VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

PROJECT REPORT

Team Numbers:

- Varanasi Sai Charan – 21BCE5870
- Kalla Bharath Vardhan – 21BCE5846
- Surya Kiran C – 21BCE1110
- Mandava Nidhish – 21BCE5840

Course Name: Cryptography and Network Security

Course Code: BCSE302L

Supervised By: Dr. Sobitha Ahila S

Malware Detection (Trojan Horse, Spyware) Through Memory Analysis by Using Hybrid Algorithms (Extreme Gradient Boosting and Random Forest Classifier).

Abstract:

Cybersecurity faces significant challenges to block malware such as Trojan horses and Spyware in the digital era. Traditional detection methods are inadequate to detect these obfuscated malware as they pose advanced threat. We have implemented Navie Bayes, Decision tree, Extreme Gradient Boosting and Random Forest. In these 4 algorithms extreme gradient boosting and random forest showed most promising results. So, we took these two algorithms together as our proposed model, which involves an innovative approach in which we use a hybrid algorithm, combining Extreme Gradient Boosting and Random Forest classifiers. The primary goal is to analyze the target system memory to identify obfuscated malware. The proposed hybrid algorithm enhances detection accuracy and efficiency by examining patterns and behaviors within the system's volatile memory. This process includes collecting and analyzing memory snapshots from various digital environments encompassing both known and unknown malware scenarios. The hybrid algorithm undergoes training on a diverse dataset to ensure optimal performance across different variants of Trojan horses and Spyware. In this proposed approach we again have three different methods: Stacking classifier recoded accuracy 94.69%, Voting classifier recoded accuracy 95.20% and Blended model recoded accuracy 95.45%. The results of this study are expected to be highly beneficial in advancing memory-based malware detection techniques, providing a robust defense against evolving cyber threats.

Introduction:

The cybersecurity landscape is constantly evolving, with malware posing a persistent threat to digital systems worldwide. Trojan horses and spyware are particularly notorious for their ability to infiltrate systems undetected, causing significant damage to sensitive data and network infrastructure. Traditional methods of malware detection often fail to identify sophisticated threats, necessitating innovative approaches to combat their proliferation. Integration of machine learning algorithms presents a compelling opportunity to enhance the efficiency and accuracy of malware detection, especially for sophisticated threats like Trojan horses and spyware. Development of an adaptive malware detection system leveraging memory analysis and machine learning capabilities. Firstly, we used 4 algorithms Naive Bayes, decision tree, Random Forest & Extreme Gradient Boost aiming that we got high accuracy for Extreme Gradient Boost and random forest. So, Utilization of a hybrid algorithm combining Extreme gradient boosting and random forest classifiers for enhanced detection accuracy and resilience. Ability to choose different algorithms dynamically for better accuracy as needed. Acquisition and pre-processing of real-world memory dump datasets containing various malware types, with emphasis on Trojan horses and spyware. Design and implementation of the hybrid algorithm to bolster the accuracy and efficiency of malware detection. Experimentation and validation methodologies including cross-validation and performance metric analysis. Assessment of the efficacy of the approach in detecting and mitigating previously unseen malware samples and zero threats. Redefining the boundaries of traditional malware detection methodologies to contribute meaningfully to cybersecurity.

The proliferation of ransomware attacks presents a significant cybersecurity challenge for businesses worldwide, causing financial losses and disrupting essential services. Traditional detection methods struggle to keep pace with the evolving tactics of ransomware operators. In response, this study proposes a novel approach leveraging machine learning and memory analysis to enhance ransomware detection. By constructing a robust dataset comprising recent ransomware samples and benign applications, and employing advanced machine learning models such as XGBoost, the study achieves a remarkable 97.85% accuracy with a minimal 2% false positive rate. Through extensive analysis of memory traces, this research sheds light on the efficacy of memory

artifacts in detecting ransomware, providing a significant contribution to the field of cybersecurity.[1]

In the ever-evolving landscape of cybersecurity threats, malware remains a pervasive issue, posing risks to individuals and organizations worldwide. Among the various types of malwares, obfuscated malware presents a particularly challenging detection problem due to its ability to evade traditional signature-based approaches. Addressing this challenge, this paper introduces MalHyStack, a novel hybrid classification model designed for detecting obfuscated malware within network traffic. By integrating conventional machine learning algorithms with deep neural networks in a stacked ensemble learning scheme, Haystack offers superior detection capabilities. Through feature subset selection and performance evaluation using the CIC-MalMem-2022 dataset, the proposed model demonstrates a significant improvement in accuracy and time efficiency compared to existing classification models, marking a noteworthy advancement in malware detection technology.[2]

Delving into the ever-present menace of malware, this paper addresses the ongoing challenge of detecting malicious software in the rapidly evolving landscape of technology. With a focus on machine learning methodologies, the study conducts a systematic literature review to categorize approaches for malware detection. By analysing 77 research works, it aims to provide a comprehensive taxonomy that sheds light on the effectiveness of machine learning algorithms in combating malware.[3]

In response to the persistent threat posed by malware, this paper presents a novel behaviour-based detection technique. Departing from traditional signature-based methods, which rely on known patterns, the approach utilizes dynamic analysis to capture runtime features. By leveraging the Cuckoo sandbox environment, various runtime behaviours such as printable strings, API calls, and network activities are extracted and processed using advanced techniques. Through experimental validation, the paper demonstrates the high accuracy of ensemble machine learning algorithms in detecting malware.[4]

Tackling the challenges posed by obfuscated malware in IoT Android applications, this paper introduces a visualization-based detection system. With traditional static and dynamic analysis techniques proving inadequate for identifying obfuscated malware, the study proposes a novel approach using Markov images generated from executable files. By training convolutional neural network models on these images, the system achieves remarkable accuracy in detecting obfuscated malware and identifying the specific obfuscation techniques employed. Through rigorous experimentation, the paper showcases the efficacy of the proposed approach in enhancing malware detection in IoT environments.[5]

Literature Survey:

Literature Survey - 1:

Title: Ransomware detection based on machine learning using memory features,Malak Aljabri , Fahd Alhaidari ,Aminah Albuainain , Samiyah Alrashidi , Jana Alansari ,Wasmiyah Alqahtani , Jana Alshaya December 2023, Egyptian Informatics Journal, Science Direct.

Techniques Used: Xg boost, light gbm, extra tree, adaptive boosting, random Forest are the models used and have chosen the best model according to the accuracy.

Model	Accuracy	F1-Score	Precision	ROC-AUC	Feature selection Algorithm
XGBoost	97.85	97.58	97.12	97.88	Chi-2
XGBoost	96.15	96.15	95.24	96.62	SFS
LightGBM	97	96.59	97.06	96.9	Chi-2
LightGBM	96.14	95.65	95.19	96.14	SFS
Extra Tree (ET)	96.57	96.08	97.03	96.42	Chi-2
Adaptive Boosting (AdaBoost)	93.99	93.14	94.06	93.81	Chi-2
Random Forest (RF)	97	96.65	95.28	97.11	Chi-2

Challenges Identified: Ransomware often employs techniques to obfuscate its code and behavior, making it challenging to detect using traditional methods or even memory analysis. Ransomware encrypts files and network data, making it difficult to locate the malicious program through the study of encrypted files or network activity alone.

Literature Survey - 2:

Title: A hybrid stacked ensemble learning framework with feature engineering schemes for obfuscated malware analysis Kowshik Sankar Roy, Tanim Ahmed, Pritom Biswas Udas, Md. Ebtidaul Karim, Sourav Majumdar September 2023, Intelligent Systems with Applications, Science Direct.

Techniques Used: SVM, decision tree, KNN, Random forest, Xg boost, Adaboost, extra trees, DNN are the algorithms used for the malware detection and have chosen the best model.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SVM	98.96	98.92	98.98	98.96
Decision Tree	98.22	98.18	99.26	98.22
KNN	99.89	99.85	99.93	99.89
Random Forest	99.83	99.67	99.99	99.83
XgBoost	99.92	99.91	99.94	99.93
AdaBoost	98.22	99.21	99.23	98.22
Extra Trees	99.96	99.92	99.98	99.96
DNN	99.98	99.97	99.99	99.98

Challenges Identified: Machine learning in security faces challenges including acquiring quality data, vulnerability to adversarial attacks, and ensuring understandable decision-making processes. This makes it difficult for security systems to spot and stop threats effectively

Literature Survey - 3:

Title: Obfuscated Malware Detection in IoT Android Applications Using Markov Images and CNN, Dhanya K. A., Vinod P., Suleiman Y. Yerima, Abul Bashar, Anwin David, Abhiram T., Alan Antony,Ashil K. Shavanas, and Gireesh Kumar T, June 2023, IEEE Systems Journal, IEEE.

Techniques Used: The models used are SVM, Random forest, Decision forest, KNN.

Model	Accuracy	F-Measure	ROC-AUC
SVM	99.54	99.47	99.54
Random forest	99.25	99.09	99.29
Decision tree	99.92	99.91	99.92
KNN	99.66	99.66	99.82

Challenges Identified: The main challenge is to design and optimize Convolutional Neural Network (CNN) models capable of effectively detecting obfuscated malware in IoT Android applications. These challenges require a combination of advanced machine learning techniques, domain expertise in cybersecurity.

Literature Survey - 4:

Title: Machine Learning Algorithm for MalwareDetection: Taxonomy, Current Challenges,and Future Directions, NOR ZAKIAH GORMENT , (Member, IEEE), ALI SELAMAT, (Member, IEEE),LIM KOK CHENG, (Member, IEEE), AND ONDREJ KREJCAR,March 2023, IEEE Systems Journal, IEEE.

Techniques Used: The algorithms used are SVM, N gram and decision are algorithms used and chosen the best model after the implementation.

Algorithm	Accuracy	Precision	Recall	F1_Score
KNN	0.9445	0.8997	0.9971	0.9459
Naïve Bayes (NB)	0.9162	0.8470	0.9938	0.9245
SVM	0.9190	0.8522	0.9938	0.9076
Decision Tree	0.9427	0.8971	0.9942	0.9431
Random Forest	0.9451	0.8997	0.9942	0.9946
Gradient Boosting	0.9441	0.8971	0.9971	0.9344
Adaptive Boosting	0.9455	0.8997	0.9971	0.9459

Challenges Identified: The challenges in malware detection using machine learning include obfuscation techniques employed by malware creators, classification complexity in distinguishing between different malware families, the dynamic nature of malware behavior, Text classification studies encounter the sparse matrix problem, which complicates the classification process, particularly in distinguishing between malware and benign files.

Literature Survey - 5:

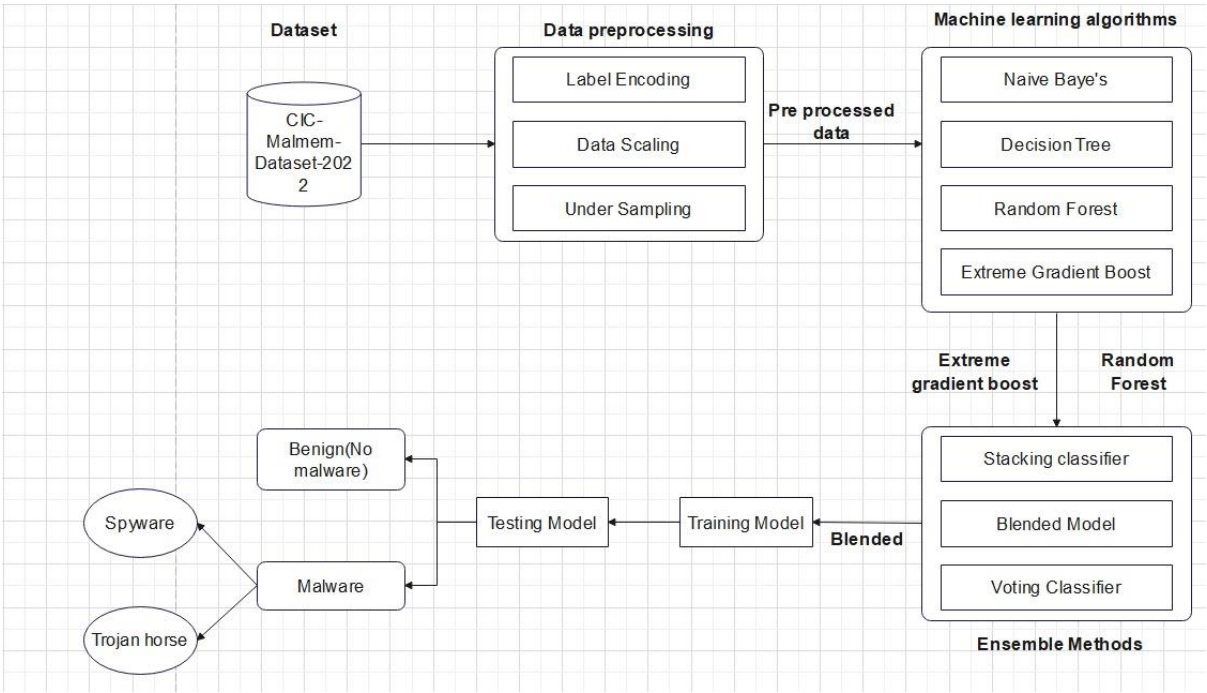
Title: Detection of malicious software by analyzing the behavioral artifacts usingmachine learning algorithmsJagsir Singh,Jaswinder Singh January 2020,Information and Software Technology, Science Direct.

Techniques Used: Models used are KNN, Navie Baya's, SVM, Decision tree, Random forest, Gradient boosting, Adaptive boosting.

Model	ROC	F1-Score	Precision	Recall	Accuracy
SVMS	99.93	98.24	98.94	98.46	98.62
Decision tree	99.64	97.45	98.85	96.08	96.49
N-gram	99.81	97.68	98.81	97.04	97.43

Challenges Identified: The challenges in computer security against malware include coping with the exponential growth and complexity of malware, addressing targeted and stealthy attack techniques, managing network-related threats, and handling the complexities of dynamic analysis for malware detection.

Architectural Diagram:



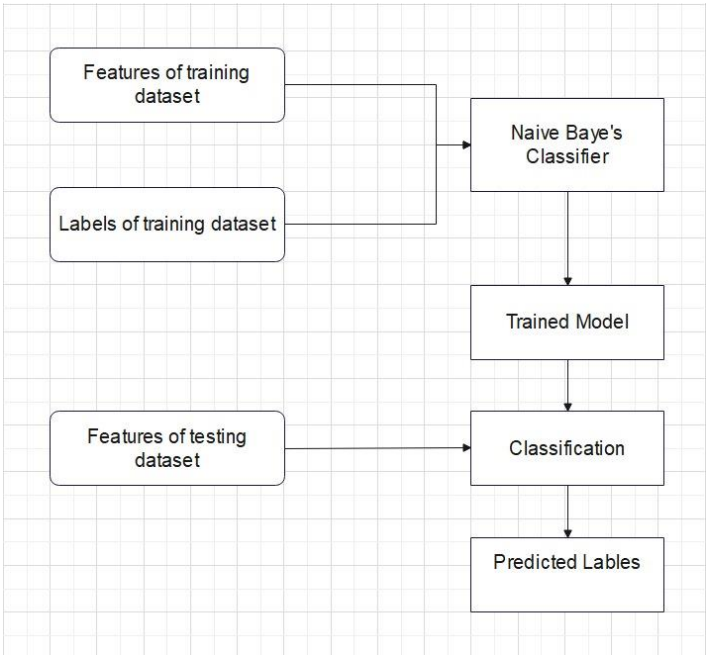
Module-1: Data Preprocessing

In data preprocessing phase, we cleaned real-world memory dump dataset, ensuring the integrity and quality of the data. This involved removing the irrelevant or duplicate entries, handling missing values, and standardizing the data. Additionally, we conducted exploratory data analysis to gain insights into the underlying distribution and characteristics of the dataset, enabling us to make decisions regarding feature selection and model training. Furthermore, in the data processing stage, we implemented one-hot encoding to transform categorical variables such as malware types (Trojan horse, Spyware) into numerical representations, facilitating their incorporation into machine learning algorithms. We also employed feature scaling techniques to normalize the data, ensuring that all features contribute proportionally to the model's learning process. This meticulous data preprocessing and processing steps are essential for optimizing the performance and accuracy of our hybrid algorithm in detecting and clearing.

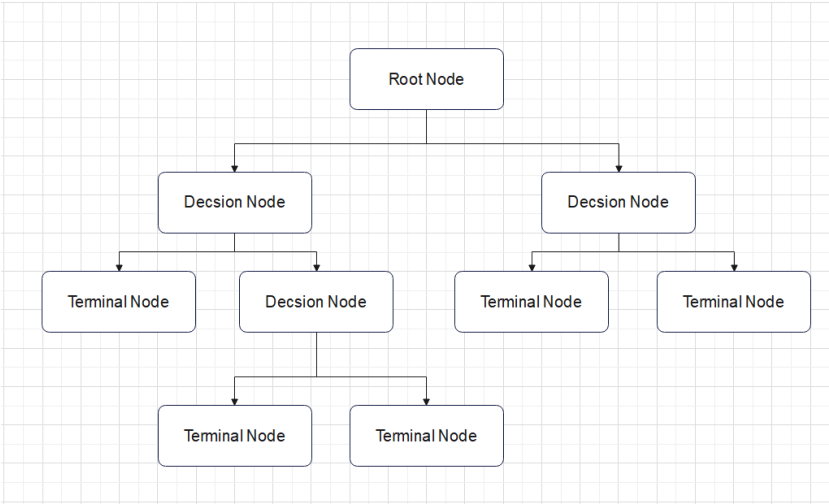
Module-2: Machine learning algorithms

In this module we take the preprocessed data and run it through four different machine learning algorithms,

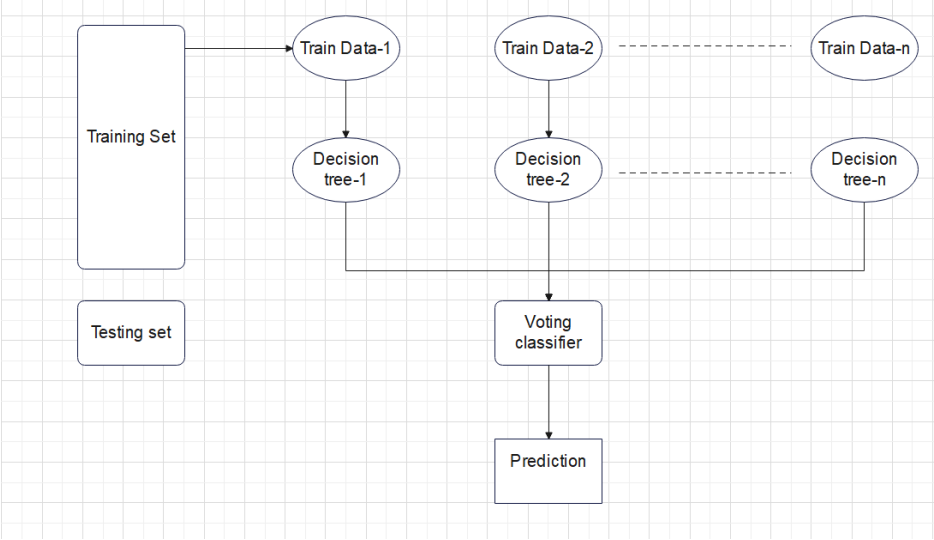
1) Naive Bayes: Naive Bayes offers a straightforward yet effective approach to malware detection through memory analysis. By assuming independence between features, it efficiently calculates the probability of a sample belonging to a particular class based on its feature distribution. Its simplicity and computational efficiency make it a viable option for real-time detection tasks, particularly in scenarios where resource constraints are a concern.



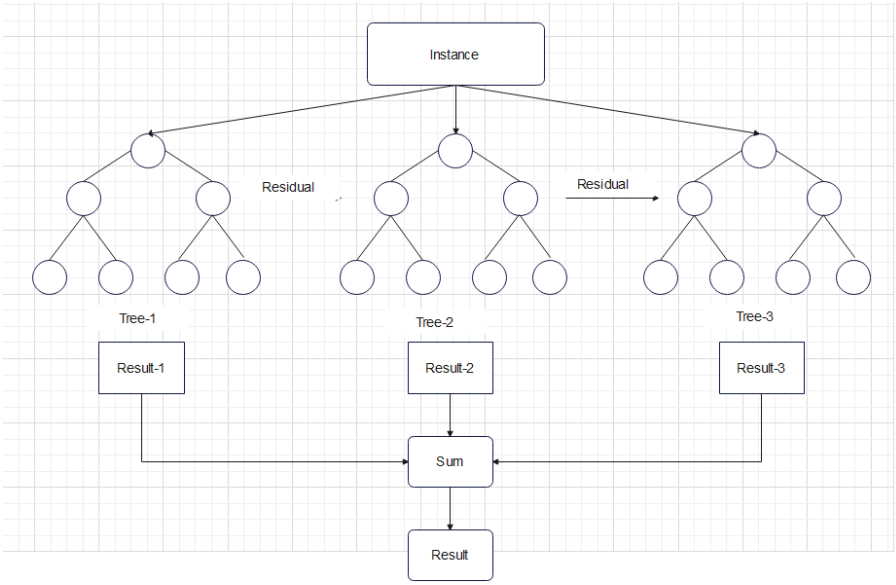
2) Decision Trees: Decision trees serve as a versatile tool in the arsenal of malware detection methodologies. Through recursive partitioning of feature space, they delineate decision boundaries that separate benign from malicious behavior. Their interpretability aids in understanding the underlying logic of classification, facilitating insights into malware characteristics and aiding in feature selection. However, they may suffer from overfitting and lack robustness when dealing with complex, high-dimensional data.



3) Random Forest: Random Forest emerges as a powerful ensemble technique for malware detection through memory analysis. By aggregating predictions from a multitude of decision trees, it mitigates overfitting and enhances generalization capabilities. Its inherent robustness to noise and outliers makes it well-suited for handling the intricate and dynamic nature of malware behavior. Furthermore, its ability to handle large datasets and parallelize computations expedites the detection process.



4) Extreme Gradient Boosting: Extreme Gradient Boosting (XGBoost) is a powerful machine learning algorithm renowned for its efficiency and accuracy. It employs a boosting technique, combining weak learners to form a strong model. XGBoost minimizes loss functions using gradient descent optimization, enhancing predictive performance. Its parallelization capabilities make it highly scalable, fitting well with large datasets. With its regularization techniques, XGBoost effectively prevents overfitting, ensuring robust model generalization.



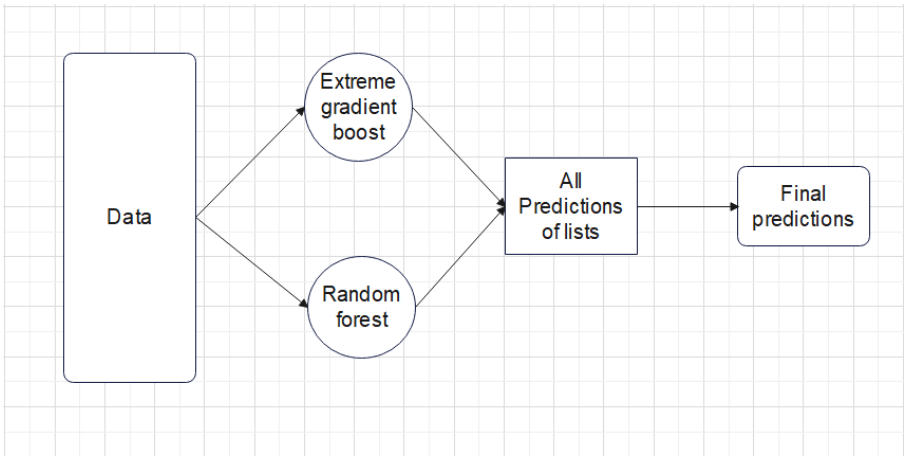
Module-3: Ensemble methods

In Module-2 we found that Extreme Gradient Boosting and Random Forest shows more promising results, so we combine them using ensemble methods in three different categories such as,

1) Voting Classifier: Voting Classifier harness the collective wisdom of multiple base models to bolster malware detection efficacy. By combining predictions from diverse classifiers, it mitigates individual model biases and variance, resulting in more reliable classifications. This democratic approach ensures that each model's strengths contribute to the final decision, enhancing overall accuracy and robustness in detecting malicious software.

2) Stacking Classifier: Stacking Classifier represents a sophisticated ensemble technique that orchestrates a hierarchical fusion of diverse base learners for malware detection. By training a meta-classifier on the predictions of multiple base models, it learns to capture higher-order relationships among their outputs, thus refining the decision-making process. This meta-learning paradigm enables the exploitation of complementary strengths across different models, culminating in enhanced detection performance and adaptability to evolving malware threats.

3) Blended Model: The Blended Model embodies a strategic amalgamation of diverse machine learning techniques tailored for malware detection via memory analysis. By blending predictions from multiple base models with varying strengths and characteristics, it harnesses the collective intelligence of each component to yield a more robust and accurate detection framework. Through careful weighting and combination strategies, it leverages the complementary nature of different algorithms, synergistically enhancing detection efficacy while mitigating individual model biases.



Results:

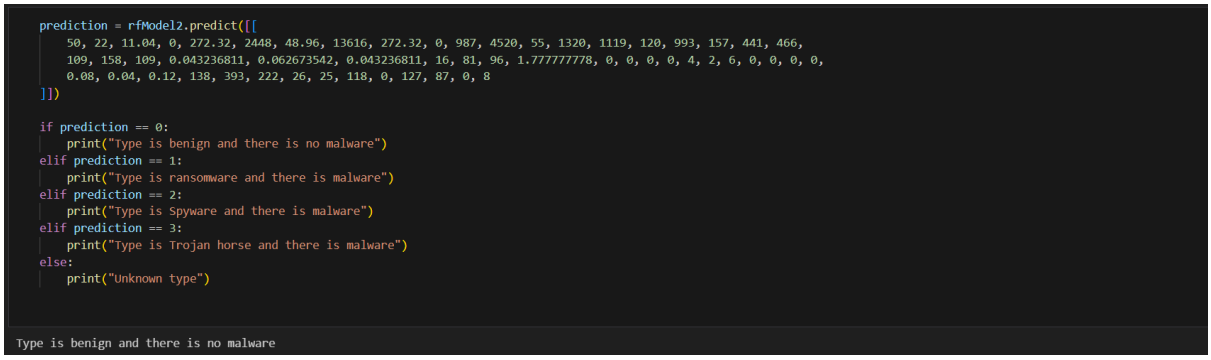
For Individual Models:

S. No	Model	Accuracy	Precision	Recall	F1-Score
1.	Gaussian Naïve Baye's	81.44	83.72	81.44	78.99
2.	Random Forest	91.95	92.40	91.95	91.90
3.	Decision Tree	90.88	91.00	90.88	90.88
4.	Extreme gradient boost	93.61	93.64	93.61	93.61

Ensemble Models:

S. No	Ensemble method	Accuracy
1.	Voting classifier	95.20%
2.	Stacking classifier	95.36%
3.	Blend Model	94.45%

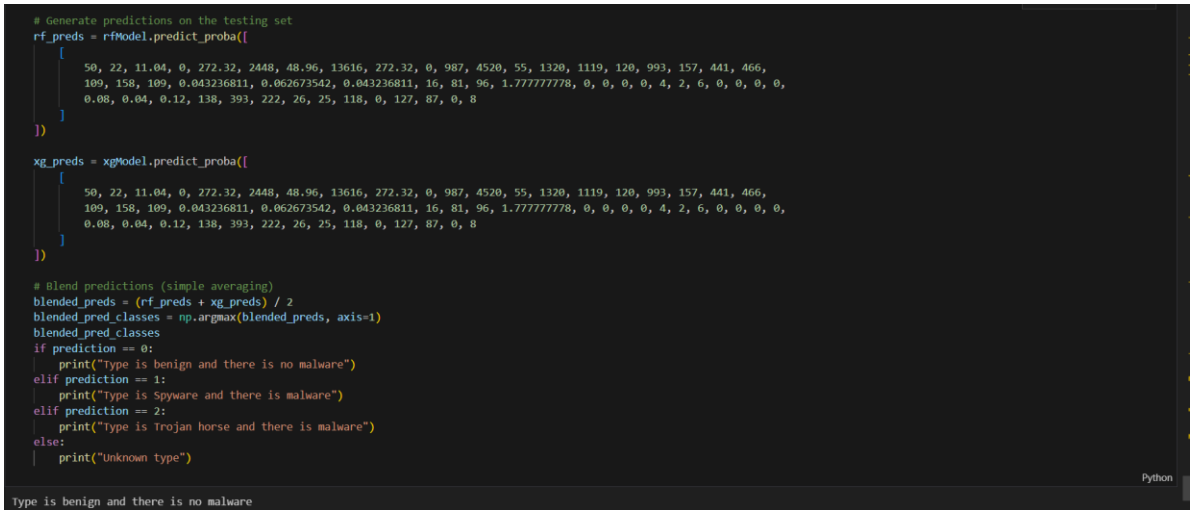
Result Snapshots:



```
prediction = rfModel2.predict([[
    50, 22, 11.04, 0, 272.32, 2448, 48.96, 13616, 272.32, 0, 987, 4520, 55, 1320, 1119, 120, 993, 157, 441, 466,
    109, 158, 109, 0.043236811, 0.062673542, 0.043236811, 16, 81, 96, 1.777777778, 0, 0, 0, 0, 4, 2, 6, 0, 0, 0, 0,
    0.08, 0.04, 0.12, 138, 393, 222, 26, 25, 118, 0, 127, 87, 0, 8
]])

if prediction == 0:
    print("Type is benign and there is no malware")
elif prediction == 1:
    print("Type is ransomware and there is malware")
elif prediction == 2:
    print("Type is Spyware and there is malware")
elif prediction == 3:
    print("Type is Trojan horse and there is malware")
else:
    print("Unknown type")
```

```
Type is benign and there is no malware
```



```python
# Generate predictions on the testing set
rf_preds = rfModel.predict_proba([
    [
        50, 22, 11.04, 0, 272.32, 2448, 48.96, 13616, 272.32, 0, 987, 4520, 55, 1320, 1119, 120, 993, 157, 441, 466,
        109, 158, 109, 0.043236811, 0.062673542, 0.043236811, 16, 81, 96, 1.777777778, 0, 0, 0, 0, 4, 2, 6, 0, 0, 0, 0,
        0.08, 0.04, 0.12, 138, 393, 222, 26, 25, 118, 0, 127, 87, 0, 8
    ]
])

xg_preds = xgModel.predict_proba([
    [
        50, 22, 11.04, 0, 272.32, 2448, 48.96, 13616, 272.32, 0, 987, 4520, 55, 1320, 1119, 120, 993, 157, 441, 466,
        109, 158, 109, 0.043236811, 0.062673542, 0.043236811, 16, 81, 96, 1.777777778, 0, 0, 0, 0, 4, 2, 6, 0, 0, 0, 0,
        0.08, 0.04, 0.12, 138, 393, 222, 26, 25, 118, 0, 127, 87, 0, 8
    ]
])

# Blend predictions (simple averaging)
blended_preds = (rf_preds + xg_preds) / 2
blended_pred_classes = np.argmax(blended_preds, axis=1)
blended_pred_classes
if prediction == 0:
    print("Type is benign and there is no malware")
elif prediction == 1:
    print("Type is Spyware and there is malware")
elif prediction == 2:
    print("Type is Trojan horse and there is malware")
else:
    print("Unknown type")
```

Python

```
Type is benign and there is no malware
```

Future work:

In future work, we envision several avenues for enhancing the proactive detection of obfuscated malware on Android IoT platforms. Lastly, we aim to further enhance our models to identify or generate obfuscation signatures from image representations, thereby improving the accuracy and effectiveness of obfuscated malware detection systems.

Conclusion:

In our analysis, we employed four distinct algorithms: Naive Bayes, Decision Tree, Random Forest, and Extreme Gradient Boost, with the objective of achieving high accuracy. Notably, Extreme Gradient Boost emerged as the top performer with an accuracy of 93.61%, closely followed by Random Forest at 91.95%. To further enhance accuracy and robustness, we proposed a hybrid approach amalgamating Gradient Boosting and Random Forest classifiers, leveraging their individual strengths. Subsequently, employing ensembling techniques such as Voting Classifier, Stacking Classifier, and a Blended Model, we achieved remarkable accuracy improvements. Specifically, the Voting Classifier yielded 95.20% accuracy, the Stacking Classifier improved it to 95.36%, and our innovative Blended Model culminated in an impressive accuracy of 95.45%. This comprehensive approach not only demonstrates the efficacy of ensemble methods but also underscores the significance of combining high-performing algorithms for superior detection accuracy and resilience in our model.

References:

1. Ransomware detection based on machine learning using memory features,Malak Aljabri , Fahd Alhaidari ,Aminah Albuainain , Samiyah Alrashidi , Jana Alansari ,Wasmiyah Alqahtani , Jana Alshaya December 2023, Egyptian Informatics Journal, Science Direct.
2. A hybrid stacked ensemble learning framework with feature engineering schemes for obfuscated malware analysis Kowshik Sankar Roy, Tanim Ahmed, Pritom Biswas Udas , Md. Ebtidaul Karim ,Sourav Majumdar September 2023,Intelligent Systems with Applications, Science Direct.
3. Obfuscated Malware Detection in IoT Android Applications Using Markov Images and CNN, Dhanya K. A., Vinod P., Suleiman Y. Yerima, Abul Bashar, Anwin David, Abhiram T., Alan Antony,Ashil K. Shavanas, and Gireesh Kumar T, June 2023, IEEE Systems Journal, IEEE.
4. Machine Learning Algorithm for MalwareDetection: Taxonomy, Current Challenges,and Future Directions, NOR ZAKIAH GORMENT , (Member, IEEE), ALI SELAMAT, (Member, IEEE),LIM KOK CHENG, (Member, IEEE), AND ONDREJ KREJCAR,March 2023, IEEE Systems Journal, IEEE.
5. Detection of malicious software by analyzing the behavioral artifacts usingmachine learning algorithmsJagsir Singh,Jaswinder Singh January 2020,Information and Software Technology, Science Direct.