recall)
Accuracy= (TP+TN)/(TP+TN+FP+FN)

Table 2: Comparison of different metrics

Algorithm	Precision	Recall	TPR	FPR	F1-Score
Decision Trees	98.66	98.58	98.6	0.58	98.62
Random Forest	98.9	99.24	99.2	0.43	99.11
GradientBoosting	98.14	98.21	98.1	0.73	98.18
AdaBoost	97.87	97.76	97.7	0.88	97.82
Gaussian Naive Bayes	100	0.02	0.02	0	0.48

C. Accuracy

Most of the employed machine learning algorithms performed well. Decision tree scores with a 98.6 True positive rate and 0.5 False positive rates. Gradient boosting scored with 98.1 true positive rate and 0.73 false-positive rate but random forest topped all the other algorithms with 99.2 True positive rate with 0.43, false positive rate with an accuracy of 99.4% . Below is the comparison of all the accuracies.

Table 3 : Comparison of accuracy

Algorithm	accuracy
Decision Trees	99.18
Random Forest	99.47
GradientBoosting	98.92
AdaBoost	98.7
Gaussian Naive Bayes	70.3
Linear Regression	52.93

VI. CONCLUSION

In this paper, a real-time malware classification approach using pe files is presented. We mined different features based on the dataset of 56 features and used extra-tree-classifier to select 13 most important features which are further trained on 6 different ML algorithms resulting in a 99.4% accuracy rate. In the future other features of static alongside dynamic can be integrated to make the model more effective.

VII. ACKNOWLEDGMENT

We thank CMR Technical Campus for supporting this paper titled "Malware analysis with Machine learning: Classifying malware based on PE Header " ,which provided good facilities and support to accomplish our work. Sincerely thank our Chairman, Director, Deans,Head Of the Department, Department Of Computer Science and Engineering, Guide and Teaching and Non- Teaching faculty members for giving valuable suggestions and guidance in every aspect of our work.

REFERENCES

[1] Joshua Saxe and Hillary Sanders, “ Malware Data Science”,No Starch Press,2018
[2] T. Wang, C. Wu and C. Hsieh, "Detecting Unknown Malicious Executables Using Portable Executable Headers," 2009 Fifth International Joint Conference on INC, IMS and IDC, 2009, pp. 278-284, doi: 10.1109/NCM.2009.385.
[3] Daniel Gibert, Carles Mateu, Jordi Planes, “The rise of machine learning for detection and classification of malware: Research developments, trends and challenges”, Journal of Network and Computer Applications, Volume 153, 2020
[4] M. Belaoued and S. Mazouzi, “A chi-square-based decision for realtime malware detection using PE-file features,” J. Inf. Process. Syst.,
[5] R. Vyas, X. Luo, N. McFarland, and C. Justice, “Investigation of malicious portable executable file detection on the network using supervised learning techniques,” Proc. IM 2017 - 2017 IFIP/IEEE Int. Symp. Integer
[6] S. L. Shiva Darshan and C. D. Jaidhar, “Windows malware detection system based on LSVC recommended hybrid features,” J. Comput. Virol. Hacking Tech., 2018.
[7] J. Bai, J. Wang, and G. Zou, “A malware detection scheme based on mining format information,” Sci. World J., vol. 2014, 2014.
[8] Schultz and Zadok [15] presented some kinds of features such as Dynamic Linked Libraries (DLLs), Application 2009 Fifth International Joint Conference on INC, IMS and IDC 978-0-7695-3769-6/09 $26.00 © 2009 IEEE DOI 10.1109/NCM.2009.385 278
[9] JOHN LOVE, (APR 5, 2018),"A Brief History of Malware — Its Evolution and Impact", accessed 16,March,2022 ,https://www.lastline.com/blog/history-of-malware-its-evolution-and-impact/
[10] sonicwall , “2022 SonicWall Cyber Threat Report” accessed 16,March,2022,https://bit.ly/3IejdHd
[11] purplesec, “2021 Cyber Security Statistics The Ultimate List Of Stats, Data & Trends" accessed accessed 16,March,2022,https://purplesec.us/resources/cyber-security-statistics/
[12] Bojan Jovanovic, DataProt(March 15,2022) "A Not-So-Common Cold: Malware Statistics in 2022 " accessed accessed 16,March,2022, https://dataprot.net/statistics/malware-statistics/
[13] Anton Terekhov, Hotspot Shield, "history-of-the-antivirus", accessed accessed 16,March,2022 , https://www.hotspotshield.com/blog/history-of-the-antivirus/
[14] Pandorafms (Oct 10, 2018) , "History of computer viruses: Creeper and Reaper." accessed 16,March,2022 ,https://pandorafms.com/blog/creeper-and-reaper/

[15] Ori Or-Meir, Nir Nissim, Yuval Elovici, and Lior Rokach. 2019. Dynamic Malware Analysis in the Modern Era—A State of the Art Survey. ACM Comput. Surv. 52, 5, Article 88 (September 2019), 48 pages. https://doi.org/10.1145/3329786
[16] Satyajit Daulaguphu, tech-zealots(May 10, 2018), "A Comprehensive Guide To PE Structure, The Layman's Way" accessed 16,March,2022, https://bit.ly/35Xja5R
[17] Chiheb Chebbi(June 25, 2020), "Malware Analysis - Part 1: Static Analysis", accessed 16,March,2022,https://www.theta432.com/post/malware-analysis-part-1-static-analysis
[18] Mark Esposito,Kariappa Bheemaiah, Terence Tse, (May 3, 2017), "What is machine learning?",accessed 16,March,2022, https://theconversation.com/what-is-machine-learning-76759.
[19] DSC-CLASS (August,2021), "Malware " ,accessed 16,March,2022, https://www.kaggle.com/dscclass/malware
[20] Cisco, "What Is Malware?",accessed 16,March,2022, https://bit.ly/3qdqtgG
[21] Swetha36,(April 23, 2021),"Discovering the shades of Feature Selection Methods, accessed 16,March,2022, https://bit.ly/36vW0mE
[22] Jason Brownlee, April 27, 2021, "How to Develop an Extra Trees Ensemble with Python", accessed 16,March,2022, https://machinelearningmastery.com/extra-trees-ensemble-with-python/

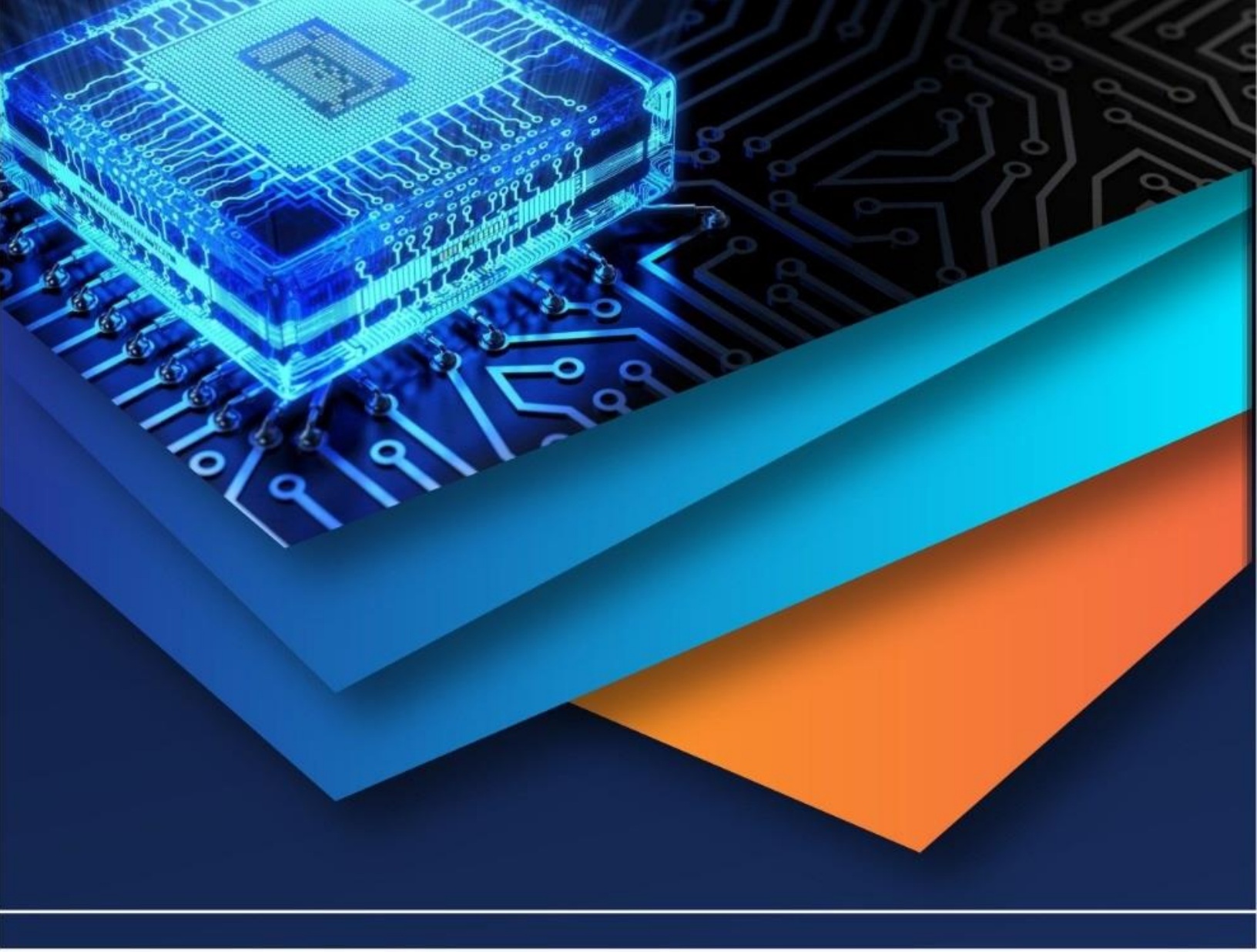
Competing interests

The authors declare that they have no competing interests.

Authors' contributions

Sundeep Varma conducted research on malware,anti virus softwares and machine learning algorithms. He explored the datasets and built the system.

Narshimarao guided the project as an academic project mentor.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY