A STUDY ON DETECTION OF MALWARE USING MACHINE LEARNING ALGORITHMS

[1.]D. Swetha
Research Scholar, Bhartiya Science & Technology Innovation University
Asst.Professor, St.Francis College For Women, Hyderabad.
dswetha.6262@sfc.ac.in

[2.]Dr.V.Goutham
Professor,St.Mary's Engineering College, Hyderabad
Department of Computer Science
v.goutham@gmail.com

ABSTRACT:

One of the most significant issues facing internet druggies currently is malware. Polymorphic malware is a new type of vicious software that's further adaptable than former generations of contagions. Polymorphic malware constantly modifies its hand traits to avoid being linked by traditional hand- grounded malware discovery models. Counter-attacking measures have been more effective, with antivirus companies expanding their signature database, which is routinely updated, although they are inefficient and ineffective in the case of polymorphic malware. To identify vicious pitfalls or malware, we used a number of machine literacy ways. A high discovery rate indicated that the algorithm with the stylish delicacy was named for operation in the system. As an advantage, the confusion matrix measured the number of false cons and false negatives, which handed fresh information regarding how well the system worked. In particular, it was demonstrated that detecting dangerous business on computer systems, and thereby perfecting the security of computer networks, was possible using the findings of malware analysis and discovery with machine literacy algorithms (Naive Byes, SVM,RF, and with the proposed approach) integrals. The results showed that when compared with other classifiers, DT(99), CNN(97.76), and SVM(94.41) performed well in terms of discovery delicacy. These results are significant, as vicious software is getting decreasingly common and complex.

Keywords:

CNN, SVM, DT, cybersecurity, cyberattack, cyber warfare, cyber threats, suspicious activity.

1. Introduction:

Cyberattacks are presently the most burning concern in the realm of ultramodern technology. The word implies exploiting a system's excrescencies for vicious purposes, similar as stealing from it, changing it, or destroying it. Malware is an illustration of a cyberattack. Malware is any program or set of instructions that's designed to harm a computer, stoner, business, or computer system [1]. The term "malware" encompasses a wide range of pitfalls, including contagions, Trojan nags, ransomware, spyware, adware, mischief software,

wipers, scareware, and so on. vicious software, by description, is any piece of law that's run without the stoner's knowledge or concurrence [2]. In particular, this study demonstrated that detecting dangerous business on computer systems, and thereby perfecting the security of computer networks, was possible employing the findings of malware analysis and discovery with machine literacy algorithms
(Naive Byes, SVM, RF, and with the proposed approach) integrals. Malware discovery modules are responsible for analysing data they've collected and been trained with to determine whether or not a specific piece of software or network connection constitutes a security concern [3,4]. As an illustration, consider a machine literacy system that can explicitly express the principles that uphold the patterns it has observed
[5]. Algorithms that have been trained by machine literacy systems can ameliorate their capability to prognosticate using feedback regarding how well they performed on former tasks and using that information to make changes [6].

Worldwide, cybercriminals pose a serious trouble to businesses, universities, governments, and individualities through the use of vicious software and the theft of nonpublic data[7]. Every day, thousands of fraudsters employ dangerous software in an attempt to gain access to networks, steal data, or transfer plutocrat. As a result, keeping sensitive information safe has come an critical concern in the scientific world. This study aimed to give a comprehensive frame for discovering vicious programs and guarding private information from hackers by employing data mining and machine literacy bracket approaches.
Malware is software created with the express purpose of causing detriment to a computer or network, for illustration, by covering its druggies or stealing their plutocrat. Malware attacks are getting decreasingly common and now indeed affect IoT bias, medical gear, and environmental and artificial control systems. ultramodern spyware is notoriously hard to descry, as it constantly updates its law . The proliferation of malware has rendered traditional hand- grounded defenses ineffective. rather, it's necessary to take a broader range of protective conduct.

Both static and dynamic knowledge styles may be used to identify behavioural parallels between members of the same family of malware. Unlike static analysis, which examines dangerous lines contents without actually running them, dynamic analysis takes their behaviour into account by tracking data flows, recording function calls, and adding monitoring law to dynamic binaries. Machine knowledge algorithms may work analogous static and behavioural artefacts to describe the ever- evolving structure of contemporary malware, allowing them to identify increasingly complex malware assaults that could differently avoid discovery using hand- predicated ways. As machine knowledge- predicated results do not calculate on signatures, they are more successful against lately released malware. Deep knowledge algorithms that can perform point engineering on their own can be used to gain and represent features more directly. Kill Chain used for cyberattack protection and as for security measure to cover networks.

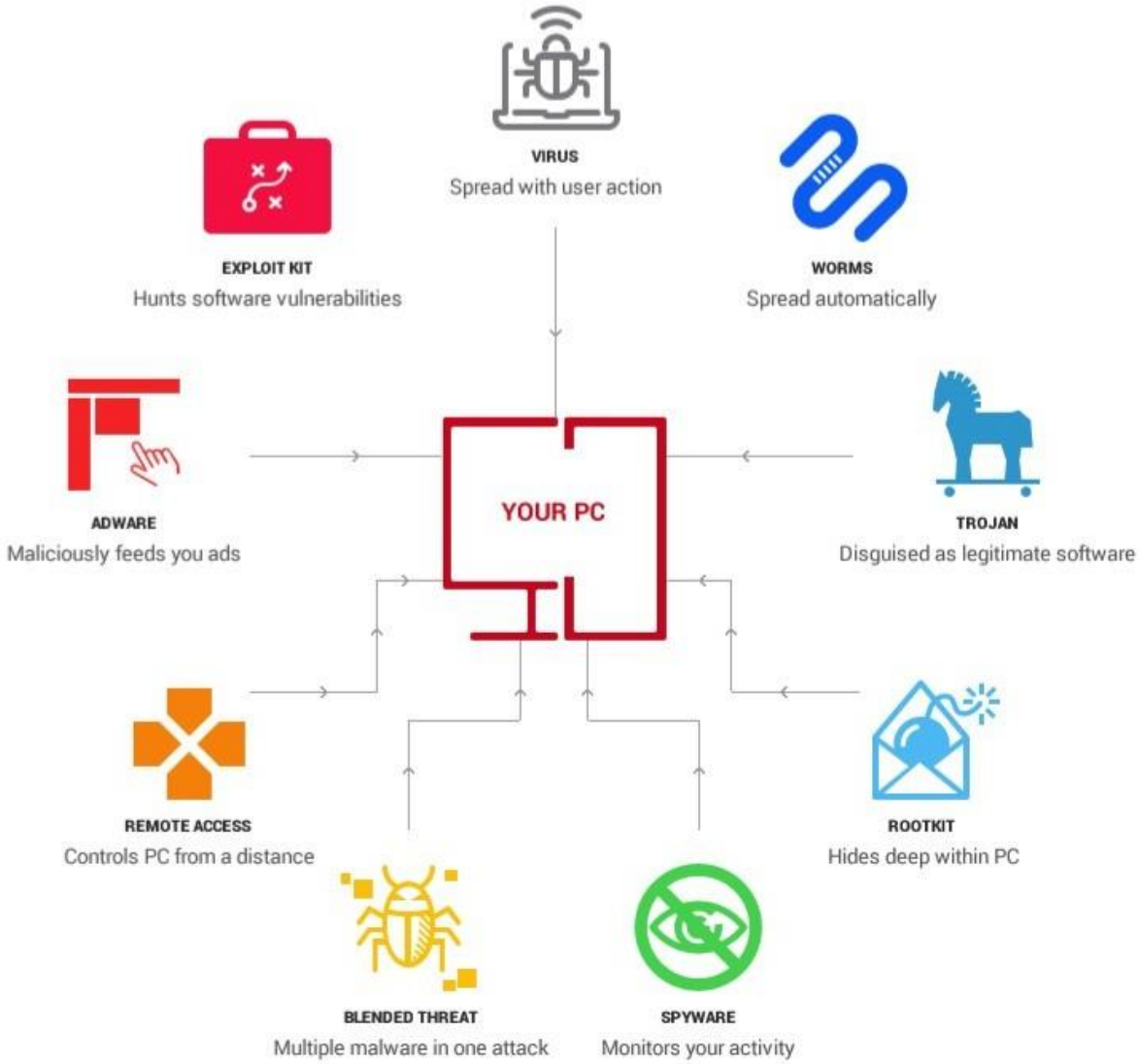


FIGURE 1

2. Literature Review

The proliferation of computers, smartphones, and other Internet- enabled widgets leaves the world vulnerable to cyber assaults. A plethora of malware discovery styles have arisen in response to the explosion in malware exertion. When trying to identify vicious law, experimenters use a variety of big data tools and machine literacy ways. Traditional machine literacy- grounded malware discovery approaches have a considerable processing time, but may effectively identify recently arising malware. Point engineering may come obsolete due to the frequence of ultramodern machine learning algorithms, similar as deep literacy. In this study, we examined a variety of malware discovery .Experimenters have created ways to use machine literacy and deep literacy to check samples for vicious intent.Without data, no operation erected for a digital platform can perform its function[8]. There are several cyber pitfalls, so it's essential that preventives be taken to guard data. Although point selection is delicate when developing a model of any kind, machine literacy is a slice-edge approach that paves the way for precise vaticination. The approach needs a workaround that's adaptable enough to handlenon-standard data. To effectively manage and help unborn assaults, we must assay malware and produce new rules and patterns in the form of creation of malware type. To find patterns, IT security professionals may use malware analysis tools. The vacuity of technologies that assay malware samples and determine their position of malice significantly profit the cybersecurity sector. These tools help cover security cautions and help malwareattacks.However, we must exclude it before it transmits its infection any farther, If malware is dangerous. Malware analysis is getting decreasingly popular as it helps businesses lessen the goods of the growing number of malware pitfalls and the adding complexity of the ways malware can be used to attack[10].

Malicious programs and their threats, also known as "malware," have become more common and sophisticated as the Internet has evolved. Its rapid spread on the Internet has given malware authors access to a wide variety of malware generation tools [11]. The scope and complexity of malware grows daily. This research focused on analysing and measuring the performance of classifiers to gain a deeper understanding of how machine learning works. Latent analysis extracted features from recovered PE files and library information. six classifiers based on ML techniques were evaluated. It was recommended to train and test an ML system to determine if a file is malicious. Experimental results confirm that the random forest method is suitable for classifying data with 99.4% accuracy. These results showed that the PE library is compatible with static analysis and that malware detection and characterization can be improved simply by focusing on a few properties. The main advantage is that users can check files for validity before opening them, thus reducing the chances of accidental installation of malicious software [12].

3. Performance Measures

There are various methods to evaluate the performance of the algorithms. One of these methods is to determine the area under the curve or the ROC curve and other parameters which are also known as Confusion Metrics. To evaluate the performance measure of the classification model for a dataset that gives the true values are known, the confusion matrix table is used.

	Predicted Class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Table1

The table shown above is known as the confusion matrix and has four sections. The two sections in the green are the True Positive and True Negative and these are the observations which are correctly predicted. The other two sections are in red because these values are wrongly predicted and thus need to be minimized. These sections are false negative and False Positive respectively and occur when there is a contradiction between actual class and the predicted class.

True Positives (TP)

These are the values which are correctly predicted and are positive values which can be described as the positive value of actual class and positive value of predicted class. It is denoted by TP.

True Negatives (TN)

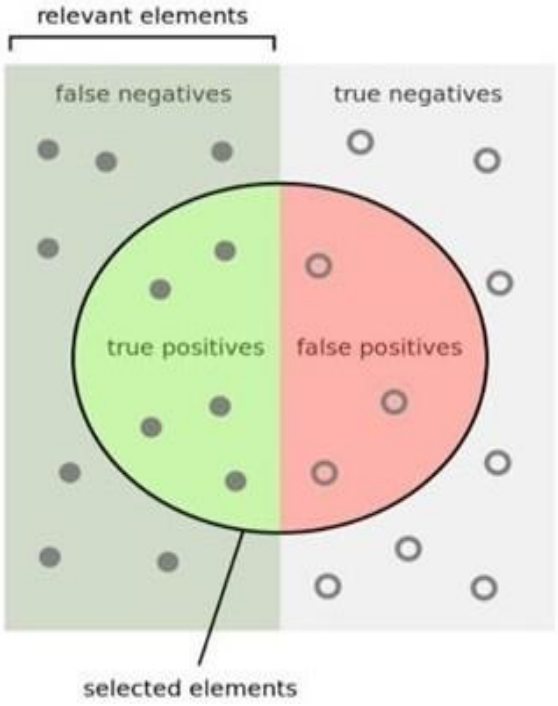
These are the values which are correctly predicted but negative values which refers to the negation of actual class and negation of predicted class.It is denoted by TN.

- **False Positives (FP)**

These are the values which are wrongly predicted but is true in reality i.e. -when we have positive values of actual class but negation in predicted class.

- **False Negative (FN)**

These are the values which are wrongly predicted and negative in actual class.



4. Research Problem:

Potentially harmful components of malware can be detected using static or dynamic analysis. Static analysis, such as the reverse engineering technique used to disassemble viruses, focuses on analysing malware binaries to discover malicious strings. However, dynamic analysis is intended to monitor dangerous software even when running in a controlled environment such as a virtual machine. Both methods have advantages and disadvantages. However, it is best to use both when analysing malware . Reducing the number of dangerous functions can improve malware detection accuracy. That way, researchers have more time to analyse the collected data. While a large number of traits are used to detect malware, we are concerned that fewer, more robust traits may work as well. The process of choosing which malicious functions to implement begins with discovering possible methods and algorithms. We find unprecedented malware and need a solution that can significantly reduce the

number of features currently required. We evaluate the accuracy among three ML techniques (DT, CNN, and SVM) for malware detection.

5 . Methodology

This research paper presents the various steps and components of a typical machine learning workflow for malware detection and classification, examines the challenges and limitations of such a workflow, and focuses on deep learning techniques to develop this Evaluate the latest innovations and trends in the field. The research methodology proposed in this research study is shown below .
To give a more comprehensive understanding of the proposed machine learning method for malware detection, Figures 2 and 3 show the workflow process from start to finish.

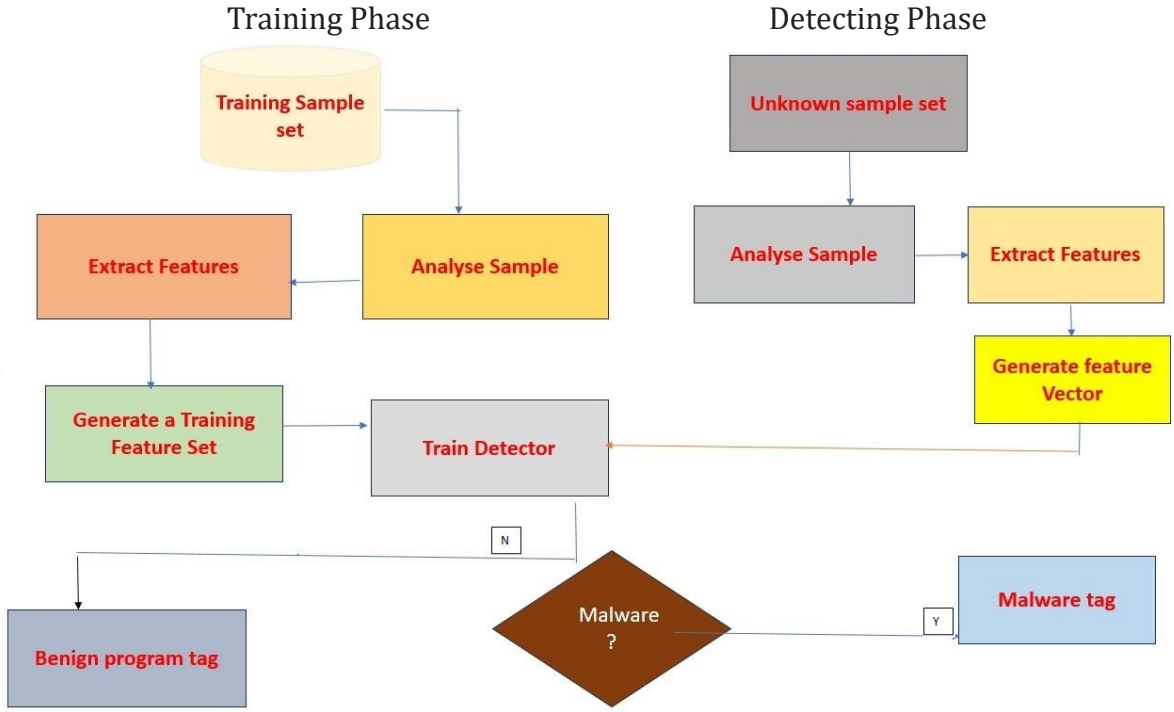


FIGURE 2

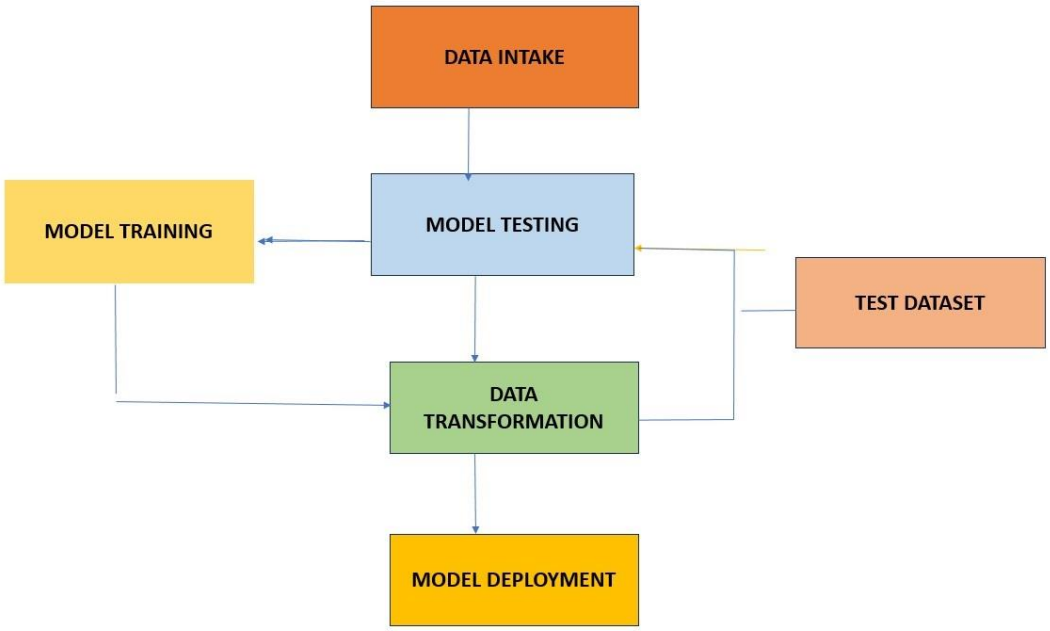


FIGURE 3

5.1. Dataset

This collection contains many data files containing log data for various types of malwares. These recovered log functions can be used to train various models. We found about 51 different malware families in the sample. It contained over 16000 data points from various locations. The dataset had 279 columns and 16000 rows.

5.2. Preprocessing

The data was stored in the file system as binary code, and the files themselves were raw executable files. A protected environment or virtual machine (VM) was required to unzip the executable. The PEiD software automated decompression of compressed executables.

5.3. Feature Extraction

Twentieth century records often contain tens of thousands of features. In recent years, it has become apparent that the resulting models are overly suitable for machine learning as the number of features increases. To address this issue, I created a smaller set of functions from a larger set of functions. This technique is commonly used to maintain the same level of accuracy while using fewer features. The aim of this study was to refine existing datasets of dynamic and static features by retaining the most useful ones and removing those of no value for data analysis

5.4. Feature Selection

Upon completion of feature extraction, including finding additional features, feature selection was performed. Feature selection, which involves selecting features from a pool of newly-recognized qualities, was an important process for improving accuracy, simplifying models, and reducing overfitting. Researchers have used many functional classification strategies to identify malicious code in software. The feature rank method was extensively used in this study because it is highly effective in selecting suitable features for building malware detection models.

Algorithm for Decision Tree

Algorithm 1: Decision Tree Pseudocode

Input: I, where "I" is a set of classified instances.

Output: Decision Tree

Require: I is not empty, no_of_attributes is greater than 0.

1: **Procedure** Build the Tree
2: **Repeat**
3: maximum_gain = 0
4: split_A = null
5: e = Entropy (Attributes)
6: **for all** Attributes a in S **do**
7: gain = Information_Gain(a, e)
8: **if** gain **is greater than** maximum_gain
9: **then**
10: maximum_gain = gain
11: split_A = a
12: **end if**
13: **end for**
14: Partition(I, split_A)
15: **until** All partitions processed
16: End Procedure

6. Results and Discussion:

The two main phases of the classification process are training and testing. Malicious files were sent to the system to train it . An automatic classifier was trained using a learning algorithm. Each classifier

```
result1 = knn.predict(legit_test)
conf_mat1 = confusion_matrix(mal_test,result1)

print(conf_mat1)
```

```
[[19223   166]
 [   95  8126]]
```

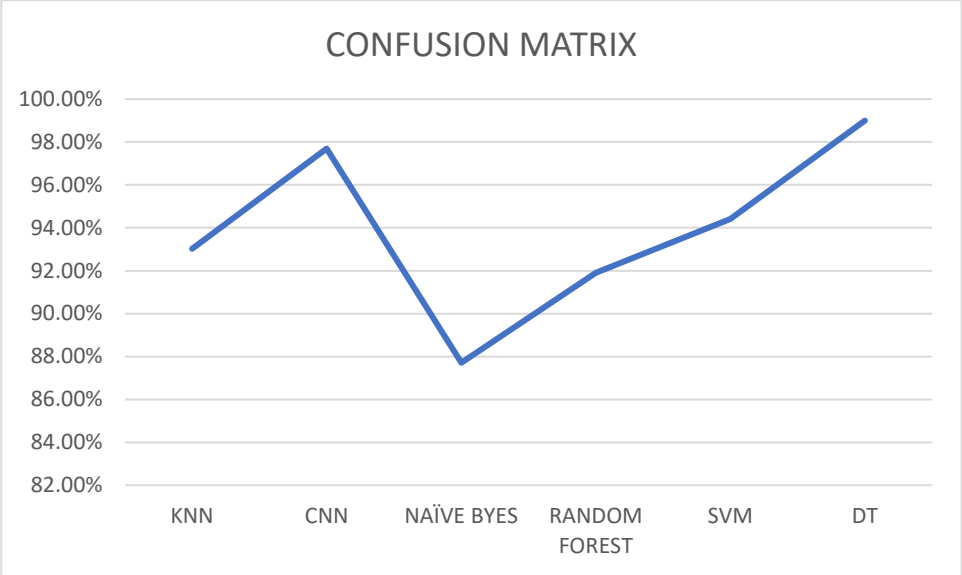
```
from sklearn.metrics import confusion_matrix

result = classif.predict(legit_test)
conf_mat = confusion_matrix(mal_test,result)

print(conf_mat)
```

```
[[19305    84]
 [   52  8169]]
```

(KNN, CNN, NB, RF, SVM, or DT) got smarter for each annotated data set. During the testing phase, a new collection of files containing malicious and non-malicious files was submitted to the classifier. A classifier determined whether the file was malicious or clean.

The proposed method for classifying and detecting malware was evaluated experimentally against collected malware and clean ware. Malware was investigated and characterized using supervised machine learning algorithms or classifiers (KNN, CNN, NB, RF, SVM, DT).

The classifier accuracy results (KNN = 93.02%, CNN = 97.7%, Naive Byes = 87.71%, Random Forest = 91.91%, SVM = 94.41%, DT = 99 %). We have shown that DT is the best model for malware detection strategies. Classifier TPR (%) (KNN = 93.17%, CNN = 99.22%, Naive Byes = 90%, Random Forest = 91.9%, SVM = 98%, DT = 99.07%) shows that CNN is the second-best model showed that SVM was the third best model for malware detection and identification. We assumed that CNN, SVM, DT, and ANN classifiers have all-around equally high accuracy and performance. Using the three best algorithms (DT = 99%, SVM = 94.41%, and CNN = 97.76%) with higher TPR rate (%) and accuracy, DT accuracy and DT performed best for discrimination.



7. Conclusion:

This research highlights the increased interest in ML algorithm solutions for malware identification among academics in recent times. We provided a safeguard that considered three ML algorithm approaches to malware detection and selected the best one. The findings demonstrate that DT (99%), CNN (97.76%), and SVM (94.41%) outperformed other classifiers in terms of detection accuracy. In a specific dataset, the malware detection performances of the DT, CNN, and SVM algorithms (DT = 2.01%, CNN = 3.97%, and SVM = 4.63%) were compared. The detection accuracy of a machine learning (ML) classifier that used static analysis to extract features based on PE data was assessed and quantified in this experiment by contrasting it to two other ML classifiers.

Our work has enabled machine learning algorithms to distinguish between harmful and benign data. Of all the classifiers we tested, the DT machine learning approach had the best accuracy (99%). Static analysis based on PE information and properly chosen data showed promise in trial results, possibly delivering the best detection accuracy and precisely characterizing malware. The fact that we can assess whether data are harmful without having to do anything is a big plus. Using the dataset, the three machine learning models (DT, CNN, and SVM) were trained, evaluated, and their efficacy was compared.

8. References

1. Nikam, U.V.; Deshmuh, V.M. Performance evaluation of machine learning classifiers in malware detection. In Proceedings of the 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, 23–24 April 2022.

2. Akhtar, M.S.; Feng, T. IOTA based anomaly detection machine learning in mobile sensing. *EAI Endorsed Trans. Create. Tech.* **2022**.

3. Sethi, K.; Kumar, R.; Sethi, L.; Bera, P.; Patra, P.K. A novel machine learning based malware detection and classification framework. In Proceedings of the 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Oxford, UK, 3–4 June 2019.

4. Abdulbasit, A.; Darem, F.A.G.; Al-Hashmi, A.A.; Abawajy, J.H.; Alanazi, S.M.; Al-Rezami, A.Y. An adaptive behavioral-based increamental batch learning malware variants detection model using concept drift detection and sequential deep learning. *IEEE Access* **2021**, *9*, 97180–

5. Chandrakala, D.; Sait, A.; Kiruthika, J.; Nivetha, R. Detection and classification of malware. In Proceedings of the 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 8–9 October 2021; pp. 1–3.

6. Gibert, D.; Mateu, C.; Planes, J.; Vicens, R. Using convolutional neural networks for classification of malware represented as images. *J. Comput. Virol. Hacking Tech.* **2019**, *15*, 15–28.

7. Firdaus, A.; Anuar, N.B.; Karim, A.; Faizal, M.; Razak, A. Discovering optimal features using static analysis and a genetic search based method for Android malware detection. *Front. Inf. Technol. Electron. Eng.* **2018**, *19*, 712–736.

8. Dahl, G.E.; Stokes, J.W.; Deng, L.; Yu, D.; Research, M. Large-scale Malware Classification Using Random Projections And Neural Networks. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing-1988, Vancouver, BC, Canada, 26–31 May 2013; pp. 3422–3426.

9. Akhtar, M.S.; Feng, T. An overview of the applications of artificial intelligence in cybersecurity. *EAI Endorsed Trans. Create. Tech.* **2021**, *8*.

10. Hamid, F. Enhancing malware detection with static analysis using machine learning. *Int. J. Res. Appl. Sci. Eng. Technol.* **2019**, *7*, 38–42.

11. Prabhat, K.; Gupta, G.P.; Tripathi, R. TP2SF: A trustworthy privacy-preserving secured framework for sustainable smart cities by leveraging blockchain and machine learning. *J. Syst. Archit.* **2021**, *115*, 101954.

12. Kumar, P.; Gupta, G.P.; Tripathi, R. A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks. *J. Ambient Intell. Human. Comput.* **2021**, *12*, 9555–9572.

13. Pavithra, J.; Josephin, F.J.S. Analyzing various machine learning algorithms for the classification of malwares. *IOP Conf. Ser. Mater. Sci. Eng.* **2020**, *993*, 012099