Comparison and Analysis of Various Machine Learning and Deep Learning Algorithms for Malware Detection System

[1]Amulya R, [2]Radhika KR

[1]Student, [2]Professor
Department of Information Science and Engineering,
BMS College of Engineering, Bangalore, India

***Abstract*- People depend on the Cyberspace for their everyday tasks, and it has long been a component of people's lives. The Internet has a lot of benefits and disadvantages. When a system is connected to the Internet, then those are prone to attacks. Few attacks are noticeable, but few go unnoticed. These unnoticed attacks have a huge impact and are not easy to detect or recognize. One among those is Malware attack. In this attack, the system tends to behave differently at a slow pace. There are numerous methodologies that could be utilized to find the dangerous software. As part of this study, various works are referred to and compared with their methodology and limitations.**

***Index Terms*- malware detection, cyber-crimes, machine learning, deep learning, malware**

I. INTRODUCTION

Computers are digital apparatuses that are aimed to do a whole range of arithmetic and logical tasks without human interference or with minimal intervention. Usually, it comprises hardware, software and operating systems. Computers have a broad range of usage.[1] It is used in industries, hospitals, schools and colleges, banks and other various sectors for multiple purposes. In industries, computers are utilized to store required data, operate the machines. Like that in education institutions its utilized to store the details of employees and students, in the laboratories for educational purposes. In banks or financial sectors, computers are majorly deployed for managing details and transactions. Attacks on these computers lead to high damages to the individuals and to businesses as well. Since computers are under Internet use, it's easy for attackers to sabotage the computers. As these attacks use the Internet, it's called Cyber-Crime. Cyber-Crimes, not only impacts on people or business but also on economy of the state or country.[2]There are several cyber crimes like phishing attack where the person gains the trust and obtains the information, spoofing which is the Spoofing is the practice of impersonating a legitimate entity by fabricating information in order to get access to resources or information that one is not otherwise permitted to use, eavesdropping which means to quietly listening to someone's talk without their consent . Among the committed crimes on computers is Malware attack. Malware shortly defined for Malicious Software is any software that is specifically designed for causing disturbances to a network, server or client, exposing private information, or by violating the confidentiality and safety of computer users, information can be accessed in an unethical manner. Malware falls into a huge variation of genres, including Trojan horses, spyware, worms, viruses, ransomware, adware, wiper, and keyloggers. Detection of these crimes are tricky and difficult. [3] The main goal of any malware is to hide itself from detection by the end-user or antivirus software . So the methods are usually analysis of operation, to determine whether any activity is blocked or not. The protective measures include installation of antivirus, updated software usage.

II. LITERATURE SURVEY

This research highlights the increasing attention in ML algorithm techniques for malware identification among academics in recent times.[3]They provided a safeguard methodology that considered three ML algorithm approaches to malware detection and selected the best one. The findings demonstrate that DT , CNN , and SVM with 99%, 98.76%, 96.41% respectively outperformed other classifier like Random Forest, KNN in the context of detection performance. There were no Deep Learning techniques used or referred. [4] Aimed at defining if a Portable Executable file is dangerous or not, a Machine Learning (ML)-based malware detection system is introduced. Using the executable's header, this system extracts features. To deal with the malware, a number of ML models are applied after the data has been preprocessed, including Random Forest, Support Vector Machine (SVM), Decision Tree, AdaBoost, Gaussian Naive Bayes (GNB), and Gradient Boosting. Also, comparison between ML models is done to determine which is best for the specified issue. As stated by the testing findings, the Random Forest fared better than the others in detecting malware, with an accuracy level of 99.44 percent. With this, it is possible to create a desktop application for the Windows platform that can be customized to check for malware. But this study was concentrated specifically, not broadly, and solely on the Windows platform.It was suggested to incorporate a revolutionary deep-learning-based architecture to categorize malware variants using a hybrid model. [5] The authors provide DL-FHMC, a fine-grained ordered learning approach for effective detection of IoT malware. Then they add Suspicious Behavior Detector, a module that accurately detects AEs by extracting detailed behavioral patterns from three well-known IoT malware families: Gafgyt, Mirai, and Tsunami. This module works as model-independent without any assumptions done before. However, this module's disadvantage is that it only detects three categories of malware and misses other risks like incurable infections and worm wars.For identifying the black-box aggressive threats on an industrial Internet of Things(IIoT), the authors of this specific paper ,present the stateful query analysis (SQA) method, which examines sequences of queries received by malware classifiers (IIoT). [6] The comparison encoder and the classifier, each built using convolutional neural networks, are two parts of the SQA pipeline.

Tracking the history of inquiries enables the system to spot adversarial scenarios and stop attacks in their tracks, in contrast to state-of-the-art techniques that seek to identify specific adversarial occurrences. Evaluations show that SQA is valid by detecting 93.1% of hostile cases across a wide range. The authors, however, provide the opinion that the detecting rate can be still optimized.In this research, the authors demonstrate how deep learning networks can further boost accuracy. [7]Deep learning improves automatic detection and categorization of malware variants because it provides superior categorization by building neural networks with a greater number of potentially different layers. In this study, they provide a methodology which includes extraction of multiple feature-sets from malware files, inclusion of system calls, operational codes, sections, and byte codes. They also look at the accuracy obtained respective features, demonstrating that the system call feature vector has the highest accuracy. The authors also discuss how deep learning approaches outperform more conventional machine learning techniques.

Former Android malware detection techniques, such as those that rely on signatures or monitor battery life, could miss more modern spyware. [8]As a result, the authors describe a cutting-edge technique for finding malicious software in Android apps that makes use of Gated Recurrent Units (GRU), a kind of (RNN). The (API) calls and permissions are two static features that they infuse from Android applications. The CICAndMal2017 dataset is used in training and evaluation. The testing results represent that the deep learning algorithm performs superior than other techniques like machine learning algorithms, with an accuracy of 98.2%. The use of a certain dataset is one of the restrictions. Only specific features were present in the dataset. So the malware intervention cannot be detected if the feature used in it was different from the specified in the dataset. In this paper,the authors suggest a method for categorizing visualized malware called VisMal, which offers very effective categorization with tolerable accuracy.[9] Using a contrast-limited dynamic histogram equalization approach, VisMal turns malicious programs into images with the purpose of improving the similarity across malware image regions belonging to the same family. Even though, the solid implementation is provided, the accuracy collected is not high.

This study examines the robustness against adversarial attack of twenty-four different malware detection models across four categories that were created using two features and twelve learning techniques . [10]The categories include machine learning, bagging classifiers, boosting classifiers, and neural network.The authors specifically uses GradAA and GreedAA, to reveal weaknesses. They additionally suggested two defensive tactics, namely Adversarial Retraining and Correlation Distillation Retraining, as defenses against adversarial attacks on detection models. Even though additional tactics were discussed, it was not robust.DenseNet, ResNet, InceptionResNet, and EfficientNet are the four model fusions. [11] In comparison to other fused models and non-fused CNN-based pretrained models, EfficientNet-based fused models performed better. Also, the fused models based on EfficientNet beat the current methodologies for detecting malware on Android. The suggested model demonstrated similar outcomes on both testing datasets while achieving superior results during testing and training phases. All model performances were demonstrated on two distinct testing datasets. This demonstrates how the suggested solution is more durable and generalizable and that it can be used to create a utility that can be downloaded from any app store. Even though the approach is resilient and generalized, it was not scalable and required multiple dependent packages to be installed and time consuming.The feature engineering procedure can be entirely avoided. Advanced machine learning approachs, such as Deep Learning, can accomplish this. From 2011 to 2021, the authors address the issue in 42 of the most referenced papers.[12] From this they extract the features used, models employed, dataset used. So using that information, the authors tend to provide improvements to be done to fill the missing space of information. Even though the review of models are done by the authors, the likeliest methods are not noticed. The authors created a fresh adversarial-example attack strategy based on the bi-objective GAN.[13] Tests reveal that their method outperforms the state-of-the-art technique by 247.68% in that over 95% of the adversarial samples it generates bypass the Android's firewall-equipped malware detection mechanism. Despite that, this technique was restricted to the Android platform exclusively but failed to perform in other platform like Windows.

In this investigation, the authors have suggested decentralizing the current cloud-based security architecture to neighborhood fog nodes for the purpose of create an anomaly-based intrusion detection system.[14] Several machine learning approaches, such as Random Forest, k-Nearest Neighbor, and Decision Tree, are utilized to assist the effectiveness of the intended model. Using real IoT-based datasets, our suggested model's performance is evaluated. The evaluation of the basic method outperforms utilising the Random Forest algorithm with regard to of a high detection rate alongside a low rate of false alarms. Again as discussed in the above paper, in this study too the authors have utilized the traditional methods for detection of conventional malwares.

Wildlife fires are one of the threats to the forests. This arises due to the rise in the environment's temperature. [15] The authors presented a distributed sensor network-based reduced power and reduced cost for wildfire monitoring system. The device integrated smoke and humidity sensors with a camera. Convolutional neural networks provide a practical method for deep learning to evaluate camera photos and identify the presence of smoke or a wildfire (CNNs). We train, validate, and assess the classificwildfireecision of the CNN classifiers using a sizable wildfire image set. In accordance with the findings, CNNs can accurately identify the presence of fire and smoke in photographs of wildfires. The only challenge is to distinguish between camera photos with and without fire or smoke.[17] The author developed a malware detection method by transforming malware files into a visual illustration and sorting the input photos using CNN. The spatial pyramid pooling layers (SPP) are used in the construction of the CNN to handle input of various sizes. They evaluate the effectiveness of our system using both unmodified and malicious data that has been subjected to duplicate API injection, as well as the efficacy of SPP and RGB and greyscale picture colour spaces. This was very much good for malware detection through images. But the only challenge was conversion of picture color to grayscale. If this was done incorrectly, the accuracy is effected. The two datasets the authors use for their experiments—8928 malware pattern from VXHeavens and 3293 data sets from manual analysis—are largely composed of packed malware. [18] The outcomes show that, using the hybrid traits they select, are able to categorize malicious samples into families with accuracy. Additionally, they can successfully extract family selected features by using stepwise selection from 37 feature categories. The grouping clustering algorithm is used for intrafamily clustering and found that family feature sets are significantly more precise than common feature sets, allowing for more precise lineage attribution of packed malware. Even though they suggested a way for choosing a cluster head, a cluster might not contain any unpacked samples. As a result, a packed sample from such a cluster will be chosen, which has significant drawbacks for lineage inference. Additionally, because their techniques are built on heuristics, some samples might not adhere to the criterion.

The protection against mobile malware and the high availability of Iot systems while their authentication is being done are the two key goals of PRoM.[19] The suggested method uses a randomized strategy and little secure hardware to find roaming malicious programs in IoT devices. With usage of Raspberry Pi configuration, the authors demonstrate an experimental analysis of the suggested technique. When compared to other attestation methods, PRoM has a substantially shorter execution time while yet having a larger likelihood of being discovered by an opponent. Furthermore, PRoM ensures high availability by not interfering with IoT devices' regular operations. It's not flexible and also not scalable to all devices.

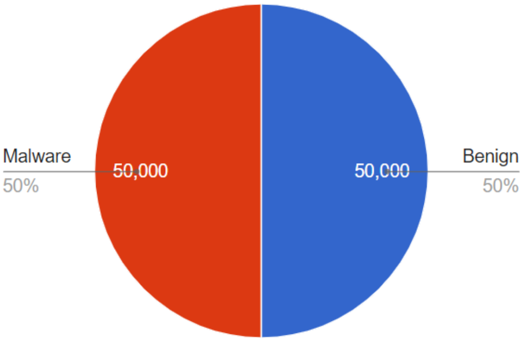
Complex deep learning algorithms are not tested, and neither is operational integrity.[20] This work performs two escape situation of attacks on the feature space of Android apks and builds feature preserved Android malware. Features that are omitted from harmful samples but present in genuine apps are injected into the malware samples. The sample created in this way will be statistically diverse but functionally same. Utilizing Euclidean distance (ED), that compares malware and benign samples, is one of the attack scenarios put into practise. The second method of approach involves creating variations via PSO.Even though two situations are discussed, the authors have not provided the insights about the chances and behaviors of model in other situations. In this paper[21], the main focus is to put forward a new hybrid architecture that has 4 wide-ranging already trained network models using a metaheuristic algorithm. This has four phases including dataset creation, DNN architecture design, enhancement of DNN architecture and estimation of the trained Deep neural network. This estimation of performance is rendered using benchmark datasets. The outcomes showed that human behaviours predictions were superior to the prior architecture, however this is not true for sophisticated tasks like computer vision and human-computer interaction.

III. **METHODOLOGY**

Dataset Description

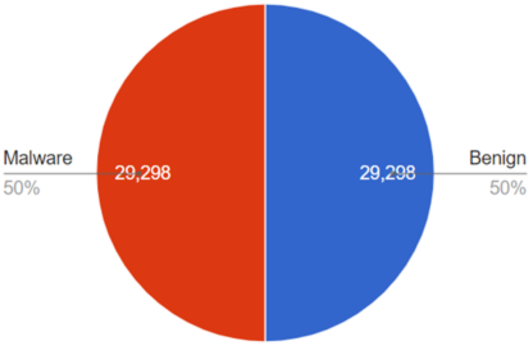
The first dataset is retrieved from Kaggle.[23]. This consists of data where the hash values of file is categorized as benign and malware. It has 1,00,000 data with 35 features. Figure 1's pie graph, which depicts the 50,000 malware and 50,000 benign samples in this dataset, provides an illustration.

Figure 1:Pie Chart of Malware-Benign classification for the dataset 1



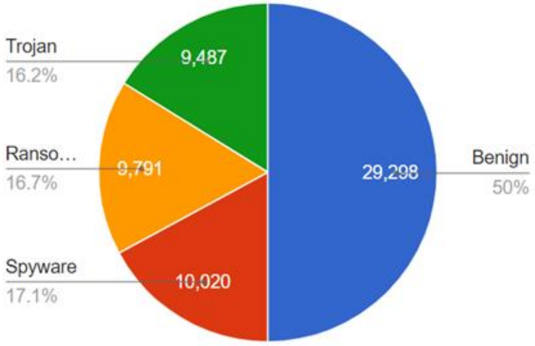
The second dataset adopted in this paper, is CIC-MalMem2022. It is an academic dataset that is released by “Canadian Institute for Cybersecurity” having the intention for researching in malware classification. It is an organised dataset that was created by extracting features from memory dumps. A total of 58,596 records—29,298 benign and 29,298 malicious and 57 features, make up the balanced dataset. Figure 2 can be employed to depict this.

Figure 2:Pie Chart of Malware-Benign classification for the dataset 2

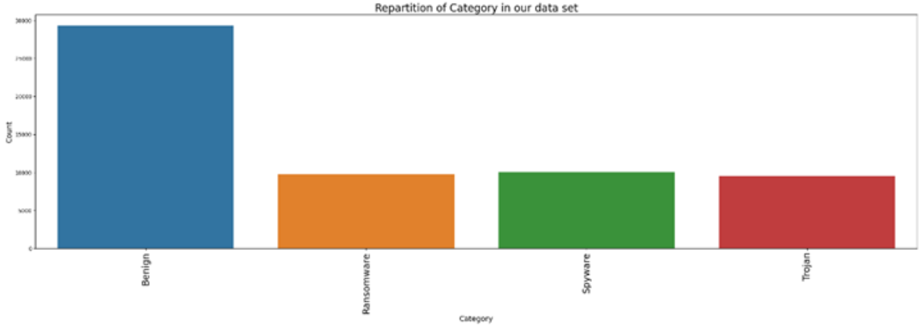


These Malicious instances are further distinguished into categories namely, Spyware, Ransomware and Trojan. So the dataset now contains 29,298 benign samples, 10020 spyware , 9791 ransomware and 9487 trojan samples. Consequently, this is depicted in figure 3 as a pie chart.

Figure 3:Pie chart of Malware categories-Benign Classification of dataset 2



The bar graph in figure 4 could also serve as a representation of the categories count.
Figure 4:Bar chart of Malware categories-Benign Classification of dataset 2



The three categories further comprises of 15 families that include: Spyware like 180Solutions, Coolwebsearch, Gator, Transponder, and TIBS; Trojan-Horse like Reconyc, Zeus, Emotet, Refroso, and Scar; and ransomware like Maze, Shade, Ako, and Pysa.

Comparative Study

Many researchers have used the first dataset in their journal. The comparative study is done on the basis of the specific dataset. So the comparative tables for various journals with the methods (strategies) they have used and the limitations is provided by table 1.[24][25][26][27][28]

Table 1:Comparative table for the dataset

Paper	Dataset used	Methodology	Accuracy	Limitations
Paper 1	Kaggle dataset by N.Saravana	Usage of four algorithms like Naïve Bayes, SVM, CNN and RNN with feature selection and non-feature selection	NB-69.56% SVM-98.61% CNN-99.67% RNN-99.96%	By using the features to their full potential, the machine-learning methods can be fine-tuned to generate superior outcomes.
Paper 2	Kaggle dataset by N Saravana Kaggle dataset by SHASHWAT TIWARI	Usage of Dense Model and LSTM for both the datasets	1st dataset- DM-99.96 %, LSTM-99.75% 2nd dataset: DM-98.38% LSTM-94.59%	was that we focused on the general malware detection task without going deep into the types of malware.
Paper 3	Kaggle dataset by N Saravana	The model was created using Logistic Regression, extreme gradient boost, LightGBM,	XGBoost-89.33 LR-84 LightGBM-93.33	The outcome for this system was not good and was comparatively low.
Paper 4	Kaggle dataset by N Saravana	K-Nearest Neighbors Algorithm (KNN) & Naive Bayes Algorithm (NB)	KNN-99.4% NB-62.8%	Only traditional methods are used.

Paper 5	Kaggle dataset by N Saravana	The innovative Naive Bayes Algorithm (NB) and the Logistic Regression Algorithm are the two categories in this work (LR). A sample group of 30 people makes up each group.	NB-61% LR-94%	It was just the comparison of the two machine-learning algorithms

Data Preprocessing

The datasets must go through some preprocessing for it to be appropriate for the classification. These stages are crucial for enhancing classification model performance and preparing the data for usage with machine-learning and deep-learning algorithms. Additionally, some procedures for data balance are accomplished against the overfitting issue, particularly in deep learning methods. Both the datasets used, contains classes named benign and malware. The datasets are balanced, which makes it robust to the overfitting issue. Hence, the overfitting issue in the study was not addressed. In this investigation, the Label-Encoder was employed to translate categorical class values into numerical values. With the assistance of Label Encoder the categorical value was replaced with numeric data which starts from 0. The samples having benign value was assigned with 0 and samples with malware was assigned with 1. This makes the datasets to be suitable for various ML and DL algorithms.

However, the CIC-MalMem 2022 dataset utilized in this study has categories namely Benign, Ransomware, Trojan and Spyware. Using the Label Encoder, these categorical values are replaced with numeric data that begins with 0. So Benign values are assigned with 0, Trojan values are assigned with 1, Ransomware values are assigned with 2 and Spyware values are assigned with 3.

Data Preparation

One of the crucial steps before using any ML or DL techniques on the dataset is data preparation. Data processing is another name for data preparation. The dataset is separated into X and Y, where X includes various features from which the classification must be made and Y includes the key feature. The dataset is then divided into two sets: a training set and a testing set. We divided 75:25, meaning that 75 percent must be utilized for training purpose and 25 percent must be used to test it.

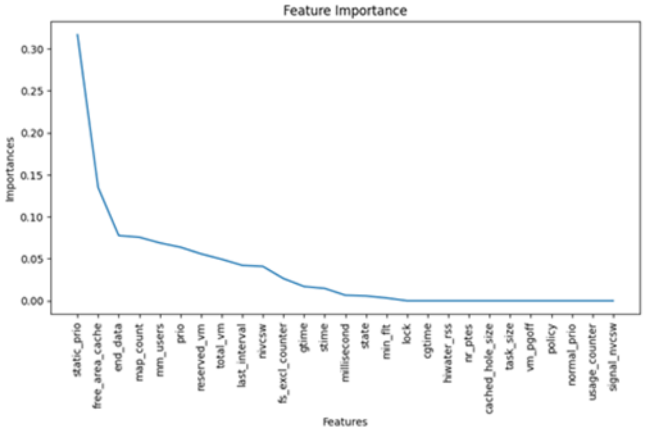
Machine-Learning and Deep-Learning Techniques

For this study, five Machine Learning approaches: KNN, Decision Tree, Random Forest, Naïve Bayes, XGBoost Classifier and two Deep Learning algorithms : Deep-Neural-Network and Multi- Layer-Perceptron are used. These techniques are used to build classification models using the features that were extracted. The success of the approach is demonstrated by how accurately the classification predictions were made. Heat maps are raster map representations of geographic data density that are created by applying a kernel density estimate of a specified radius to point or linear input data.[31] The amount of detail of heatmaps is reflected in the radius of the kernel estimator. The higher the radius, the more generalized the map is and the hotspots are blurrier. Generalization is critical, especially for non-interactive maps that cannot be dynamic when rescaled, this factor affects the effectiveness of web maps.

IV. RESULTS AND DISCUSSION

35 characteristics make up the bulk of the first dataset [23]. Some of them are given more weight than the others. In order to recognize the prominence of the features, the feature-importance graph is plotted. Figure 5 demonstrates this.

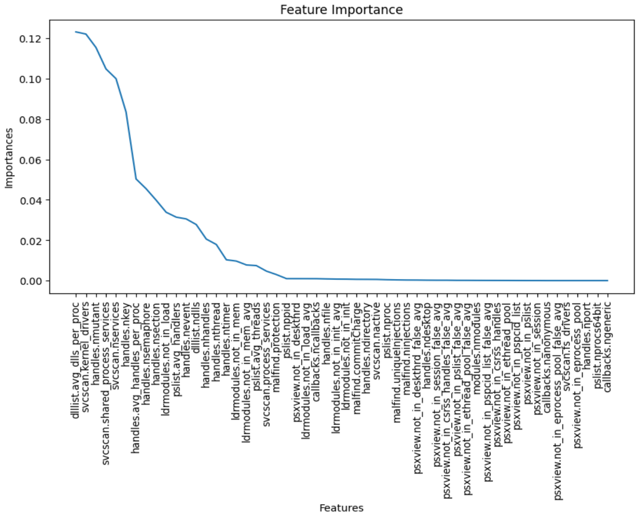
Figure 5:Feature Importance Graph of dataset 1



From the graph, we can have a conclusion that, feature static_prio has highest importance and if modified, then dataset is highly effected. Similarly, signal_nvcsw is least important and if modified will have no change on the data.

The dataset two has 57 features. To determine, the feature importance, a graph is plotted which is represented in figure 6.

Figure 6:Feature Importance Graph of dataset 2



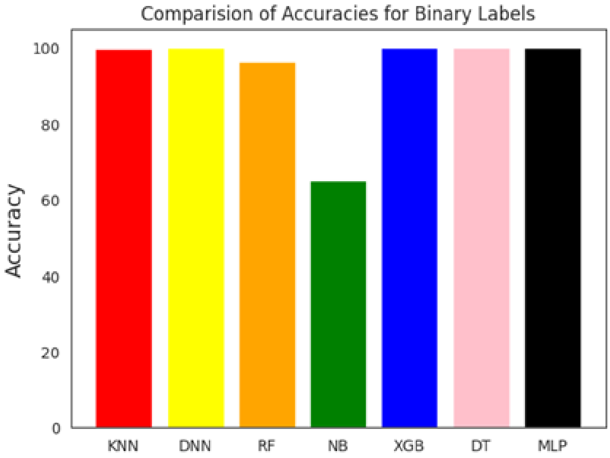
Dataset one is divided as 75:25 , where 75 percent of dataset is employed as training-set and 25 percent is employed as testing-set. So the seven algorithms are applied to the dataset and the comparative table is drawn from the accuracies obtained from the algorithms. As per the outcome, table 2 for dataset 1 contains the comparative table for the data on the basis of highest to lowest.

Table 2:Comparative table of algorithms for dataset 1

Algorithms	**Accuracy in %**
XGBoost Classifier	100
Decision Tress Classifier	100
Multi Layer Perceptron	99.995
Deep Neural Network	99.985
KNN	99.905
Random Forest Classifier	96.335
Naïve Bayes	65.155

From the table 2, we can clearly say that both XGBoost Classifier and DecisionTree Classifier has hundred percent accuracy, which means both these algorithms are best suitable and are efficient for the dataset. Naïve Bayes algorithm yields about 65 % accuracy making it not suitable and efficient for the dataset. These comparison of algorithms can be represented in terms of bar graph shown as figure 7.

Figure 7:Bar Graph for algorithm comparison for dataset 1



The dataset 2 is also categorised into 75:25, where 75 percent of dataset is employed as training-data and 25 percent of dataset is employed as testing-data. The dataset is analyzed separately as binary classification and as multi-class classification.

In binary classification, the dataset is organised based on Class feature which contains malware or benign. For this, seven methodologies are employed for identifying the suitable algorithm to be efficient for binary classification of the dataset. Table 3 contains a comparison table of algorithms.

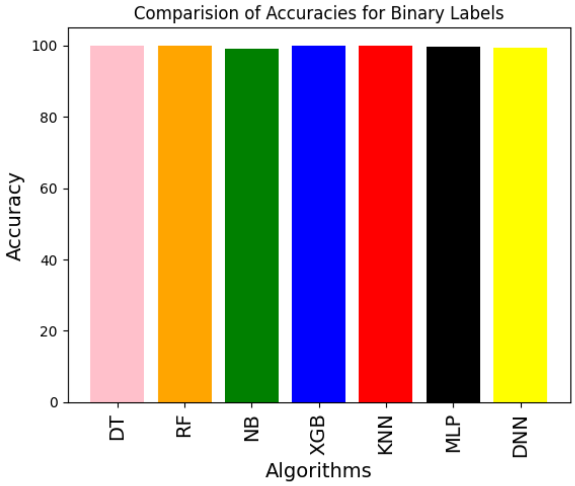
Table 3:Comparative table of algorithms for dataset 2 on binary classification

Algorithms	**Accuracy in %**

XGBoost	99.986
DecisionTree	99.980
Random Forest	99.980
KNN	99.890
Multi-Layer Perceptron	99.652
Deep-Neural Network	99.324
Naïve Bayes	99.126

From the table, we can know that three algorithms i.e., XGBoost with approximate 99.99 percent accuracy has highest value making it efficient algorithm in classifying the dataset with binary values. The bar graph in Figure 8 can depict the same comparison.

Figure 8:Bar Graph for algorithm comparison for dataset 2 on binary classification



From the above table and graph, we can say that all the used algorithms have efficiently classified the data and the accuracies only vary slightly. The XGBoost technique thus outperforms other algorithms by a little margin.

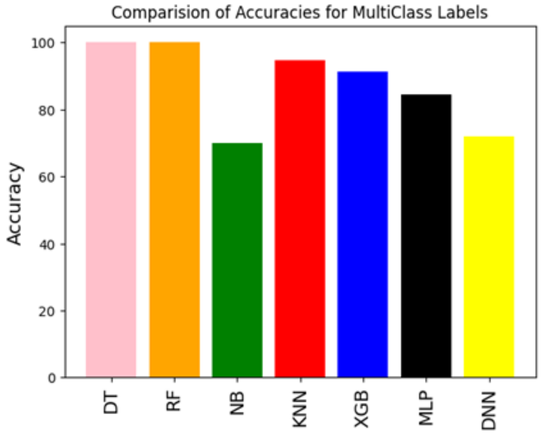
The dataset 2 can also be organised based on Multi-class values. The multiclass values are Spyware, Trojan, Ransomware and Benign. The seven methodologies are implemented to the dataset to determine the suitable algorithm that efficiently classify the dataset for multiclass values. The comparison table of algorithms is given table 4.

Table 4:Comparative table of algorithms for dataset 2 on multi-class classification

Algorithms	**Accuracy in %**
DecisionTree	99.986
RandomForest	99.986
KNN	94.744
XGBoost	91.310
Multi-Layer Perceptron	84.006
Deep-Neural Network	74.770
Naïve Bayes	70.114

From the comparative table, many insights can be obtained. Decision--Tree Classifier and RandomForest Classifier algorithms has highest accuracy making it most efficient algorithms for the multi-class classification on dataset 2. This comparison of procedures can be represented as bar graph in figure 9.

Figure 9:Bar Graph for algorithm comparison for dataset 2 on multi-class classification



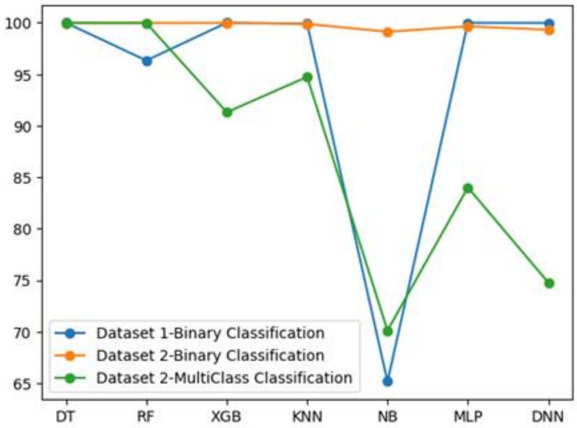
From the bar graph, we can also come to know that naïve bayes algorithm is not efficiently classifying the dataset on multi-class values.

Table 5:Comparative table of algorithms for both the datasets

Algorithm	Dataset 1-Binary Classification	Dataset 2-Binary Classification	Dataset 2-MultiClass Classification
Decision Tree	100	99.980	99.986
RandomForest	96.335	99.980	99.986
XGBoost	100	99.986	91.310
KNN	99.905	99.890	94.744
Naïve Bayes	65.155	99.126	70.114
Multi-Layer Perceptron	99.995	99.652	84.006
Deep-Neural Network	99.985	99.324	74.770

Table 5 represents the comparison of accuracies of various algorithms on both the datasets under different scenarios. We may conclude that Naive Bayes is the least efficient method because it has the lowest accuracy score across all three cases, while Decision-Tree Classifier has the highest accuracy scores across all three scenarios. This comparison of algorithms' performance can be represented in form of line graph shown in figure 10.

Figure 10:Line Graph representation of Comparison of algorithms for both the datasets under three scenarios



V. CONCLUSION

Because of the above study, we can conclude that performance of algorithms relies on the dataset. DecisionTree Classifier performs uniformly for both the datasets under three scenarios. Similarly, Naïve Bayes algorithm performs differently in all the three scenarios as illustrated in figure 10. Likewise, Multi Layer Perceptron and Deep Neural Network algorithms performs consistently in classifying binary values for both the datasets but vary in classifying the datasets on multiclass values. So, we can conclude that both these algorithms are unsuitable for multi-class classifications. Naïve Bayes algorithm performs better in binary classification for second dataset but has still lower value than other. We can conclude that DecisionTree Classifier is a better classification algorithm, from this study.

The first dataset has large data but a smaller number of attributes but it can only be categorized on the basis of benign and malware. There are no types and hence dataset cannot be classified based on the types. In contradiction, second dataset has limited data but more characteristics and the dataset can be classified for both binary values and multi-class values. From the comparative study for first dataset, we had obtained insights that most employed algorithms i.e., Naïve Bayes and Logistic Regression algorithms were outperformed by many algorithms. But DecisionTree classifier algorithm was not used by any of the authors that outperformed the outcomes of many of the methodologies used by them.

REFERENCES:

1. F. Zhong, "Malware-on-the-Brain: Illuminating malware Byte Codes with Images for malware classification in IEEE Transactions on Computers", vol. 72, 2023.
2. M. Cooper, "How Cyber Crime Damages Lives", ITNOW, vol. 62, no. 1, pp. 36–37, Feb. 2020.
3. Akhtar, M.S, Feng, T, "Malware Analysis and Detection using Machine Learning Algorithms", Symmetry 2022, 14, 2304, Nov. 2022.
4. A. Hussain, M. Asif, M. B. Ahmad, T. Mahmood, M. A. Raza, "Malware Detection using Machine Learning algorithms for Windows Platform", Lecture Notes in Networks and Systems, pp. 619–632, 2022.
5. A. Abusnaina., "DL-FHMC: Deep Learning-Based Fine-Grained Hierarchical Learning approach for Robust Malware Classification", IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 5, pp. 3432–3447, Sep. 2022.
6. B. Esmaeili, A. Azmoodeh, A. Dehghantanha, H. Karimipour, B. Zolfaghari, and M. Hammoudeh, "IIoT Deep Malware Threat Hunting: From Adversarial Example Detection to Adversarial Scenario Detection", IEEE Transactions on Industrial Informatics, vol. 18, no. 12, pp. 8477–8486, Dec. 2022.
7. R. Patil ,W. Deng, "Malware Analysis using Machine Learning and Deep Learning techniques", 2020 SoutheastCon, Mar. 2020.
8. Omar N. Elayan , Ahmad M. Mustafa, "Android Malware Detection Using Deep Learning" ,Procedia Computer Science, vol. 184, pp. 847-852, 2021.
9. F. Zhong, Z. Chen, M. Xu, G. Zhang, D. Yu, , X. Cheng, "Malware-on-the-Brain: Illuminating Malware Byte Codes With Images for Malware Classification", IEEE Transactions on Computers, vol. 72, no. 2, pp. 438–451, Feb. 2023.
10. H. Rathore, A. Samavedhi, S. K. Sahay, M. Sewak, "Towards Adversarially Superior Malware detection Models: An Adversary Aware Proactive Approach using Adversarial Attacks and Defenses", Information Systems Frontiers, Sep. 2022.
11. V. Ravi , R. Chaganti, "EfficientNet deep learning meta-classifier approach for image-based android malware detection," Multimedia Tools and Applications, Dec. 2022.
12. Mehrabi Koushki, M., AbuAlhaol, I., Raju, A.D., "On building machinelearning pipeline for Android malware detection: a procedural survey of practices, challenges and opportunities", Cybersecurity 5, 16, 2022.
13. H. Li, S. Zhou, W. Yuan, J. Li, H. Leung, "Adversarial-Example Attacks Toward Android Malware Detection System", IEEE Systems Journal, vol. 14, no. 1, pp. 653–656, Mar. 2020.
14. Prabhat Kumar, G. P. Gupta, and R. Tripathi, "Design of Anomaly-Based-Intrusion-Detection System Using Fog Computing for IoT Network", Automatic Control and Computer Sciences, vol. 55, no. 2, pp. 137–147, Mar. 2021
15. M. S. Khan, R. Patil, S. Ali Haider, "Application of Convolutional Neural Networks For Wild Fire Detection", 2020 SoutheastCon, Mar. 2020."
16. S. Srihari, "[PDF] Convolutional Networks | Semantic Scholar", [PDF] Convolutional Networks | Semantic Scholar, Jan. 01, 2022.
17. K. He , D.-S. Kim, "Malware Detection with Malware Images using Deep-Learning Techniques", Trust, Security and Privacy in Computing and Communications, Jan. 2019.
18. L. H. Park, J. Yu, H.-K. Kang, T. Lee, T. Kwon, "Birds of a Feature: Intrafamily Clustering for Version Identification of Packed Malware", IEEE Systems Journal, vol. 14, no. 3, pp. 4545–4556, Sep. 2020.
19. M. N. Aman et al., "PRoM: Passive Remote Attestation Against roving Malware in Multicore IoT Devices", IEEE Systems Journal, vol. 16, no. 1, pp. 789–800, Mar. 2022.
20. R. G, V. P, A. S, "Evading Machine-Learning based Android Malware Detector for IoT Devices", IEEE Systems Journal, pp. 1–11, 2022.
21. R. Komatwar , M. Kokare, "RETRACTED ARTICLE: A Survey on Malware-Detection and Classification", Journal of Applied Security Research, vol. 16, no. 3, pp. 390–420, Aug. 2020.
22. F. Cui, Q. Cui, Y. Song, "A Survey on Learning-Based Approaches for Modeling and Classification of Human–Machine Dialog Systems", IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 4, pp. 1418–1432, Apr. 2021.
23. "N SARAVANA. Malware Detection|Kaggle. 2018. Available online: https://www.kaggle.com/datasets/nsaravana/malwaredetection?select=Malware+dataset.csv."
24. "Malware Memory Analysis | Datasets | Canadian Institute for Cybersecurity | UNB, Malware Memory Analysis | Datasets | Canadian Institute for Cybersecurity | UNB. https://www.unb.ca/cic/datasets/malmem-2022.html"
25. B. A. S. Dilhara, "Classification of Malware using Machine-learning and Deep-learning Techniques", International Journal of Computer Applications, vol. 183, no. 32, pp. 12–17, Oct. 2021.
26. E. S. Alomari et al., "Malware Detection Using Deep Learning and Correlation-Based Feature Selection", Symmetry, vol. 15, no. 1, p. 123, Jan. 2023.
27. D. A. . Kumar , S. K. . Das, "Machine-Learning Approach for Malware-Detection and Classification Using Malware Analysis Framework", Int J Intell Syst Appl Eng, vol. 11, no. 1, pp. 330–338, Feb. 2023.
28. P. Borra Madhan Mohan Reddy, "Detection of Malware in Cloud Storage Data using Naive Bayes", BALTIC JOURNAL OF LAW & POLITICS, vol. 15, no. 4, pp. 458-465, 2022.

29. P. Borra Madhan Mohan Reddy, “Detection of Malware Attacks Using Naive Bayes Algorithm Comparing Logistic Regression Algorithm to have Improved Accuracy Rate”, BALTIC JOURNAL OF LAW & POLITICS, vol. 15, no. 4, pp. 458-465, 2022.
30. W. Jia, X. Liu, Y. Wang, W. Pedrycz, J. Zhou, “Semisupervised Learning via Axiomatic Fuzzy Set Theory and SVM”, IEEE Transactions on Cybernetics, vol. 52, no. 6, pp. 4661–4674, Jun. 2022.
31. K. Słomska-Przech, T. Panecki, W. Pokojski, “Heat Maps: Perfect Maps for Quick Reading? Comparing Usability of Heat Maps with Different Levels of Generalization”, ISPRS International Journal of Geo-Information, vol. 10, no. 8, p. 562, Aug. 2021.