**Mini Project Report on**



**MALWARE DETECTION AND MITIGATION**  


**Submitted in partial fulfilment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name:-** **SWETA VARSHNEY**

***Under the Mentorship of***

**Prof. (Dr.) Mohammad Wazid**

**Designation:- Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**January 2023**

A white background with red and black text

Description automatically generated

**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Malware attack detection and mitigation”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Prof. (Dr.) Mohammad Wazid, Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

**Name:-SWETA VARSHNEY**

**Signature:-**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **4-6** |
| Chapter 2 | Literature Survey | **7-8** |
| Chapter 3 | Methodology | **9-12** |
| Chapter 4 | Result and Discussion | **13-16** |
| Chapter 5 | Conclusion and Future Work | **17-19** |
|  | References | **20** |

**Chapter 1: -**

**Introduction**



**Malware**, short for malicious software, is a general term for any software or code that has been created with the intent to compromise, harm, or provide unauthorized access to computer networks, systems, or devices. It includes a wide variety of harmful software, including as viruses, worms, Trojan horses, ransomware, spyware, adware, and more. Cybercriminals design and spread malware with the goal of exploiting security flaws, stealing confidential data, interfering with business processes, or taking over systems without authorization.

Malware's main objective is to jeopardise the security and integrity of the targeted system or network, frequently without the user's knowledge or consent. It can spread via a number of channels, including email attachments, compromised websites, portable media, and software flaws. Malware can carry out its malicious actions after installation or execution, which may include:-

Data theft, System disruption, ransomware attack, adware and spyware and keylogging and screen capture

It is essential for people, organizations, and cybersecurity experts to remain aware and proactive in protecting against malware attacks as the landscape of cyber threats continues to change. To reduce the hazards brought on by malware, it is crucial to build strong security procedures, conduct regular security awareness training, and stay current with security upgrades.

**Introduction 1.1 : -**

**What is Machine Learning?**

Machine learning (ML) is a field of study focused on comprehending and developing "learning" methods, or methods that use data to enhance performance on a particular set of tasks. It is considered to be a component of artificial intelligence.

Without being explicitly programmed to do so, machine learning algorithms create a model from sample data, also referred to as training data, in order to make predictions or decisions.

Machine learning algorithms are used in a wide range of applications, including computer vision, speech recognition, email filtering, medicine, and agriculture, where it is challenging or impractical to create conventional algorithms that can perform the required tasks.

**How can machine learning be used to detect hate speech?**

****

Modern malware detection and prevention mostly depend on machine learning. It has developed into a useful tool for cybersecurity experts to find and categorise malware, spot suspicious trends, and create preventative defences.

Malware classification is a crucial area where machine learning is used in malware detection. Large datasets of known malware samples are used to train machine learning algorithms so that these models can understand the patterns and traits that separate harmful code from good software. Machine learning models can accurately categorise files or processes as malware or non-malware by analysing factors such as file behaviour, code structure, API calls, or network traffic.

**Algorithms I used for malware detections are:-**

1, **Random Forest:** Random Forest is a popular machine-learning algorithm used in malware detection. It operates by constructing multiple decision trees and combining their predictions to make a final classification. Each decision tree in the forest is trained on a different subset of the data and features, making it robust against overfitting. Random Forest can handle high-dimensional feature spaces and is effective in identifying important features for malware classification.

**Logistic Regression:** Logistic Regression is a statistical modelling technique commonly used in binary classification problems, including malware detection. It estimates the probability of a sample belonging to a particular class based on the input features. In malware detection, logistic regression can be used to model the relationship between the features extracted from malware samples and the probability of being classified as malware. It provides interpretable coefficients that indicate the impact of each feature on the classification decision.

**Support Vector Machines (SVM):** Support Vector Machines is a powerful supervised learning algorithm used for classification tasks, including malware detection. SVM works by finding an optimal hyperplane that separates the data points into different classes with the largest margin. In malware detection, SVM can be trained on labelled samples to build a decision boundary that separates malware from non-malware samples. It can handle high-dimensional data and is effective in dealing with both linear and non-linear relationships between features.

**Neural Networks:** Neural Networks, particularly deep learning architectures, have gained significant popularity in various domains, including malware detection. They consist of interconnected layers of artificial neurons that can learn complex patterns and relationships in the data. In malware detection, neural networks can be trained on large-scale datasets to automatically extract features and classify malware samples accurately. Deep learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have shown promising results in detecting malware based on file content, code sequences, or network traffic.

Each of these machine learning methods has advantages and disadvantages when it comes to detecting malware. Analysis of robustness and feature importance is provided by Random Forest. An interpretable coefficient is provided via logistic regression. Finding decision boundaries with high margins is an area where SVM excels. Deep learning neural network models, in particular, can learn intricate patterns. The algorithm chosen will depend on the particular requirements of the malware detection task as well as the type of data that is available. In real-world applications, ensembles or combinations of these algorithms are frequently employed to take use of each algorithm's specific advantages and boost performance.

**Chapter 2: -**

**Literature Survey**

**![Shape

Description automatically generated with low confidence](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAADUAAAAwCAMAAACG9cLxAAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAASUExURQAAAAAAAAAAAAAAAAAAAAAAAOArGaIAAAAFdFJOUwAeH77+RZSefQAAAAlwSFlzAAAh1QAAIdUBBJy0nQAAAJRJREFUSEvtlMsSQDAQBOP1/78spJVEdq3sCZW+iJnpohwEJwvXNrqV47RcWrdy5v9+Do82vf8VfdZHHmZoUj2Ymthv1q0m95a299VATk9Sr2iqN9JfB6SaSKlrEaKd+IsoIQdCG/ZAaML8gNSAcQbFHSxL6FSYVVDLsBFhUkOvww7SLdVz/BbHBlxStDg04ZIyQlgBkdMIcGaba4IAAAAASUVORK5CYII=) Literature on Malware detection and Mitigation:**

1. "Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code" by Michael Ligh, Steven Adair, Blake Hartstein, and Matthew Richard: This book provides practical guidance and hands-on techniques for analyzing and dissecting malware. It covers a range of topics, including malware behavior analysis, code disassembly, reverse engineering, and effective countermeasures.
2. "The Art of Computer Virus Research and Defense" by Peter Szor: This book provides an in-depth exploration of computer viruses, their history, evolution, and defensive strategies. It covers topics such as virus behavior, propagation techniques, antivirus technologies, and the challenges faced in the battle against malware.
3. "Malware: Fighting Malicious Code" by Ed Skoudis, Lenny Zeltser, and Michael Gregg: This comprehensive book covers various aspects of malware, including its types, behavior, propagation methods, detection techniques, and incident response. It offers practical insights into malware analysis, network defense, and effective countermeasures.
4. "Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software" by Michael Sikorski and Andrew Honig: This book offers a hands-on introduction to malware analysis, covering the tools, techniques, and methodologies used to dissect and analyze malicious software. It includes real-world case studies and step-by-step analysis walkthroughs.

* **Famous articles on hate speech detection: -**

1. "End-to-End Machine Learning Models for Malware Detection and Classification" by Lei Wu, Heng Yin, and Zhiqiang Lin: This article proposes an end-to-end machine learning approach for malware detection and classification. It introduces a framework

that combines feature extraction, feature selection, and classification algorithms to build effective models for malware detection and classification tasks.

1. "Hierarchical Classification of Malware Based on Machine Learning Algorithms" by Abu Bakar Siddique, Muddesar Iqbal, and Siddique Ullah Khan:This article proposes a hierarchical classification approach for malware detection using machine learning algorithms. It introduces a system that can classify malware into multiple levels of categories, providing granular insights into the characteristics and behaviors of different malware strains.
2. Malware Detection and Classification Using Machine Learning Algorithms" by Sarfaraz Hussein, Gaurav Kumar, and Mohammad S. Alam:

This article presents a comprehensive study on the application of machine learning algorithms for malware detection and classification. It evaluates the performance of various classifiers, including random forests, support vector machines, and neural networks, in accurately identifying and classifying malware samples.

1. DeepDGA: Adversarially-Tuned Domain Generation and Detection" by Hyrum S. Anderson, Andrew Rothman, Paul Pearce, and Michael Hicks:

This article explores the use of deep learning techniques, specifically recurrent neural networks (RNNs), for detecting and mitigating domain generation algorithms (DGAs) used by malware. It presents DeepDGA, a model that can effectively identify and block malicious domain requests generated by DGAs.

**Chapter 3: -**

**Methodology**

* **Important libraries of the code: -**

1. **Panda**:- Pandas is a data analysis and manipulation software package created for the Python programming language. It includes specific data structures and procedures for working with time series and mathematical tables.
2. **Numpy: -** The Python package NumPy is used to manipulate arrays. Additionally, it has matrices and linear algebra-related functions. You may use it without restriction as it is an open-source project. Numerical Python is referred to as NumPy.
3. **Train test split: -** The train test split() method is used to divide our data into train and test sets. Our data must first be divided into features (X) and labels (y). The data frame is divided into X train, X test, Y train, and Y test. Using the X train and Y train sets, the model is trained and fitted.



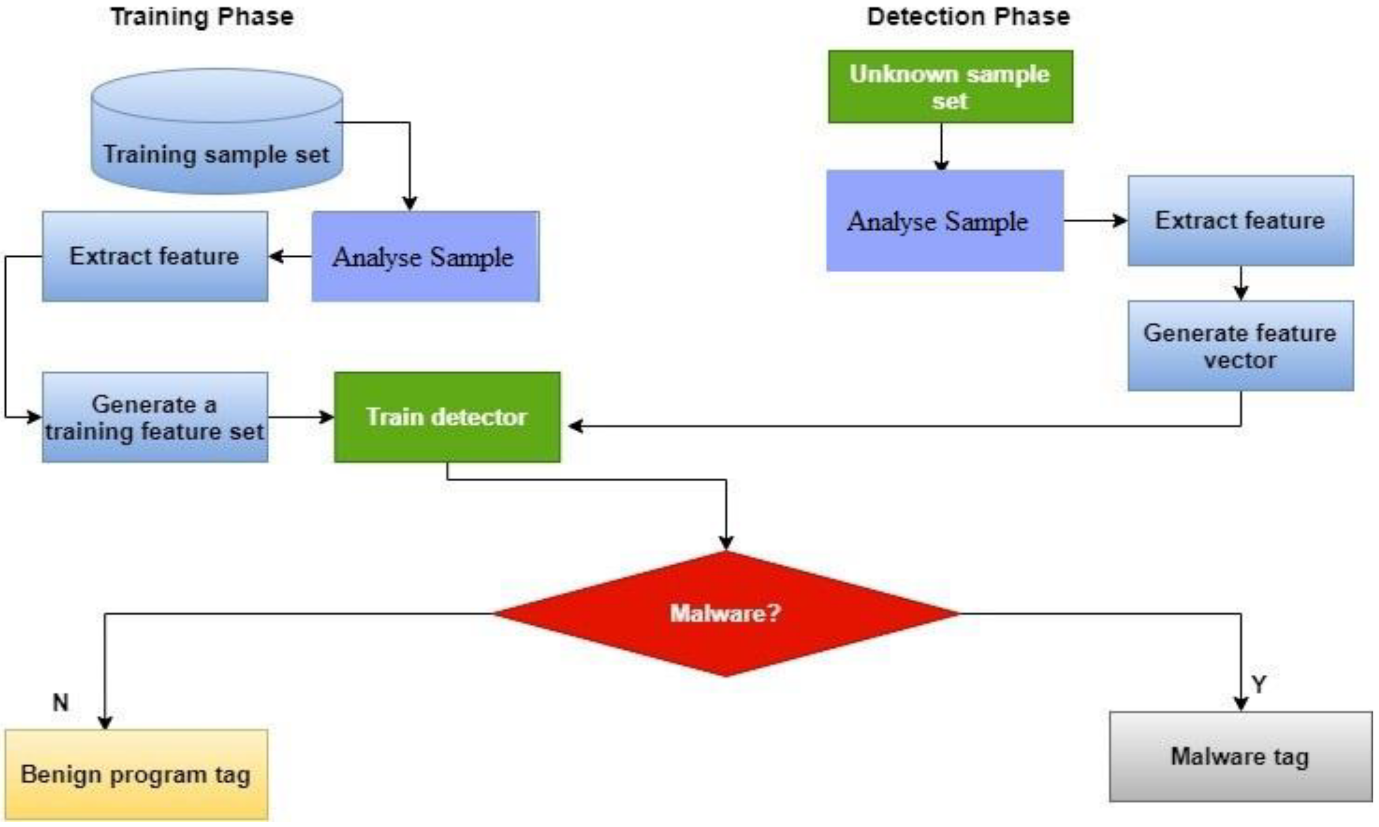
1. **Matplotlib:**- For the Python programming language and its NumPy numerical mathematics extension, Matplotlib is a graphing library. For integrating charts into programmes utilising all-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK, it offers an object-oriented API.

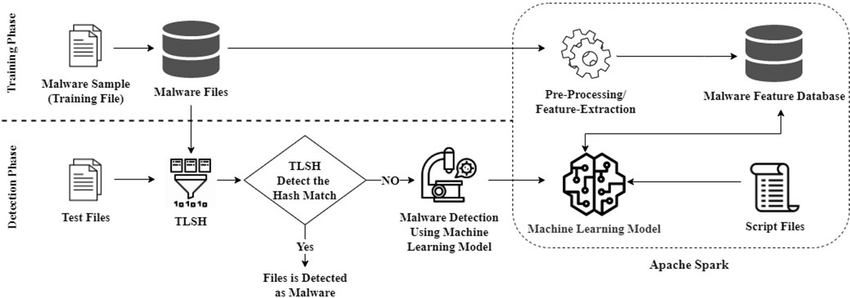
Line plots, scatter plots, bar plots, histograms, 3D plots, and many more types of graphs can be made with Matplotlib.

1. **Logistic regression:-** A statistical technique for categorization issues is logistic regression. In logistic regression, the probability of a binary result is predicted using a linear combination of input features (0 or 1). The class or label of the input is then predicted using this probability.
2. **SVC:-** Support Vector Machines (SVM) are powerful machine learning algorithms used for classification tasks. They find an optimal hyperplane that separates different classes with the largest possible margin. SVMs are widely used in malware detection, image classification, and other domains due to their ability to handle high-dimensional data and nonlinear relationships between features.
3. **Random Forest Classifier:-** The Random Forest classifier is a popular machine learning algorithm that combines multiple decision trees to make predictions. It operates by aggregating the results of individual trees to reach a final classification. Random Forest is effective in handling high-dimensional data, reducing overfitting, and providing feature importance analysis.
4. **Tensorflow:-** TensorFlow is a popular open-source library for machine learning and deep learning tasks. It provides a flexible and efficient framework for building and training neural networks. With its extensive ecosystem and GPU acceleration support, TensorFlow enables developers to implement complex neural network architectures and achieve high-performance results.

* **the methodology for the code: -**

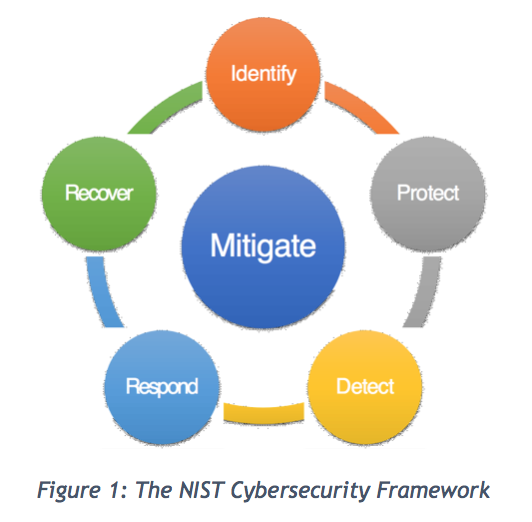
1. The first step is to import the necessary libraries such as pandas, numpy, nltk, matplotlib, SVC, logistic regression, Random forest classifiers, tensoflow ,train\_test\_split, decision tree from sklearn.
2. **Data Collection:-** Next, we use pandas library to read a CSV file from a local file and convert it into a pandas Data Frame. This dataset contains supervised text which will be used to train the model. .We have 2 dataset one we will use for training our dataset and other one is for checking mititgation.
3. **Data Preprocessing:-** The dataset should be cleaned and preprocessed to eliminate unnecessary data, standardize data formats, and transform binary data into appropriate numerical representations that machine learning algorithms can process. In order to effectively depict the malware samples, feature extraction is a crucial step.
4. **Feauture engineering:**- Engineering characteristics that can help distinguish between safe and dangerous files is important. API calls, file header data, strings, code snippets, and behavioral patterns are examples of common features.
5. **Model selection:**- Choose an appropriate machine learning algorithm for the task. I have included include Support Vector Machines (SVM), Random Forests, logistic regression and Neural Networks (RNNs).
6. **Model training:**- Create training and validation sets from the preprocessed data. Input the training data into the model of choice, then tweak its parameters to improve performance. Based on the given attributes, the model should be able to distinguish between good and bad examples.
7. **Performance evaluation:-** Assess the trained model's performance using the validation set. Common evaluation metrics include accuracy, precision, confusion matrix, F1-score.
8. **Test on unseen data:-** When you are satisfied with the model's performance on the validation set, test it on a brand-new dataset that has never been used before to gauge how well it performs in practise.





* **Mitigation steps:-**

1. **Adversarial Robustness:-** Adversarial Robustness: Adversarial attacks entail changing the input data to trick the machine learning model into identifying malware incorrectly. Use methods like adversarial training, which strengthens the model's defences against such attacks by supplementing the training data with adversarial cases, to reduce this.
2. **Ensemble methods:** To increase detection accuracy and decrease false positives, combine many machine learning models (e.g., SVM, Random Forest, Neural Networks) into an ensemble. The answer might be more dependable and robust when using ensemble approaches.
3. **Improper Data Handling:** Malware datasets are frequently unbalanced, with the bulk of samples being clean and the malware making up a small minority. To solve the class disparity and avoid biassed models, employ strategies like oversampling, undersampling, or the use of synthetic data.
4. **Confidence Thresholding:-** In order to balance false positives and false negatives, choose suitable confidence criteria for model predictions. Setting the threshold precisely can assist manage the trade-off between detection precision and resource usage.
5. **Continuous Model Updates:** To keep the machine learning model up to date and effective against new malware threats, update it frequently with new data.



**Chapter 4:**

**Result And Discussion**

The specific dataset and the mode's performance would determine the results and discussion for the code.

* **Result:-**

We used 4 algorithms for malware detection and mitigations.Those four are Random forest classifier, SVM, logistic regression, neural network.

We calculated accuracy and created confusion matrix for all four algorithm.

Accuracy for Random forest is 96%, logistic regression is 95%, for SVM it is 94% and for neural network is 66%.

* **Discussion:-**

On the basis of the outcomes, the model's performance would be examined during the discussion. If the accuracy is high, the model can correctly identify whether the link is malware or otherwise. Low accuracy indicates that the model cannot accurately classify the text.

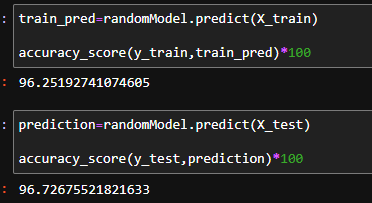
**Bar-chart of different classes in data set:-**

A yellow rectangular object with numbers

Description automatically generated

* **Random Forest classifier:-**

**Accuracy:-**

****

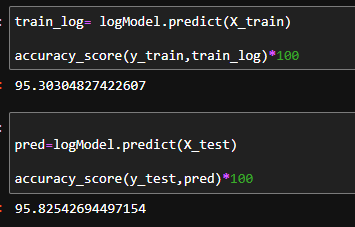
**Classification report of the data set:-**

**A screenshot of a computer

Description automatically generated**

* **Logistic regression:-**

**Accuracy:-**



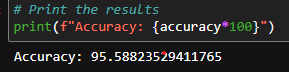
**Classification report of the data set:-**

**A screenshot of a computer

Description automatically generated**

* **SVM:-**

**Accuracy:-**

****

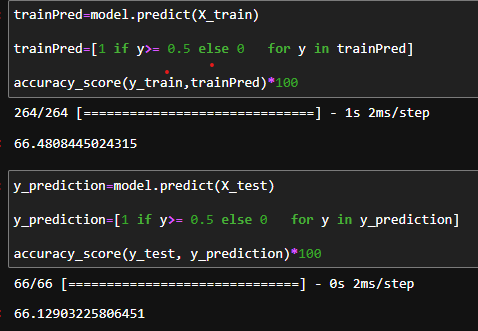
**Classification report of the data set:-**

**A screenshot of a computer

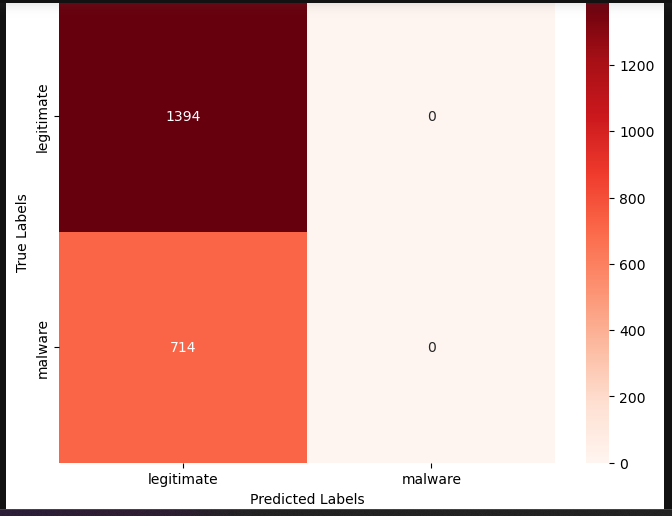
Description automatically generated**

* **Neural Network:-**

**Accuracy:-**

****

**Classification report of the data set:-**

****

**Chapter 5: -**

**Conclusion And Future Work**

* **Conclusion:-**

**Random forest:-**

1 An excellent machine learning approach for detecting malware is called Random Forest.

2. Due to the decision tree's decision ensemble nature, accuracy is increased and overfitting is decreased.

3. Large datasets and high-dimensional characteristics are both effectively handled by Random Forest.

4. It offers insightful information on the significance of features, assisting in the comprehension of malware traits.

5. Random Forest is a dependable option for creating trustworthy virus detection systems due to its durability and scalability.

**Logistic regression:-**

1. The straightforward and understandable approach known as logistic regression offers insights into how malware is classified.
2. When data is few, it is a good option because it performs well with smaller datasets.
3. Binary classification tasks can be successfully handled using logistic regression, which can distinguish between benign and malicious samples.
4. Despite its ease of use, Logistic Regression can produce competitive performance with careful feature engineering.
5. However, more complex algorithms like neural networks might be investigated for non-linear, extremely complex data to increase accuracy.

**SVM:-**

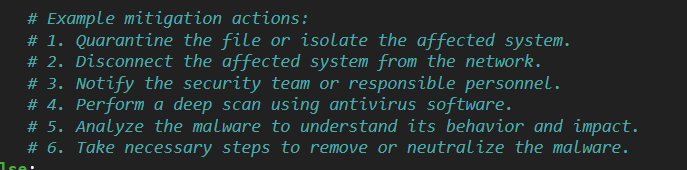
1. SVM is a powerful binary classifier for malware detection since it distinguishes between dangerous and legitimate software with high performance.
2. SVM can handle high-dimensional and complicated data with the right feature creation and tuning, making it possible for it to capture subtle trends in malware features.
3. SVM's capacity to choose the best hyperplane for maximum separating the two classes enables reliable and accurate detection of malware that hasn't been seen before.
4. Security professionals can grasp the decision-making process thanks to SVM's interpretability, which makes it easier to pinpoint individual attributes that contribute to malware identification.
5. SVM can be deployed in malware detection systems in real-time or close to real-time due to its computational efficiency.

**Neural network:-**

1. utilises the capacity of deep learning to automatically extract sophisticated features from raw data.
2. shows cutting-edge performance in identifying numerous malware kinds.
3. For efficient training, a sizable amount of labelled data is needed.
4. High accuracy is provided, however training may be computationally expensive.
5. offers a promising method for effective and reliable malware detection in practical cybersecurity applications.

* **Future work:-**

**Mitigation steps:-**



We can improve the efficiency of our model by:-

1. Use a larger and more varied dataset: The quality and diversity of the training data are two of the most critical aspects of a malware detection system performance. The accuracy and robustness of the detector can be enhanced by employing a larger and more varied dataset that contains a variety of different datasets.
2. Investigate and improve the application of cutting-edge deep learning architectures, like Transformers or Graph Neural Networks, for malware detection. These architectures have demonstrated promise in a number of fields and may enable advancements in malware representation and feature extraction.
3. Focus on creating compact and effective machine learning models that can deliver real-time malware detection without having a substantial negative influence on system resources.
4. Multi-Modal Data merging: Examine the merging of many data sources to increase the precision of malware detection and decrease false positives.
5. I can work more on various mitigation techniques, to handle the the malware more appropriately.



References

1. Multi-Modal Data merging: Examine the merging of many data sources to increase the precision of malware detection and decrease false positives.
2. Saxe (2015), Berlin (2015), Hamilton (2015), Johnson (2015), and Re (2015). deep neural networks for classifying malware. Preprint for arXiv is arXiv:1508.03096.
3. Nicholas, C.,... & Nicholas, C. (2017). Raff, E., Barker, J., Sylvester, J., Brandon, B., Catanzaro, B. detection of malware by devouring a whole exe. preprint from arXiv, 1710.09435.
4. In 2016, Vinayakumar, R., Shanthi, P., and Sadasivam, V. an analysis of malware detecting methods. Computer and Information Sciences Journal of King Saud University, 28(3), 257–273.
5. The following authors (2016): Kolosnjaji, B., Zarras, A., Webster, G. D., Eckert, & Kirda. The classification of malware system call sequences using deep learning. 2016 ACM Asia Conference on Computer and Communications Security Proceedings, pp. 95–106.
6. (2013) Pascanu, R., and Bengio, Y. natural gradient revisited for deep networks. In Volume 28, Page 924, of the Proceedings of the 30th International Conference on Machine Learning.
7. The following authors (2014): Arp, D., Spreitzenbarth, M., Hübner, M., Gascon, H., Rieck, K., & Siemens, C. Drebin: In-pocket malware detection for Android that is both effective and understandable. In the 2014 Network and Distributed System Security Symposium Proceedings.
8. (2008). Rieck, K., Trinius, P., Willems, C., Holz, T., Bos, H., and Giampietro. machine learning-based study of malware behaviour. 151–163 in Journal of Computer Virology, 4(3).
9. (2006). Kolter, J. Z., and Maloof, M. A. acquiring knowledge of identifying and categorising harmful executables in the wild. 7, 2721–2744, The Journal of Machine Learning Research.