e-ISSN:2582-7219

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

Volume 7, Issue 7, July 2024

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.521

☎ 6381 907 438 ☎ 6381 907 438 ✉ ijmrset@gmail.com @ www.ijmrset.com

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707014 |

Malware Detection using Machine Learning

Dr.Shruthi SK, V.Durga, G.Manikanta, M.Sravanthi

Assistant Professor, Department of CSE, Methodist College of Engineering and Technology, Hyderabad, Telangana, India

UG Student, Department of CSE, Methodist College of Engineering and Technology, Hyderabad, Telangana, India

ABSTRACT: Malicious software, or malware, presents a persistent threat to computer systems and networks, requiring effective detection mechanisms. Machine learning (ML) has emerged as a promising approach for malware detection due to its ability to analyze large datasets and identify complex patterns indicative of malicious behavior. This paper provides a comprehensive review of state-of-the-art ML techniques applied to malware detection and proposes a novel framework for enhancing detection accuracy and efficiency. The proposed framework leverages a combination of supervised, unsupervised, and semi-supervised ML algorithms to analyze various features extracted from malware samples. These features include static features such as file size, file type, and API calls, as well as dynamic features like system calls, network traffic, and behavioral patterns. By integrating multiple ML models and feature sets, the framework aims to achieve robust detection across different types of malware, including viruses, worms, Trojans, and ransomware.

KEYWORDS: Machine Learning, Malware, Accuracy, Efficiency, Supervised, Unsupervised, Ransomware

I. INTRODUCTION

Malware, short for malicious software, represents one of the most significant threats in the digital landscape. With cyberattacks becoming increasingly sophisticated and widespread, the need for robust malware detection and prevention mechanisms has never been greater. Machine learning, a branch of artificial intelligence, has emerged as a powerful tool in this domain, offering advanced capabilities to identify and combat malware effectively. At its core, malware encompasses a range of malicious software designed to disrupt, damage, or gain unauthorized access to computer systems and data. This includes viruses, worms, trojans, spyware, ransomware, and more. Traditional approaches to malware detection often relied on signature-based methods, where known malware signatures were compared against files to identify threats. While effective against known threats, this approach struggled with new and evolving malware variants that could easily evade detection through signature modifications or polymorphic techniques.

1.1 OBJECTIVE

Enhance Detection Accuracy: Machine learning models can learn from vast amounts of data to accurately identify known and unknown malware variants, including zero-day threats and polymorphic malware.

Reduce False Positives: Advanced analytics and pattern recognition capabilities in ML algorithms can help reduce false positive alerts, minimizing the impact of alert fatigue on security teams.

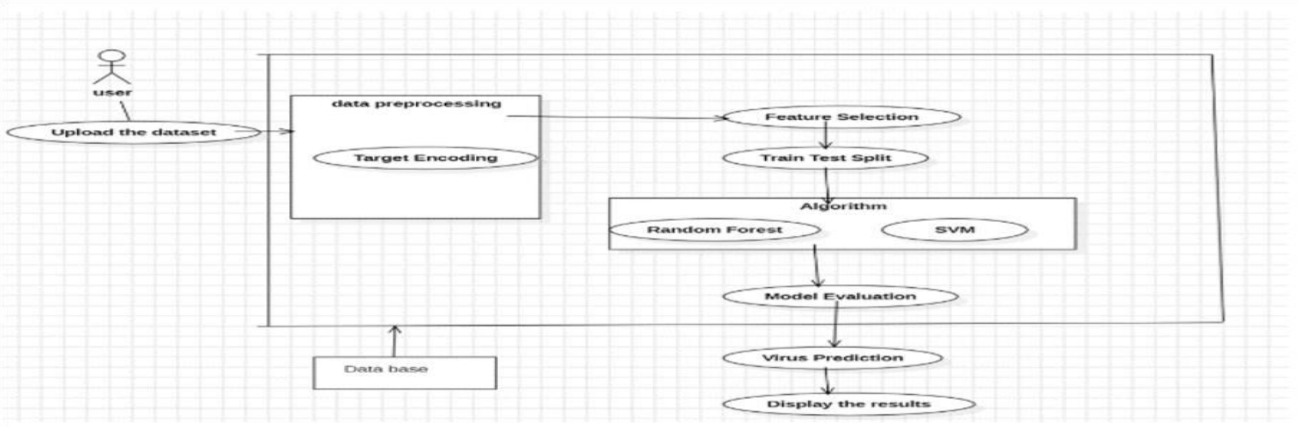
Enable Real-Time Detection: Machine learning can facilitate real-time detection and response to cyber threats, providing proactive defense mechanisms against malicious activities.

Enhance Scalability: ML-based solutions can scale effectively to handle large volumes of data, improving the scalability and performance of malware detection systems.

1.2 PROBLEM STATEMENT

Traditional malware detection methods, such as signature-based and heuristic approaches, are increasingly ineffective against sophisticated and constantly evolving cyber threats. These methods often result in high false positive rates, miss zero-day attacks, and struggle to keep pace with the rapid mutation of malware variants. Additionally, the growing volume and complexity of data further strain existing detection systems, leading to scalability challenges. To address these issues, there is a critical need to leverage advanced machine learning techniques to enhance malware detection accuracy, reduce false alarms, enable real-time threat identification, and improve scalability. The goal is to develop robust and adaptive malware detection systems capable of effectively combating modern cyber threats while minimizing operational burdens and resource constraint.

II. SYSTEM METHODOLOGY



2.1 System Architecture

In order to achieve the objective malware detection using machine learning it is divided into six steps. Data Collection, Data Pre-Processing, Data Selection, Model Training, Evaluating model, Virus Prediction. Finally the virus is detected and virus will be identified and gives precautions to be taken

III. PROPOSED SYSTEM

In this project, we proposed a system for malware detection utilizing machine learning techniques, specifically Random Forest and AdaBoost classifiers. By training these models on diverse malware datasets, we aim to enhance detection and prediction accuracy. Our approach involves extracting pertinent features from malware samples to represent malicious behavior effectively. Through rigorous experimentation and evaluation, we seek to demonstrate the efficacy of our proposed system in accurately identifying malware instances. This research contributes to advancing the field of cybersecurity by offering a robust solution that leverages the power of machine learning for proactive malware detection, thereby bolstering defense mechanisms against evolving cyber threats.

3.1 RANDOM FOREST

Random Forest is a powerful machine learning algorithm that shows promise in the realm of malware detection. By leveraging an ensemble of decision trees, Random Forest can effectively analyze diverse features extracted from malware samples, allowing for robust detection capabilities. Its ability to handle highdimensional data and mitigate overfitting makes it well-suited for complex malware detection tasks. In our research, we employ Random Forest as a key component of our proposed system, aiming to enhance the accuracy and efficiency of malware detection. Through extensive experimentation and evaluation, we seek to demonstrate the effectiveness of Random Forest in accurately identifying and classifying malware instances, thereby contributing to the advancement of cybersecurity defenses.

3.2 SUPPORT VECTOR MACHINE

Support Vector Machines (SVM) offer a sophisticated approach to malware detection in the context of machine learning. SVMs excel in classifying data by finding the optimal hyperplane that maximally separates different classes, making them particularly effective for binary classification tasks like malware detection. By representing malware samples as feature vectors, SVMs can discern complex patterns and distinguish between malicious and benign software with high accuracy. In our research, SVM is a core component of our proposed system, utilized alongside other machine learning techniques for comprehensive malware detection. Through rigorous experimentation and evaluation, we aim to demonstrate the efficacy of SVM in accurately identifying and classifying malware instances, contributing to the development of more robust cybersecurity solutions.

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707014 |

3.3 ADABOOST

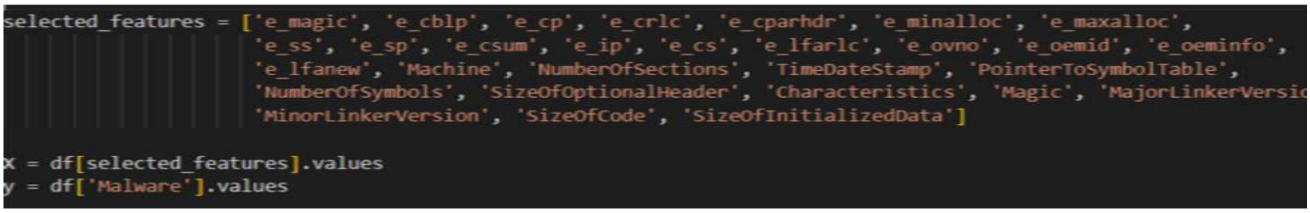
AdaBoost, an ensemble learning technique, demonstrates significant potential in the realm of malware detection using machine learning. By iteratively training a sequence of weak classifiers and giving more weight to misclassified instances in subsequent iterations, AdaBoost can effectively learn from previous mistakes and improve classification accuracy. In the context of malware detection, AdaBoost can harness the collective strength of multiple weak classifiers to accurately identify malicious software while minimizing false positives. In our research, AdaBoost serves as a fundamental component of our proposed system, working in tandem with other machine learning algorithms to enhance detection performance. Through rigorous experimentation and evaluation, we aim to showcase the effectiveness of AdaBoost in accurately detecting and classifying malware, thus contributing to the advancement of cybersecurity defenses.

IV. IMPLEMENTATION

4.1 DATA COLLECTION

Data collection is the initial step in the machine learning workflow, where the necessary data is gathered from various sources to build a predictive model. In this script, data collection is accomplished by reading a CSV file named "small_dataset2.csv" into a Pandas DataFrame using the `pd.read_csv` function.

4.2 PRE-PROCESSING

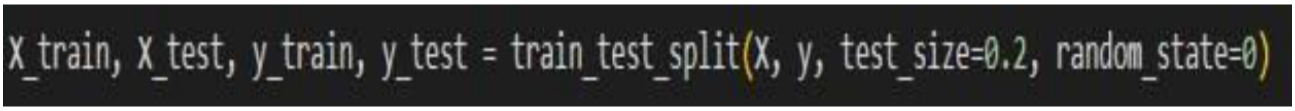


```
selected_features = ['e_magic', 'e_cblp', 'e_cp', 'e_crlc', 'e_cparhdr', 'e_minalloc', 'e_maxalloc',
                     'e_ss', 'e_sp', 'e_csum', 'e_ip', 'e_cs', 'e_lfarlc', 'e_ovno', 'e_oemid', 'e_oeminfo',
                     'e_lfanew', 'Machine', 'NumberOfSections', 'TimeDateStamp', 'PointerToSymbolTable',
                     'NumberOfSymbols', 'SizeOfOptionalHeader', 'Characteristics', 'Magic', 'MajorLinkerVersi
                     'MinorLinkerVersion', 'SizeOfCode', 'SizeOfInitializedData']

X = df[selected_features].values
y = df['Malware'].values
```

Raw text data might contain unwanted or unimportant text due to which our results might not give efficient accuracy, and might make it hard to understand and analyse. So, proper pre- processing must be done on raw data.

4.3 DATA SPLITTING

One of the first decisions to make when starting a modelling project is how to utilize the existing data. One common technique is to split the data into two groups typically referred to as the training and testing sets.

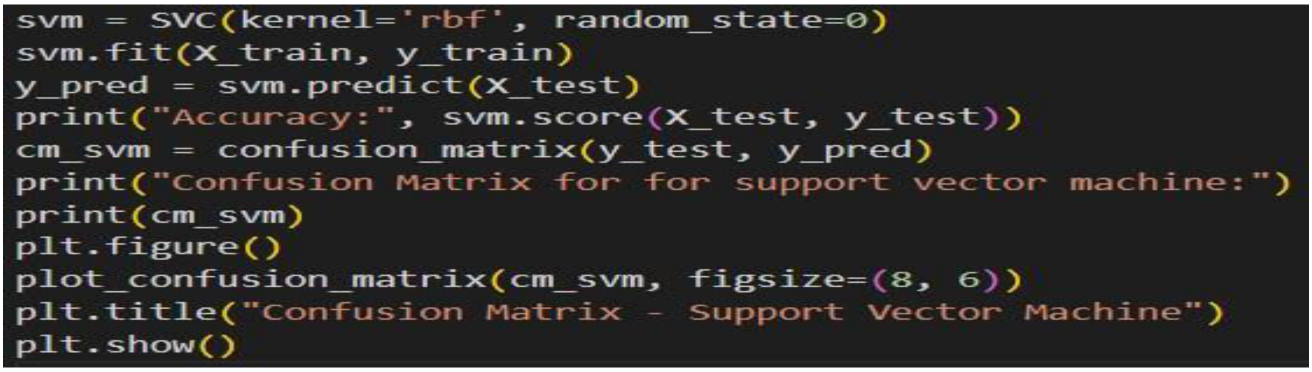


```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

4.4 MODEL TRAINING

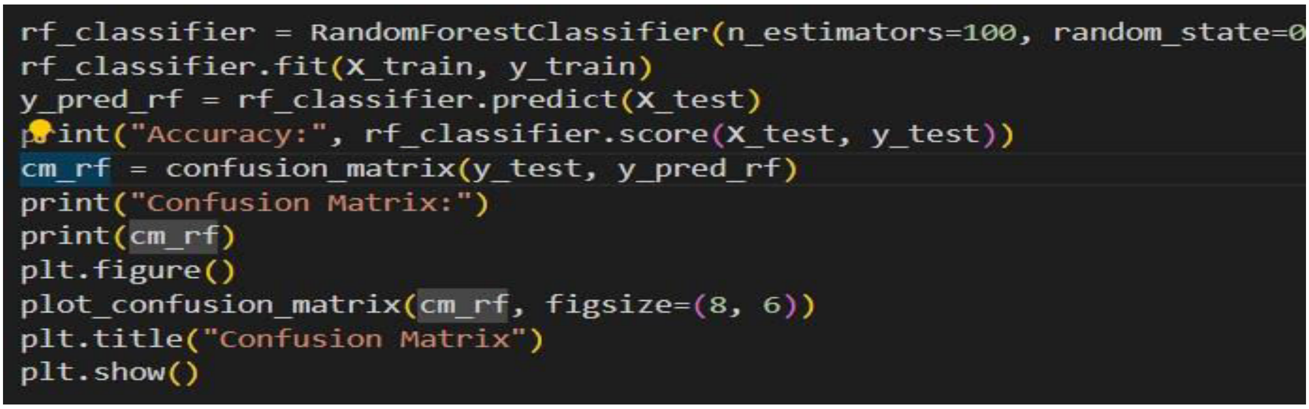
SUPPORT VECTOR MACHINE

The SVM classifier's ability to delineate between malicious and benign instances makes it a robust choice for detecting malware based on various features extracted from executable files or system behavior.



```
svm = SVC(kernel='rbf', random_state=0)
svm.fit(X_train, y_train)
y_pred = svm.predict(X_test)
print("Accuracy:", svm.score(X_test, y_test))
cm_svm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix for for support vector machine:")
print(cm_svm)
plt.figure()
plot_confusion_matrix(cm_svm, figsize=(8, 6))
plt.title("Confusion Matrix - Support Vector Machine")
plt.show()
```

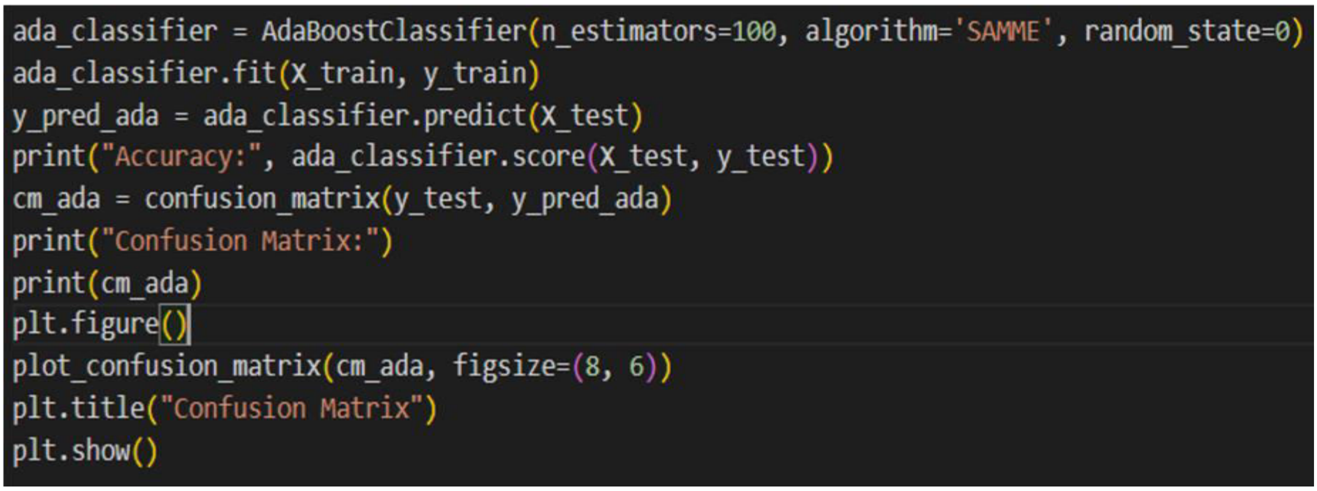
RANDOM FOREST MODEL



```
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=0
rf_classifier.fit(X_train, y_train)
y_pred_rf = rf_classifier.predict(X_test)
print("Accuracy:", rf_classifier.score(X_test, y_test))
cm_rf = confusion_matrix(y_test, y_pred_rf)
print("Confusion Matrix:")
print(cm_rf)
plt.figure()
plot_confusion_matrix(cm_rf, figsize=(8, 6))
plt.title("Confusion Matrix")
plt.show()
```

Random Forest classifier is another effective choice for malware detection using machine learning. In this approach, you would begin by loading a dataset containing features relevant to malware detection. After splitting the data into training and testing sets, you can train a Random Forest classifier using the training data.

ADABOOST MODEL



```
ada_classifier = AdaBoostClassifier(n_estimators=100, algorithm='SAMME', random_state=0)
ada_classifier.fit(X_train, y_train)
y_pred_ada = ada_classifier.predict(X_test)
print("Accuracy:", ada_classifier.score(X_test, y_test))
cm_ada = confusion_matrix(y_test, y_pred_ada)
print("Confusion Matrix:")
print(cm_ada)
plt.figure()
plot_confusion_matrix(cm_ada, figsize=(8, 6))
plt.title("Confusion Matrix")
plt.show()
```

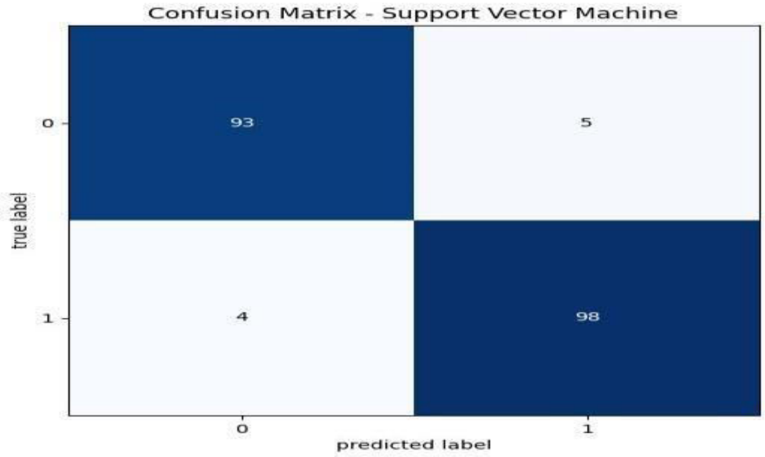
ADABoost or Adaptive Boosting is one of the ensemble boosting classifier proposed by Yoav Freund and Robert Schapire in 1996.It combines multiple weak classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method.

4.5 EVALATION METRICS

Our goal is to create and select a model which gives high accuracy on out-of-sample data. It's very crucial to use multiple evaluation metrics to evaluate your model because a model may perform well using one measurement from one evaluation metric while may perform poorly using another measurement from another evaluation metric

SUPPORT VECTOR MACHINE

Confusion Matrix of SVM



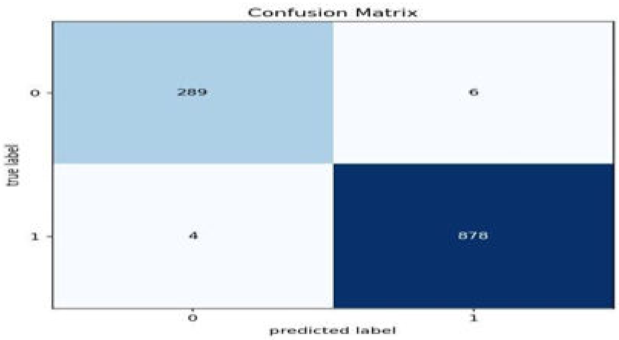
Classification Report
Support Vector Machine:

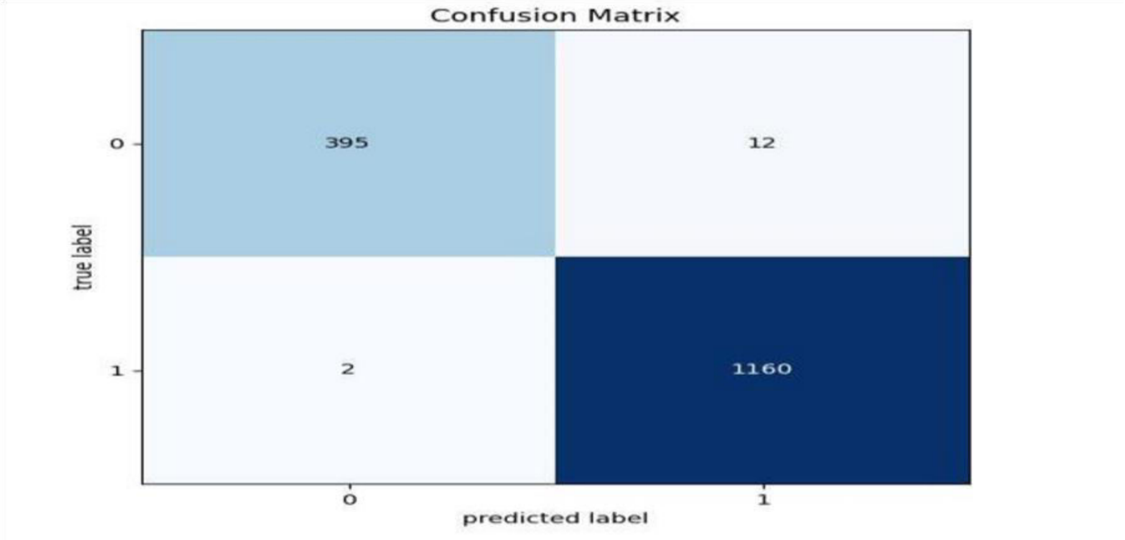
```
Classification Report:    Precision    recall  f1-score   support
0                         0.96      0.95      0.95       98
1                         0.95      0.96      0.96      102
accuracy                                               0.95       200
macro avg                              0.96      0.95      0.95      200
weighted avg                           0.96      0.95      0.95     200
```

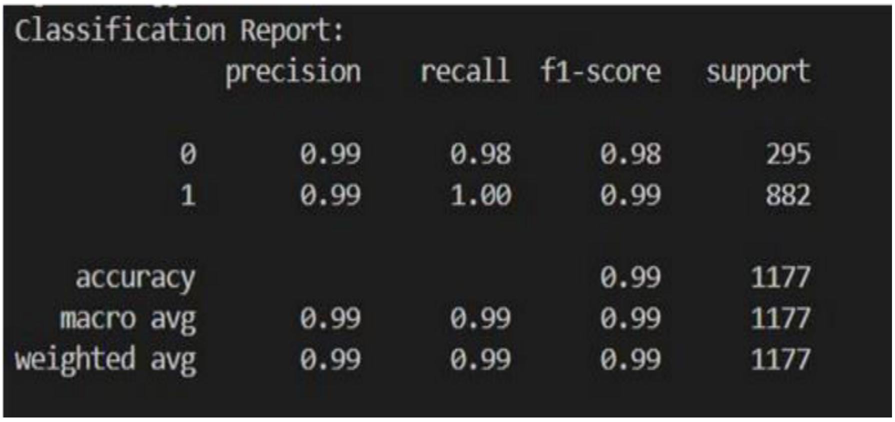
RANDOM FOREST MODEL
Confusion Matrix of Random Forest

```
Classification Report:    precision    recall   f1-score   support
0                               0.99      0.97      0.98       407
             1                       0.99      1.00      0.99      1162
accuracy                                          0.99      1569
 macro avg                        0.99      0.98      0.99      1569
weighted avg                      0.99      0.99      0.99      1569
```

ADABOOST MODEL



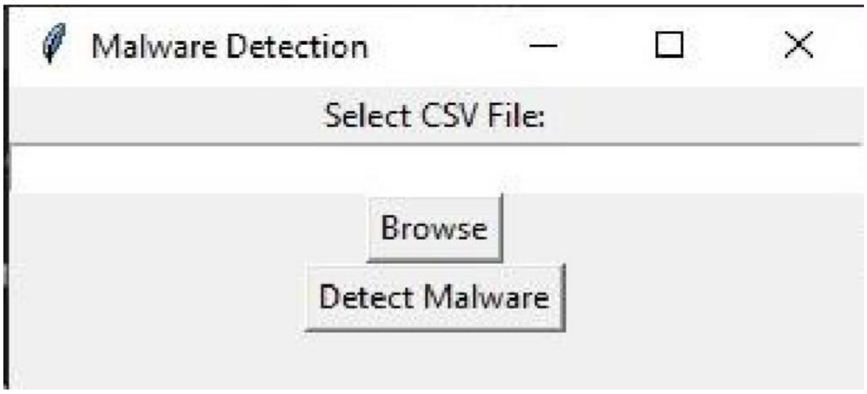


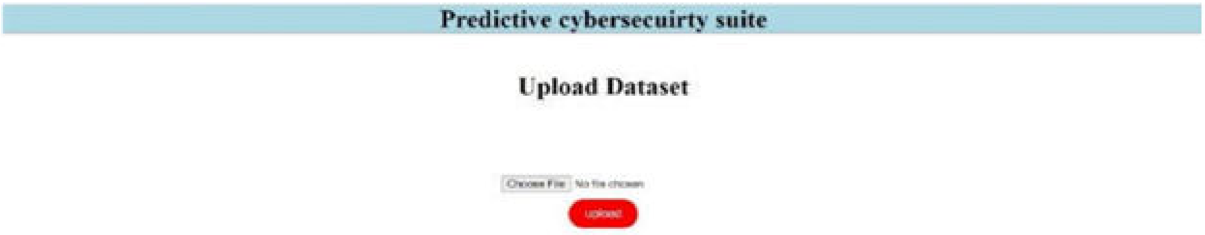


```
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.98      0.98       295
           1       0.99      1.00      0.99       882

    accuracy                           0.99      1177
   macro avg       0.99      0.99      0.99      1177
weighted avg       0.99      0.99      0.99      1177
```

V. RESULTS

As mentioned the user uploads the datasets from the flask web page either he can upload the dataset or browse the dataset then dataset is scanned processed and identify the benign values and virus values then the output will be displayed on left and right side of the window using Tkinter tool and these are the outputs.

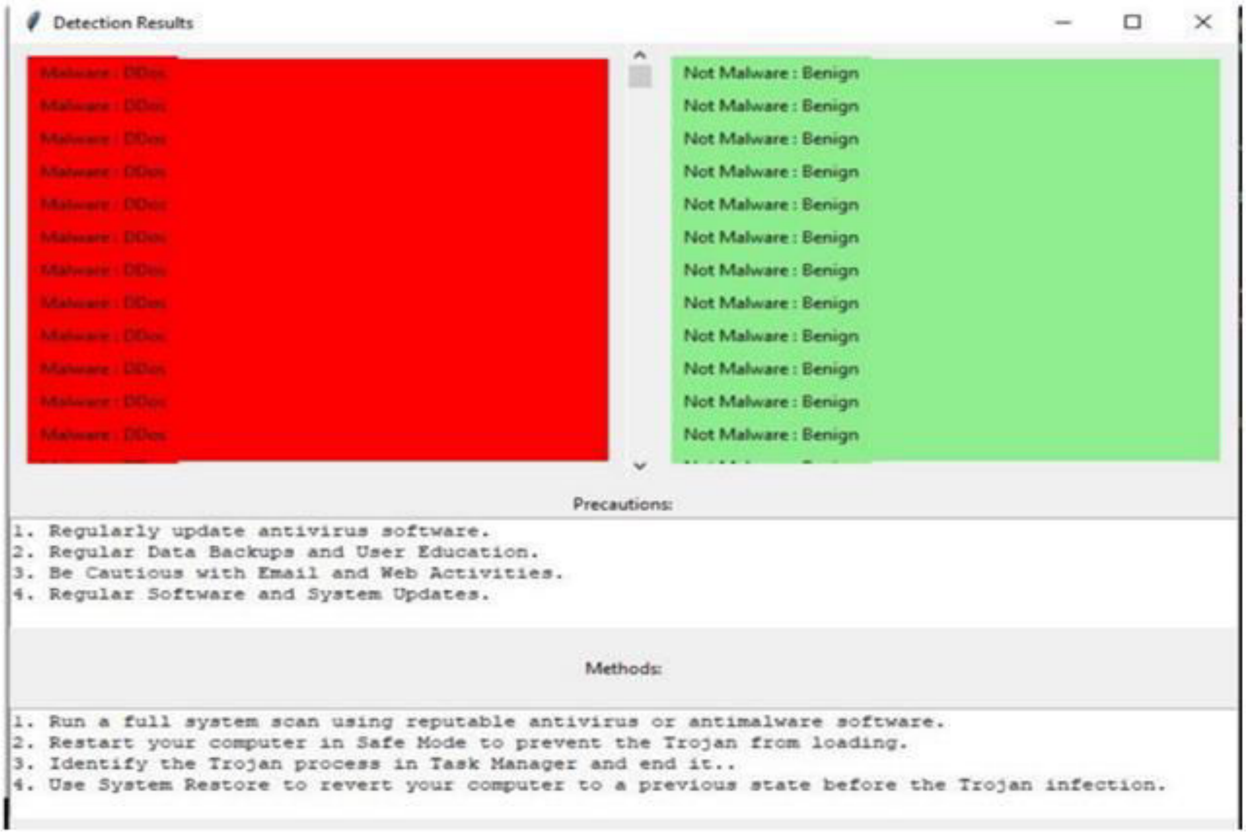




Predictive cybersecuirty suite

Upload Dataset

Input page



Detection Results

```
Malware : DDos          Not Malware : Benign
Malware : DDos          Not Malware : Benign
Malware : DDos          Not Malware : Benign
Malware : DDos          Not Malware : Benign
Malware : DDos          Not Malware : Benign
Malware : DDos          Not Malware : Benign
Malware : DDos          Not Malware : Benign
Malware : DDos          Not Malware : Benign
Malware : DDos          Not Malware : Benign
Malware : DDos          Not Malware : Benign
Malware : DDos          Not Malware : Benign
Malware : DDos          Not Malware : Benign
```

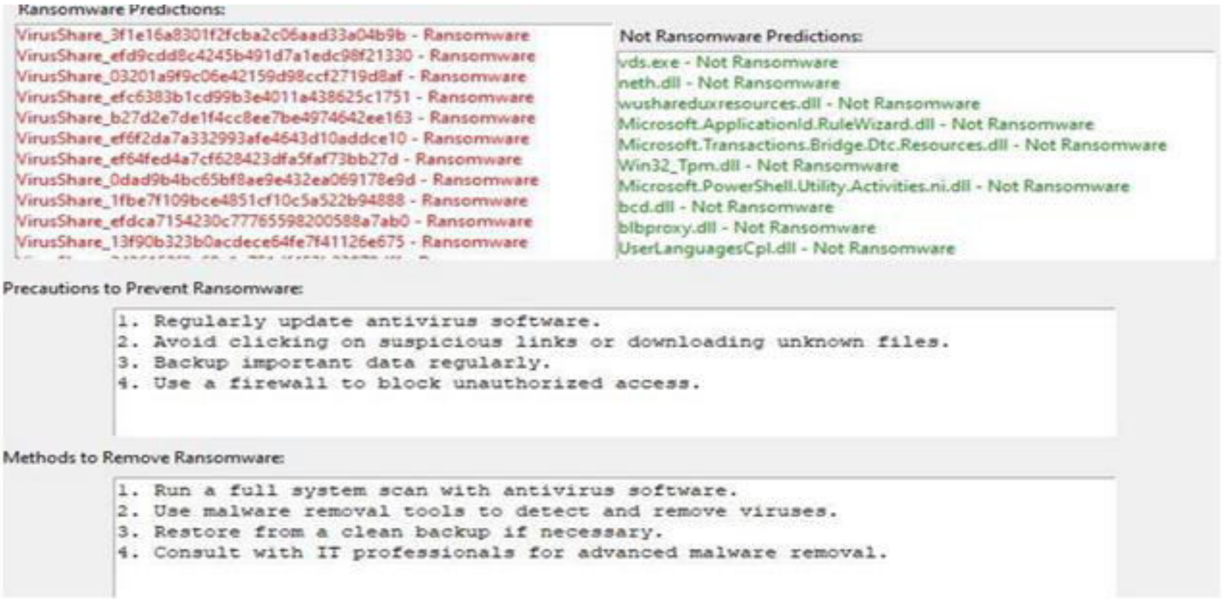
Precautions:

```
1. Regularly update antivirus software.
2. Regular Data Backups and User Education.
3. Be Cautious with Email and Web Activities.
4. Regular Software and System Updates.
```

Methods:

```
1. Run a full system scan using reputable antivirus or antimalware software.
2. Restart your computer in Safe Mode to prevent the Trojan from loading.
3. Identify the Trojan process in Task Manager and end it..
4. Use System Restore to revert your computer to a previous state before the Trojan infection.
```

Output for DDOS



Ransomware Predictions:

```
VirusShare_3f1e16a8301f2fcba2c06aad33a04b9b - Ransomware
VirusShare_efd9cdd8c4245b491d7a1edc98f21330 - Ransomware
VirusShare_03201a9f9c06e42159d98ccf2719d8af - Ransomware
VirusShare_efc6383b1cd99b3e4011a438625c1751 - Ransomware
VirusShare_b27d2e7de1f4cc8ee7be4974642ee163 - Ransomware
VirusShare_ef6f2da7a332993afe4643d10addce10 - Ransomware
VirusShare_ef64fed4a7cf628423dfa5faf73bb27d - Ransomware
VirusShare_0dad9b4bc65bf8ae9e432ea069178e9d - Ransomware
VirusShare_1fbe7f109bce4851cf10c5a522b94888 - Ransomware
VirusShare_efdca7154230c77765598200588a7ab0 - Ransomware
VirusShare_13f90b323b0acdece64fe7f41126e675 - Ransomware
```

Not Ransomware Predictions:

```
vds.exe - Not Ransomware
neth.dll - Not Ransomware
wushareduxresources.dll - Not Ransomware
Microsoft.ApplicationId.RuleWizard.dll - Not Ransomware
Microsoft.Transactions.Bridge.Dtc.Resources.dll - Not Ransomware
Win32_Tpm.dll - Not Ransomware
Microsoft.PowerShell.Utility.Activities.ni.dll - Not Ransomware
bcd.dll - Not Ransomware
blbproxy.dll - Not Ransomware
UserLanguagesCpl.dll - Not Ransomware
```

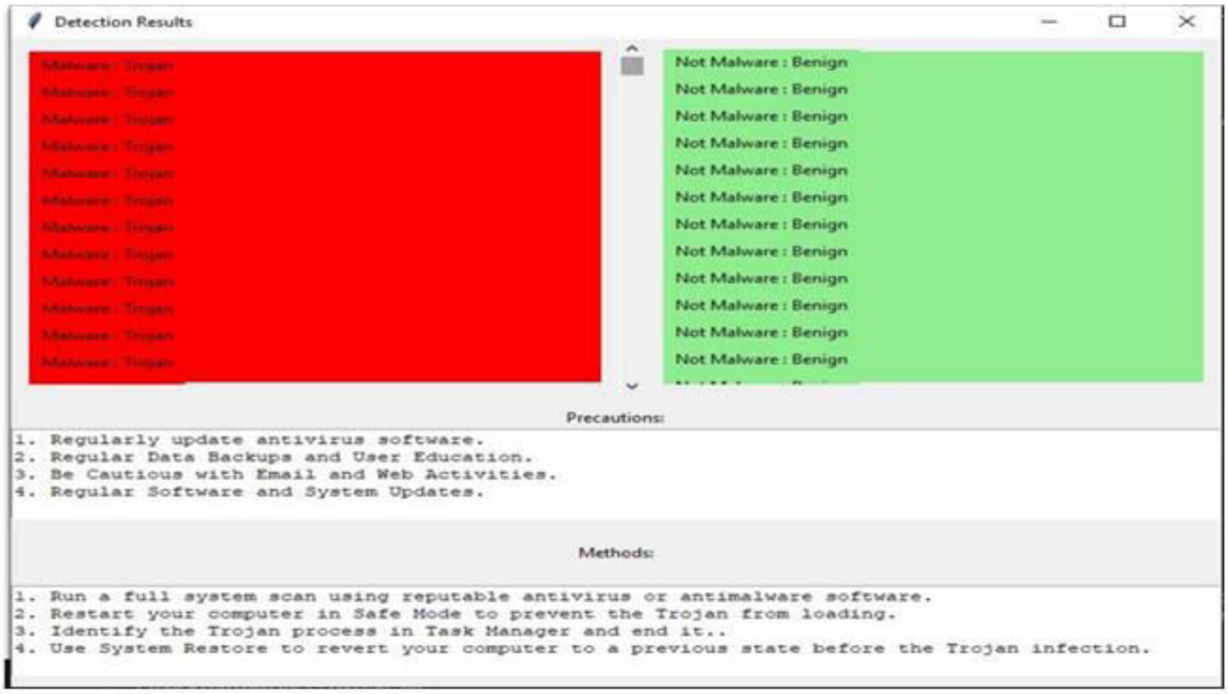
Precautions to Prevent Ransomware:

```
1. Regularly update antivirus software.
2. Avoid clicking on suspicious links or downloading unknown files.
3. Backup important data regularly.
4. Use a firewall to block unauthorized access.
```

Methods to Remove Ransomware:

```
1. Run a full system scan with antivirus software.
2. Use malware removal tools to detect and remove viruses.
3. Restore from a clean backup if necessary.
4. Consult with IT professionals for advanced malware removal.
```

Output for Ransomware



```
Malware : Trojan
Malware : Trojan
Malware : Trojan
Malware : Trojan
Malware : Trojan
Malware : Trojan
Malware : Trojan
Malware : Trojan
Malware : Trojan
Malware : Trojan
Malware : Trojan
Malware : Trojan
```

```
Not Malware : Benign
Not Malware : Benign
Not Malware : Benign
Not Malware : Benign
Not Malware : Benign
Not Malware : Benign
Not Malware : Benign
Not Malware : Benign
Not Malware : Benign
Not Malware : Benign
Not Malware : Benign
Not Malware : Benign
```

Precautions:

```
1. Regularly update antivirus software.
2. Regular Data Backups and User Education.
3. Be Cautious with Email and Web Activities.
4. Regular Software and System Updates.
```

Methods:

```
1. Run a full system scan using reputable antivirus or antimalware software.
2. Restart your computer in Safe Mode to prevent the Trojan from loading.
3. Identify the Trojan process in Task Manager and end it..
4. Use System Restore to revert your computer to a previous state before the Trojan infection.
```

Output for Trojan

VI. CONCLUSION

Machine learning has revolutionized malware detection by offering higher accuracy, adaptability to new threats, and efficient resource utilization. These models excel inanalyzing vast datasets for patterns of malicious behavior, enhancing cybersecurity measures significantly. However, challenges such as data quality, model interpretability, and cybersecurity risks must be addressed for optimal performance and reliability. Despite these challenges, the integration of machine learning with traditional security methods and human expertise promises a formidable defense against evolving cyber threats, shaping the future of cybersecurity strategies

REFERENCES

[1]https://link.springer.com/chapter/10.1007/9 78-3-030-04780-1_28
[2]https://link.springer.com/article/10.1631/FI TEE.1601325
[3]https://link.springer.com/article/10.1007/s0 0521-020-05309-4.
[4]https://onlinelibrary.wiley.com/doi/abs/10.1 002/cpe.5422
[5]https://dl.acm.org/doi/abs/10.1145/3154273
.3154326
[6]https://link.springer.com/chapter/10.1007/9 78-3-030-65384-2_5
[7] https://hal.science/hal-01704766/
[8]https://www.mdpi.com/1424-8220/23/3/1053
[9]https://ieeexplore.ieee.org/abstract/docume nt/9358835
[10]https://link.springer.com/article/10.1007/s 00500-014-1511-6
[11] “Malware Data Science: Attack Detection [13] “Practical Malware Analysis: The Hands- On Guide to Dissecting Malicious Software” by Michael Sikorski and Andrew Honig

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com