iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** I **Month of publication:** January 2024

DOI: https://doi.org/10.22214/ijraset.2024.57836

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 12 Issue I Jan 2024- Available at www.ijraset.com

Malware Detection of Portable Executable Using Machine Learning and Neural Networks

Jeet Patil[1], Semil Shah[2], Rohan Patel[3], Atharva Pamale[4], Prof. Rupesh Mishra[5]
Computer Engineering, St. Francis institute of technology Borivali, Mumbai

Abstract: ***Among the more well-known types of automatic detection technology, malware detection includes detection and protection methods against viruses caused by viruses, worms, Trojan horses, spyware, and various types of malicious code. Failure to find a malware program at its inception leaves a space where it will send a significant threat and value to online security not only for individuals, organizations but also the community and the nation. And it seems that antivirus software may fail to detect viruses if it is not updated on a website with an anti-virus engine. The great struggle is to find a new virus because when it encounters new malware behavior, it takes a series of steps based on established law. If the rule of law determines that the new behavior is safe the virus remains undetected.***

So in order to increase the discovery of more malware and precision with greater volume and variety or non- computer-assisted programming, this project uses machine learning techniques and the Neural Network conversion known as deep learning strategies. So in order to find a malware program in usable portable (PE) files we use ML techniques to first determine whether a file is malicious or not and then if it is malicious we put it on our CNN to extract a local feature.

Our binary category shows about 98% accuracy in any given PE file. It uses discarded data in a PE file .Our CNN model shows a 94% accuracy rate in identifying malicious and dangerous codes. It also shows us that CNN is very good at finding source code and binary code, can detect malicious software hidden in the wrong code, and leave malware without a hiding place. This project not only provides a simple network management solution for the detection of malicious software but also provides an easy-to-use GUI so that the average person with little technical knowledge can take security measures in a timely manner.

Keywords: Malware detection, Portable Executables, Machine Learning, Principal component analysis, RandomForest classifier, Convolutional Neural Network (CNN), Deep learning, Binary code analysis, source code analysis, FastApi, User Interface.

I. INTRODUCTION

Attacks on malware and malware have been on the rise with the progress and development of the internet. According to the AV test report which shows that the malware is growing

99.71 million to 1205.25 million in the last 10 years [1]. And the complexity of the malware, its file type, and the structure of the malware have changed dramatically. In the current world there are source code files, binary files, Perl script files, create files, shell texts and read me files in malicious source files as well as a few references and position references as they are in malicious source files. In today's world when criminals or criminals hack into a malware program they use a malicious computer program. There are malicious files included in the wrong files.

So to make the discovery of a malware program effective use of in- depth learning methods to detect malware. In recent years in-depth learning methods such as Convolutional Neural Networks and Recurrent Neural Networks have been used in the fields of information security and provide greater results than traditional methods. In particular the use of CNN has proven to be a very powerful tool. If we can transfer different types of malware to images and make it as input data for CNN to be known. This will help the system administrator to develop an effective way to detect malicious software. We also provide an easy-to-use GUI interface where any person with missing technical knowledge can take timely security measures and make arrangements to track potential internet attacks.

Finally, this project has objectives that need to be achieved:

1) Check whether the file is secure or malicious.
2) If malicious, remove local features from CNN-generated images and separate them into separate non-computer program families.
3) Provide a user-friendly GUI so that anyone can check whether the file is malicious or not and provide security measures and access to your equipment.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue I Jan 2024- Available at www.ijraset.com

II. LITERATURE SURVEY

A. *Malware Detection*

While researching computer hacking detection methods we have come to realize that there are 2 methods which are static analysis and dynamic analysis such as malware functionality [2] .Static analysis has revealed the facts about the system used by the compiler to ensure its accuracy. and effective translation from the source language to the target language [3].Dynamic analysis refers to large techniques that makes deduction based a code by noticing the runtime implementation conduct[4].

B. *Implementation Of Random Forest*

Decision trees are the structures of random forests and clever models. We can think of decision trees as a series of yes / no questions about our data that ultimately lead to the predicted phase (continuous value in the case of retreat). This is a descriptive model because it makes the distinction very similar to ours. What you can do: We run a series of queries from available data until we find a solution (appropriately) [18]. It is important to note that the drug did not make any errors in the training data. We hope this is because we have provided feedback on the tree and there is no limit to the maximum depth (number of levels). The learning model assumes that youhave a good generalization of new data that you have never seen before.

The random forest is a model made of many trees to cut down. Rather than simply predicting the prediction of trees (which we can call “forest”), this model uses two key concepts that give it a random name:

1) Random sampling of training data areas when constructing trees
2) Random subsets of the factors considered when dividing nodes

C. *Convolutional Neural Network Of DeepLearning*

In the early years of artificial intelligence, starting in the 1950s, computer scientists have been working to create computers that can interpret visual input. Yann LeCun, a post-doctoral computer science researcher, initially presented convolutional neural networks, or ConvNets, in the 1980s. Over the past ten years, the Convolutional Neural Network has shown promise in a number of pattern recognition-related fields, including word recognition and image processing [5]. In the last few years, deep learning has shown promise in a wide range of domains, including speech recognition, natural language processing, and visual perception. Convolutional neural networks are the most researched kind of deep neural networks among all of them. A notable enhancement in image processing and a rise in the quantity of annotations.

D. *Malware Detection Of Deep Learning*

Machine learning is used in the selection of pictures, voice and words. Nataraj translated a malicious computer program into a gray image and divided the 9,458 separate malware into the same group [7]. Although Nataraj has isolated a malware program outside of the neural network, the visual approach is attracting the attention of researchers. By using in-depth learning, you can save process time on feature removal and reduce the risk of sandbox analysis.

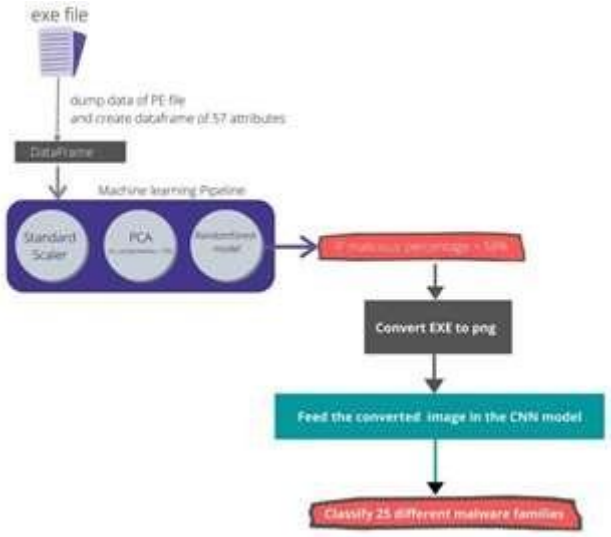
III. OVERVIEW OF APPROCH

A malware program is malicious software that creates a legitimate computer program. It is installed in a variety of ways, but the most common are e-mail spam, fake installer, infected attachments, and criminality identity links. Hackers make a malicious program appear to convince users to install it. Often, users do not know that the program is malware because it looks legitimate. Basically, it is the way malware is installed on a computer. Once installed, a malware program hides in various folders on the computer. If it is an advanced version of a computer that is not compatible with the computer, it can directly access the operating system.

Then start encrypting files and recording personal information. The detection of malware is important for the prevalence of malware online as it serves as an early warning system for malware with malware and cyber attacks. It keeps cybercriminals from infringing on your computer and prevents information from becoming compromised. In fact, there are many file types in the package of malicious files after unzip.

There is a source code file, readme file, dictionary file, script, binary file, usable file and DLL file. And there is a large directory or consecutive directory in the harmful file package. It is not enough to use Nataraj's method alone. As a result, we suggest an improved way to detect malware.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



We take a portable file as an input and read the different parts of the PE file and create a data framework for other processes that include measuring data using a standard scale and using a PCA for reduced size. of data and thus speeding up our algorithm.After the data is pre processedfrom the file we can feed it to the random forest algorithmwhich classifies the fine into benign or malicious.
If a file is found to be malicious we use our CNN model. Our CNN model is trained in 9,639 computer-generated images of 25 different non-computer programming families. We use a kernel size (3,3) that helps to integrate the 8-bit binary image location features. We need an image input for our CNN model, so we convert the PE file into an 8 bit gray scale image.

.

A. Binary Classification of Input File

The input file that we receive from the user is a PE file and the Dataset used to train the class divider has 73 attributes. So By using the PE file python module we get the required attributes according to the database. After this we have previously reviewed the data using standard scalar and PCA from sklearn. After pre- processing we measure pre-processed data in a random forest divider that gives us about 98% accuracy of test data.

Fig.1.1: Pre-Processing Data

pca.explained_variance_ratio_.cumsum (). That will return the vector x so that x [i] returns the increasing variance defined by the initial magnitude of i + 1.

Fig.1.2 Fitting Pre-process Data into Random Forest.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue I Jan 2024- Available at www.ijraset.com

B. File Merging and Labelling

In the CNN model we collected a sample of the data from Kaggle with windows system files and labeled them as malicious and dangerous. Telemetry data containing these properties and mechanical diseases are made by combining heart rate with threat reports compiled by Microsoft's endpoint protection solution. Each line in this database corresponds to a machine, uniquely identified by a machine identifier. Discovery is a basic fact and indicates that a malicious computer program has been detected on a machine.

C. Image Sampling

Here we transfer the file package to the images. Now this class in Microsoft database has been converted to 8 bit gray images.

Fig.3 Image sampling in 8 bit grey scale

D. Handling Imbalance in Dataset

In Fig.4, we can see that the database is not as straightforward as Alaple.A, it has more samples than any other in the database which will give us more negative and false points to eliminate the inequality we used in class_weights. give maximum weight in a small group of people and low weight in the majority section. The sklearn.utilis.class_weights function uses y values to automatically adjust weights as opposed to class frequencies in input data. To use this method, the y_train must not have a single temperature code.

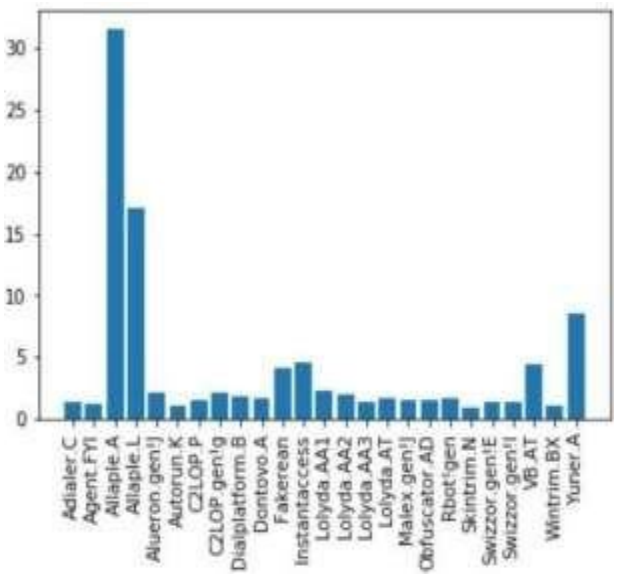


Fig.4 Imbalance dataset

©IJRASET: All Rights are Reserved

E. Dataset Training

With the wrong data and the wrong data, we can train a set of data on CNN. The CNN model learns from Microsoft databases and preliminary processing is done on an average of 70% of trains.

Model: "sequential_3"

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 62, 62, 30)	840
max_pooling2d_4 (MaxPooling2	(None, 31, 31, 30)	0
conv2d_5 (Conv2D)	(None, 29, 29, 15)	4065
max_pooling2d_5 (MaxPooling2	(None, 14, 14, 15)	0
dropout_4 (Dropout)	(None, 14, 14, 15)	0
flatten_2 (Flatten)	(None, 2940)	0
dense_6 (Dense)	(None, 128)	376448
dropout_5 (Dropout)	(None, 128)	0
dense_7 (Dense)	(None, 50)	6450
dense_8 (Dense)	(None, 25)	1275

Total params: 389,078
Trainable params: 389,078
Non-trainable params: 0

Fig.5.1: Train model results

Fig.5.2: Train model results

Fig5.3: Training model results

F. Predict Model Building

After dataset training, a prediction model will be developed to predict other sample**.**

G. Malware Detection

The forecasting model can be used to determine if a new file package is for malicious or malicious money.

So for the first time a PE file is provided as an insert in our binary split model. Now from here we extract the PE file category data and create a dataframe to process and predict whether the file is malicious or not. Here we created a pipeline for measuring data using a standard scale, minimizing data size using system component analysis (PCA) and using a random forest for binary division. Here we come to the realization that the file is malicious or not. So if the file is malicious than we have set a limit of 58% and if true the file will then be sent to our CNN model

In our CNN model the PE file will be converted to a png file (8 bit Gray scale). Now the image is given to the CNN model and we get the output which gives us which family program is not suitable for the computer.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue I Jan 2024- Available at www.ijraset.com

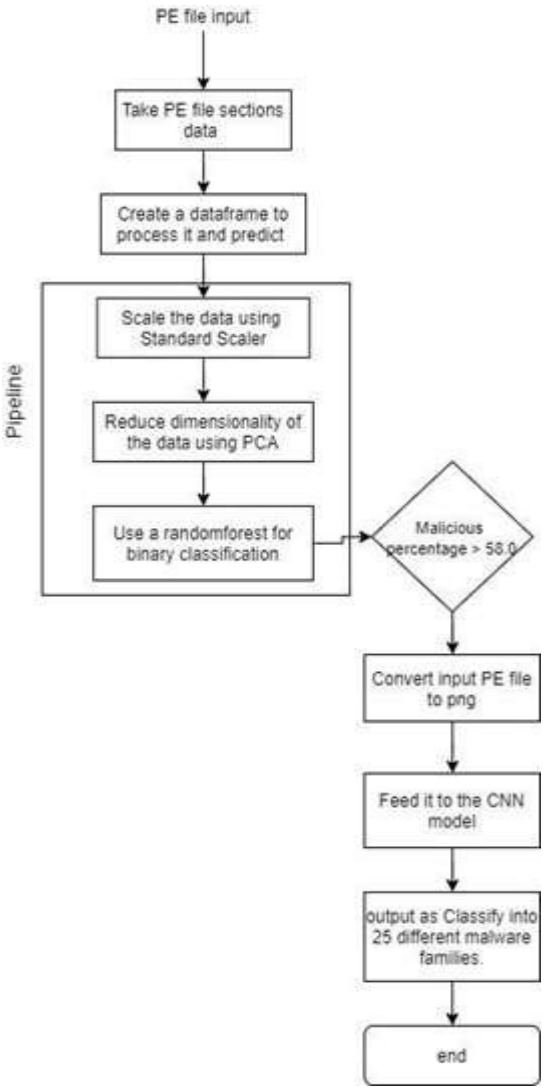


Fig.7.Flowchart of Malware detection.

IV. SOFTWARE AND TECHNOLOGY USED

A. Google Collab

"Memory Failure" - That is a common and terrifying message in Jupyter notebooks when you are trying to use a reading machine or an in-depth reading algorithm on a large data set. Most of us do not have access to unlimited computer power on our machines. It really costs a lot to get a decent GPU from existing cloud providers. Here is the role of Google in participating in our project.
Google Collab is an amazing browser-based online platform that we can use to train our machine models for free. Sounds too good to be true, but thanks to

Dev tools	Python 3.6.3
Deep Learning tools	TensorFlow 1.4.1
Neural Network	ConvolutionalNeural Network

Google, we can now work with large amounts of data,create complex models, and even share our workseamlessly.

©IJRASET: All Rights are Reserved

B. TensorFlow

TensorFlow is a free and open source software for machine learning. It can be used for a wide range of activities but focuses on training and understanding deep neural networks. TensorFlow is a symbolic mathematical library based on data flow and segmentation

C. Keras Library

Keras is an open source software library that provides Python virtual interface for neural networks. Keras acts as a visual connector for the TensorFlow library. Up to version 2.3 Keras supports multiple backends, including TensorFlow, Microsoft Cognitive Toolkit, Theano, and PlaidML.

.

D. FAST API

FastAPI is a modern, fast (highly efficient) web framework for creating APIs using Python 3.6+ based on standard Python type tips. It is a set of rules that developers follow when making their API. One of these rules states that you should be able to get a piece of data (called an app) when you link to a specific URL.

E. Postman

Postman is a collaborative and automated tool for validating APIs for your project. Postman is a Google Chrome application for interacting with HTTP APIs. It provides you with a friendly GUI for creating applications and learning responses. It works in the backend, and ensures that each API works as intended. Here again we do a test of postman automation where 10-12 lines can be punched at the same time to test.

V. RESULT OF IMPLEMENTATION

We design an experiment to evaluate the proposed method. The evaluation environment is described in Table . The TensorFlow deep learning tools was adopted with the ConvolutionalNeural Network .

Table.1 The enviorment of Evaluation

Enviornemnt	Version/Specifications
OS	Windows
CPU	I5 7th gen
Ram	4 gb
Platform	Google Collab

The attributes and source of the collected data were described in Table II. We have collected malware from 9 families in a sample of malicious Microsoft computer software in Kaggle.

Table.2 Malware Families

#	Class	Family
1.	Worm	Allaple.L
2.	Worm	Allaple.A
3.	Worm	Yuner.A
4.	PWS	Lolyda.AA 1
5.	PWS	Lolyda.AA 2
6.	PWS	Lolyda.AA 3
7.	Trojan	C2Lop.P
8.	Trojan	C2Lop.gen!g
9.	Dialer	Instantaccess
10.	TDownloader	Swizzot.gen!I
11.	TDownloader	Swizzor.gen!E
12.	Worm	VB.AT
13.	Rogue	Fakerean
14.	Trojan	Alueron.gen!J
15.	Trojan	Malex.gen!J
16.	PWS	Lolyda.AT
17.	Dialer	Adialer.C
18.	TDownloader	Wintrim.BX
19.	Dialer	Dialplatform.B
20.	TDownloader	Dontovo.A
21.	TDownloader	Obfuscator.AD
22.	Backdoor	Agent.FYI
23.	Worm:AutoIT	Autorun.K
24.	Backdoor	Rbot!gen
25.	Trojan	Skintrim.N

VI. CONCLUSION

This project proposes a feasible solution for network administrators to efficiently identify malware at the very inception in the severe network environment nowadays. It also provides a easy user interface so that common public can scan their files and install a solution to stop the spread of malware on their systems.

This study achieves the following objectives:

1) Checking whether the file is secure or malicious.
2) Find a malicious computer program hidden behindbenign files.
3) If malicious then extract local features from imagegenerated using CNN and classify into different malware families.
4) Provide a User-friendly GUI so that anyone can check whether file is malicious or not and provide steps to secure and recover your machines.

Besides, in session result and evaluation, we can realize that the proposed method has high true positive rate and low false positive rate and all the accuracy are higher than 95% in whole project.

If we can collect malicious APP and malware in the region of Internet of Things（IoT）continuously. We will have a much powerful predict model. With the predict model, system administrator can exam the unknown sample quickly and efficiently in the future.

VII. ACKNOWLEDGMENT

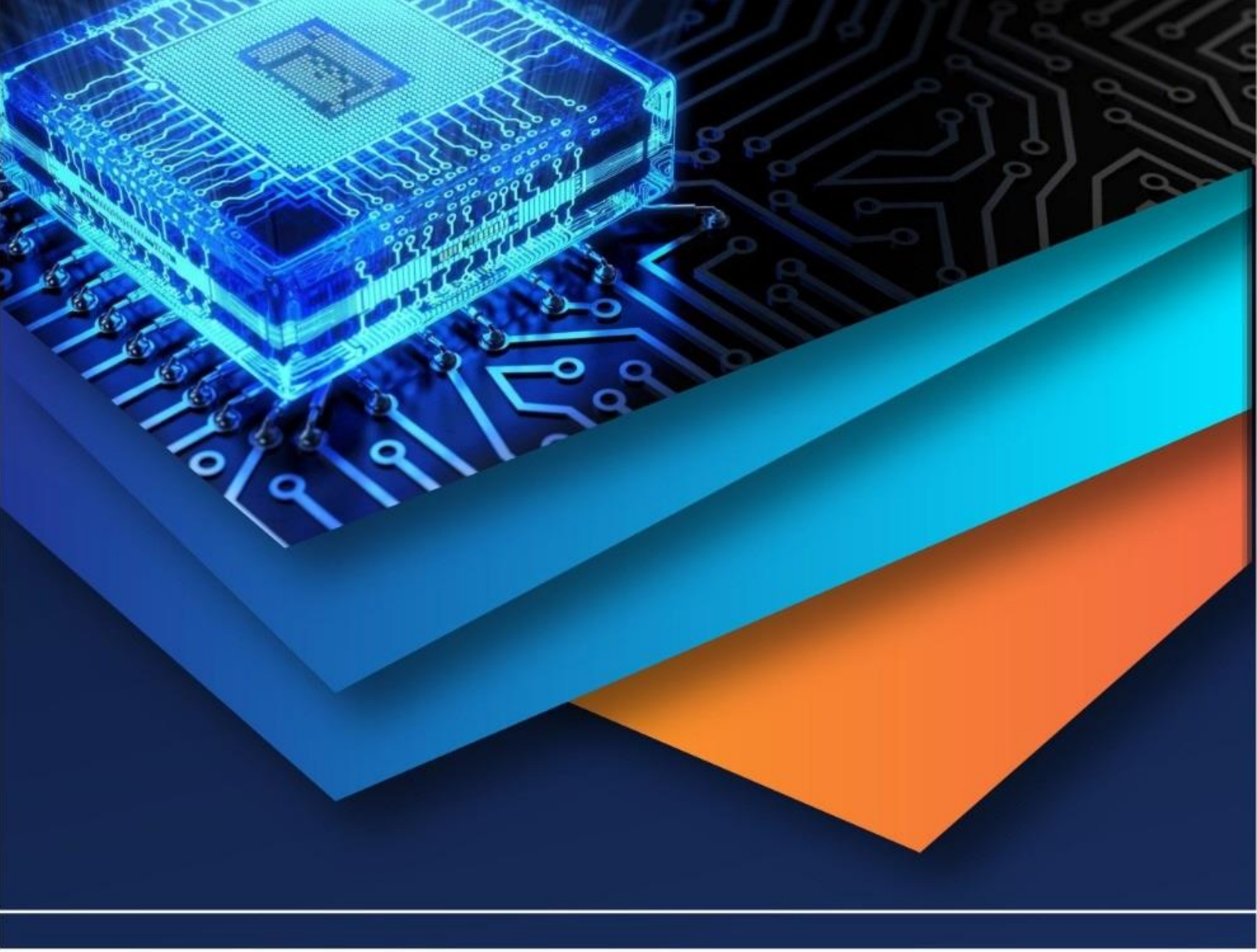
The authors would like to thank Prof. Rupesh Mishra for providing support by making available all the equipment and a suitable workplace to discuss and implement the idea. Also, the authors thank him for their guidance on the chosen topic.

REFERENCES

[1] https://www.avtest.org/en/statistics/malware/
[2] Stefan Katzenbeisser,Johannes Kinder and Helmut Veith"Malware Detection"Springer, Boston, MA. Encyclopedia of Cryptography and Security 2011 Edition pp.978-1-4419-5905-8
[3] David Brumley"Static Analysis"Springer, Boston, MA. Encyclopedia of Cryptography and Security 2011 Edition pp.978-1-4419-5905-8

[4] Mihai Christodorescu, Vinod Ganapathy"Static Analysis"Springer, Boston, MA. Encyclopedia of Cryptography and Security 2011 Edition pp.978-1- 4419-5905-8
[5] Saad ALBAWI , Tareq Abed MOHAMMED, Saad AL- ZAWI"Understanding of a Convolutional Neural Network"2017 International Conference on Engineering and Technology (ICET), 2017 pp.978-1-5386-1949-0
[6] https://towardsdatascience.com/malware-classification- using-convolutional-neural-networks-step-by-step- tutorial-a3e8d97122f
[7] Chia-Mei Chen, Shi-Hao Wang, Dan-Wei Wen
[8] "Applying Convolutional Neural Network for Malware Detection" in 2019 IEEE 8th Joint International Information Technology and Deep Learning Conference (ITADL 2019)
[9] Malware Detection on Byte Streams of PDF Files Using Convolutional Neural Networks Young-Seob
[10] Jeong , Jiyoung Woo , and Ah Reum Kang SCH Media Labs,Soonchunhyang University, Asan 31538, Republic of Korea (3rd April 2019)
[11] Detecting Malware with an Ensemble Method Based on Deep Neural Network Jinpei Yan,1 Yong Qi,1 and Qifan Rao1, Department of Computer Science andTechnology, Xi'an Jiaotong University, Xi'an, Shaanxi, China(12 Mar 2018)
[12] https://www.kaggle.com/c/malware-classification
[13] https://www.hindawi.com/journals/scn/2018/7247095/h ttps://towardsdatascience.com/autodeploy-fastapi-app- to-heroku-via-git-in-these-5-easy-steps-8c7958ef5d41
[14] https://tensorflow-object-detection-api tutorial.readthedocs.io/en/latest/
[15] https://fastapi.tiangolo.com/advanced/custom-response/
[16] https://fastapi.tiangolo.com/async/
[17] https://fastapi.tiangolo.com/
[18] https://www.smashingmagazine.com/2018/01/understan ding-using-rest-api/
[19] https://towardsdatascience.com/an-implementation-and- explanation-of-the-random-forest-in-python- 77bf308a9b76
[20] https://github.com/tiangolo/fastapi/issues/426
[21] https://ianrufus.com/blog/2020/12/fastapi-file-upload/
[22] https://stackoverflow.com/questions/63580229/how-to-save-uploadfile-in-fastapi

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)