OXFORD UNIVERSITY PRESS | The Computer Journal

Antidote: A Comprehensive Suite for Proactive Cybersecurity with Non-Proactive Features

Journal:	*The Computer Journal*
Manuscript ID	Draft
Manuscript Type:	Original Article
Date Submitted by the Author:	n/a
Complete List of Authors:	Sachan, Rohit; Bennett University, SCSET Choudhary, Suhani; Bennett University, SCSET
Key Words:	Cybersecurity, Blockchain, Machine Learning
CCS classification:	Computing methodologies-Machine Learning - Machine Learning Approaches/Algorithms, Security and privacy-Intrusion/anomaly detection and malware mitigation, Security and privacy-Human and societal aspects of security and privacy

Antidote: A Comprehensive Suite for Proactive Cybersecurity with Non-Proactive Features

Rohit Kumar Sachan and Suhani Choudhary

SCSET, Bennett University, Greater Noida-201310, INDIA
Email: rohit.sachan@bennett.edu.in

As the digital landscape continues to evolve, the imperative for robust cybersecurity measures becomes increasingly critical. This paper introduces Antidote, an innovative cybersecurity suite designed to provide comprehensive protection against a spectrum of threats. Antidote encompasses both proactive and non-proactive features to fortify cybersecurity resilience. The suite's proactive components leverage advanced threat intelligence and machine learning to identify emerging threats. Additionally, Antidote's non-proactive features focus on enhancing incident response, recovery, and forensic capabilities, ensuring a holistic cybersecurity strategy. This paper delves into the architecture, functionalities, and efficacy of Antidote, highlighting its versatility in adapting to the dynamic nature of cyber threats and offering a paradigm shift in the approach to cybersecurity.

Keywords: Cybersecurity, Machine learning, Malware, Operating System, Phishing, System Health

Received 00 February 2024; revised 00 Month 2024

1. INTRODUCTION

In the rapidly evolving landscape of internet technology, 2023 witnessed a pronounced increase in global cybercrimes, influenced by significant geopolitical developments worldwide, including Russia's invasion of Ukraine and the G20 summit in India [1]. According to a cyber threat report [1], more than 400 million instances of threats are observed across an extensive network of 8.5 million endpoints in India. It implies an average of 761 malware instances of threats per minute. The report also shows a 12.5% share of behaviour-based and 87.5% share of signature-based detections of the total observed instances. The frequency and sophistication of cyber threats have reached unprecedented levels, and they scare the world. It necessitates innovative, versatile, multifaceted, and proactive cybersecurity solutions.

Since the technological advancement and paradigm shift, there has been a drastic rise in the new types of attacks where the conventional signature-based systems cannot keep up with the attacks. Additionally, the perpetual emergence of new malware strains further accentuates the need for signature-based solutions, as they rely on pre-existing knowledge. The conventional system is not capable of handling such real-world security issues [2].

Machine learning (ML), a subset of artificial intelligence (AI), can solve various classification and prediction problems. These techniques enable learning, reasoning, and decision-making outside of human interaction in the system. These techniques can also develop data-driven recommendations and decision systems, including cybersecurity [3, 4]. These techniques have also proven to help evolve malware detection systems.

In tandem with the challenges in malware detection, the ubiquity of phishing attacks poses a significant threat to individuals and organizations. Phishers continually refine their tactics, employing social engineering techniques and creating deceptive websites to pilfer sensitive information. Traditional methods, though valuable, need help to keep pace with the evolving sophistication of these malicious activities. ***This work introduces "Antidote" a pioneering cybersecurity suite that provides a versatile defence mechanism against cyber threats, combining proactive and non-proactive features for a comprehensive security approach.***

Antidote's proactive features encompass advanced threat detection mechanisms, leveraging cutting-edge artificial intelligence and machine learning algorithms. These advanced techniques enable us to stay ahead of the intricate strategies employed by phishers, offering a more resilient defence against deceptive practices. These features include PE (Portable Executable) file scanning and URL (Uniform Resource Locator) scanning, which has adaptability and learning capabilities. ML models enable us to stay ahead of the intricate strategies employed by phishers, offering a more resilient defence against deceptive practices.

In addition to its proactive capabilities, Antidote incorporates indispensable non-proactive features aimed at minimizing the impact of successful cyberattacks. These features include a RAM Booster (RB), a Junk Cleaner (JC), and an Overall System Health Checker (OSHC). The OSHC evaluates operating System health (OSH), network health (NH), hardware health (HH), application health (AH), resource health (RH), compliance health (CH), and system updates (SU). Note that none of these non-proactive features leverage ML capabilities.

In addition to both type of features, Antidote's user-friendly interface and intuitive management dashboard facilitate seamless integration of all features. ***Our suite aims to provide a versatile, robust, and user-friendly defence against evolving cyber threats, ensuring the resilience of digital ecosystems.***

In summary, our core contributions through this work are:

- **Comparative study:** We present a comparative study of some of the state-of-the-art techniques used to detect malware PE and malicious URLs. We compare studied works based on their objective, used approach, used ML, reported performance, and data set used in the paper.
- **Proposed Antidote Suite:** We incorporate both proactive and non-proactive features in our proposed suite. We also present the usability of both types of features for our suite.
- **Commands for non-proactive features:** The non-proactive features do not leverage with ML. They are based on the OS commands. So, we present the OS commands and the approach used in the Antidote suite.
- **Feature Analysis for proactive features:** Our proactive features leverage ML capabilities. For that, we use a standard ML pipeline. In feature engineering, we perform and analyse the feature based on the feature importance and feature correlation. In the next step, ML models use the selected features to detect the proactive aspects of cyber threats.
- **Result Analysis:** We used an ML-based approach to detect the malware/malicious PE and URLs. We have used TPOT, an autoML tool, to identify the best ML pipeline for proactive cybersecurity defence.
- **User-Friendly Interface:** We have developed a user-friendly interface for managing and monitoring the suite's functionalities. It will contribute to the overall usability and effectiveness of the cybersecurity suite.

The rest of the paper is organized as follows. In Section 2, we present state-of-the-art works related to detecting malware and phishing websites. Section 3 presents our proposed Antidote suite, and proposed approach in Section 4, followed by an evaluation along with the results in Section 5. We finally conclude in Section 6.

2. LITERATURE REVIEW

This section mainly discusses the state-of-the-art works related to proactive features of the proposed Antidote suite. These features include detecting malware (i.e., PE) and phishing websites (i.e., URL scanners). Here, we consider only proactive features based on their importance in the suite and the requirement of advanced domain knowledge.

In the papers [5, 6], the authors introduce an intelligent malware detection system (IMDS) using object-oriented association (OOA) mining. It analyses the API calls of PE. This system has a PE parser, an OOA rule generator, and a rule-based classifier. The proposed system is incorporated into the KingSoft Anti-Virus software. In the paper [7], the authors address the challenges of real-time malware detection. The approach emphasizes the balance between accuracy and detection time. It utilizes the analysis of API calls and the technical features of PE. The approach uses a chi-square measure during feature selection and a phi coefficient during feature classification. The reported performance in terms of accuracy is 98%, and the real-time detection time is 0.09 sec. In another paper [8], the authors present a deep learning-based method to detect PE malware based on the header content. The method uses a deep neural network during data training and a k-means clustering algorithm for data clustering. The clustering algorithm segments samples into two clusters: malware and benign. They report that the proposed approach is fast-to-use, high performance, and has low computational overhead. In [9], authors have proposed a novel approach to detect malware using to identify malicious internet protocol (IP) addresses based on reputation. The proposed approach uses dynamic malware analysis, cyber threat intelligence, ML, and big data forensics. In [10], the authors propose a scalable, cost-effective, and efficient approach to detect malware based on deep learning (DL). The proposed models show more than 16.56% improvement in accuracy compared to the studied state-of-the-art works. Similarly, in [11], authors propose an automated machine learning approach for malware detection. They use ML, DL, and convolutional neural networks (CNN) to detect static and dynamic malware.

In the paper [12], the authors work on detecting phishing web pages in real-time. They create a rule-based classifier based on the experience. This classifier contains five rules. Based on these rules, their PhishChecker application classifies the URLs as malicious and benign. The application tested over 100 Phistank and Yahoo directory dataset URLs. The reported accuracy of PhishChecker is 96.00%, and the false negative rate does not exceed 0.105. The low false alarm rate shows the effectiveness of the PhishChecker. Similarly, in [13], the authors present a lightweight tool to detect phishing URLs without visiting the webpage. This CatchPhish tool uses the hostname, full URL, TF-IDF (term frequency-inverse document frequency) features and keywords of malicious URLs. The random forest classifier achieves 94.26% accuracy on the self-created dataset and 98.25% on benchmark datasets. The datasets used in the work are Common-crawl, Alexa, and PhishTank. In another work [14], authors extract host-based features through lexical and statistical analysis. They use five ML classifiers to detect the phishing URLs over the self-created dataset of 1L URLs. They achieve the highest accuracy of 98.0% for the Naïve Bayes Classifier with a precision = 1, recall = 0.95 and F1-Score = 0.97. In [15], the authors propose a system with improved efficiency for detecting phishing websites. The proposed system uses six well-established ML models during experimentation over a public dataset of 88K websites. They use 112 features for each web sample. Of the six algorithms, XGBoost performs remarkable performance with 99.2% accuracy and 99.4% recall. The optimal run-time for the XGBoost is about 1500ms. In another work [16], authors develop a hybrid model of XGBoost and Firefly algorithms to detect phishing URLs. They use the firefly meta-heuristic algorithm for the feature selection and the tuning of the hyperparameters of XGBoost. The proposed model is evaluated over the two public datasets, i.e., the University of California and Irvine ML repository.

All studied works are summarized in Table 1. This summary includes the objective, approach, ML algorithms, performance of the reported model, dataset, and comments regarding the limitations and/or advantages of the work.

3. PROPOSED ANTIDOTE SUITE

The proposed Antidote suite integrates the five popular cybersecurity aspects that provide a comprehensive defence to our system against cyber threats. These features are:

1. PE Scanner
2. URL Scanner
3. RAM Booster (RB)
4. Junk Cleaner (JC)
5. Overall System Health Checker (OSHC)

The first two features, PE and URL scanner, are proactive, while the remaining three features- RB, JC, and OSHC, are non-proactive. This indicates that the PE and URL modules are enriched with ML integration. In contrast, RB, JC, and OSHC rely on OS commands for their respective functionalities. The subsequent subsections delve into the details of proactive and non-proactive features, elucidating their purposes and approaches. Section 4 explores how these features are implemented within the Antidote suite.

3.1. PE Scanner (Proactive):

Purpose: This feature focuses on scanning PE files, which include executable files, DLLs, and other binary formats.
Proactive Approach: Enriched with ML integration, this scanner can proactively identify and mitigate potential threats by learning from patterns and anomalies in the PE files.

3.2. URL Scanner (Proactive):

Purpose: Designed to scan URLs for potential threats, this feature is crucial for preventing users from accessing malicious websites.
Proactive Approach: Similar to the PE Scanner, it utilizes ML to analyze URLs and identify patterns associated with malicious content, providing protection against phishing and other online threats.

3.3. RAM Booster (Non-Proactive):

Purpose: Aims to optimize the RAM usage for better system performance.
Non-Proactive Approach: Utilizes OS commands to manage and boost RAM, ensuring efficient memory utilization but reacting to system demands rather than predicting them in advance.

3.4. Junk Cleaner (Non-Proactive):

Purpose: Focused on cleaning unnecessary and temporary files that accumulate over time, potentially causing system slowdowns.
Non-Proactive Approach: Relies on OS commands to identify and remove redundant files, enhancing system performance by freeing up storage space.

3.5. Overall System Health Checker (Non-Proactive):

Purpose: Provides a comprehensive assessment of the overall system health, checking for potential issues.
Non-Proactive Approach: Uses OS commands to run various checks and diagnostics, identifying and reporting on the current state of the system. It reacts to existing conditions rather than predicting potential future threats.

By integrating both proactive and non-proactive features, proposed suite appears to offer a comprehensive defense against cyber threats, addressing both immediate and potential risks to the system. The functional

TABLE 1: Summary of studied state-of-the-art works.

Ref.	Objective	Approach	ML	Performance	Dataset	Comments
[5, 6]	Malware	OAA and Rule-based classifier	NB SVM DT IMDS	DR = 81.91,% ACC = 83.86% DR = 96.88%, ACC = 90.54% DR = 96.21%, ACC = 91.49% DR = 97.19%, ACC = 93.07%	Norton AntiVirus McAfee VirusScan	KingSoft's AntiVirus software
[7]	Malware	Chi-square and Phi coefficient	-	ACC = 98%, DTi = 0.09s	600 PE files Vxheavens.com	Accuracy improvement Limited malware categories
[8]	Malware	Neural network and K-means clustering	-	ACC = 91.13%, PR = 90.76% RE = 92.06%, F1 = 91.40%	DS1 = 4K DS2 = 9K	Fast-to-use, high performance, low computational overhead
[9]	Malware	IP Reputation	DT	-	-	High false alarm rate
[10]	Malware	DL-based approach	SVM	ACC = 99.06% ACC 16.56% Increase	Malimg	Cost-effective and efficient
[11]	Malware	DL and ML	CNN	-	SOREL-20M EMBER-2018	Automated tool
[12]	Phishing	Rule-based classifier	-	ACC = 96%, FN < 10.5%	PhishTank Yahoo	Low false negative rate, Small dataset used
[13]	Phishing	TF-IDF with hand-crafted features	RF	$\text{ACC}_1 = 94.2\%, ACC_2 = 98.2\%$	Common-crawl Alexa, PhishTank	Light-weight
[14]	Phishing	Lexical and statistical features	LR RF NB DT KNN	97.7%, RE = 96% ACC = 98.03%, RE = 96% ACC = 97.18%, RE = 95% ACC = 98.02%, RE = 96% ACC = 97.99%, RE = 97%	1L URLs	Content-based features required
[15]	Phishing	112 features	LR KNN NB RF SVM XGBoost	ACC > 93% ACC = 99.2%, PR = 99.1% RE = 99.4%, SP = 99.1% RT = 1500ms	Public dataset of 58K legitimate websites and 30K phishing websites	Low accuracy and High run time
[16]	Phishing	Hybrid of XGBoost Firefly algorithm	XGBoost	-	UC and Irvine ML repository	More testing required

^{DT} Decision Tree, ^{RF} Random Forest, ^{NB} Naive Bayes, ^{SVM} Support Vector Machine, ^{LR} Logistic Regression, ^{KNN} K-Nearest Neighbors, ^{XGBoost} Extreme Gradient Boosting, $^{TF-IDF}$ Term Frequency-Inverse Document Frequency
^{ACC} Accuracy, ^{PR} Precision, ^{RE} Recall, ^{F1} F1 Score, ^{DR} Detection Rate, ^{DTi} Detection Time, ^{SP} Specificity, ^{RT} Run time
^{UC} University of California, ^{DS} Dataset
$^{-}$ not know

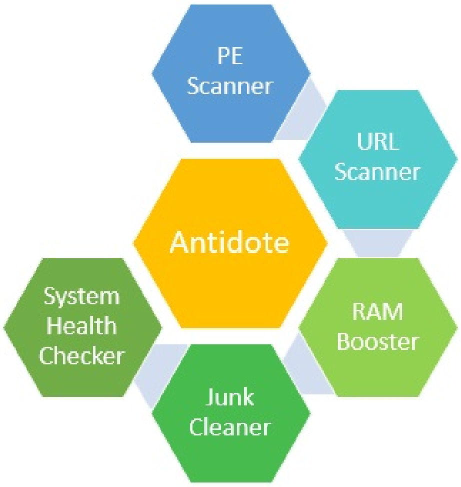


FIGURE 1: The functional diagram of Antidote.

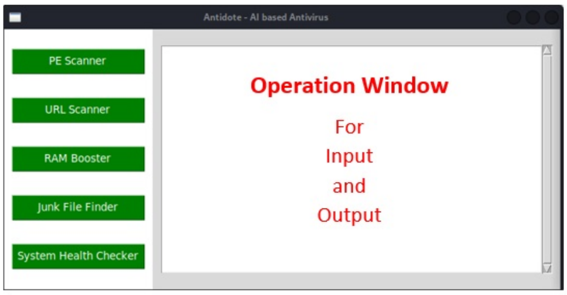


FIGURE 2: The user interface of Antidote.

diagram of Antidote is illustrated in Figure 1 and user interface is illustrated in Figure 2.

4. PROPOSED APPROACH

In this section, we present the approach in which we discuss how both proactive and non-proactive features are integrated in Antidote. Regarding proactive features, we follow the standard ML pipeline: data collection, pre-processing, feature engineering, ML algorithm, and evaluation. Each pipeline step is elaborated individually, except for the evaluation (cf. Section 5). These steps are illustrated in Figure 3. We currently use only supervised learning models in the proposed suite based on current limitations.

4.1. Data Collection and Pre-processing

A single dataset does not have sufficient information for both scanners, so we use different datasets for each type. These data are collected from public repositories like [17] and [18]. The collected datasets have different sizes and columns (i.e., features) from each other. The PE malware dataset has legitimate label values 0 and 1, where 0 means benign, and 1 means malicious malware. Similarly, in the phishing URL dataset, we have class label values -1 and 1, where -1 means benign URL and 1 means malicious URL. So, during the pre-processing stage, annotating the labels of datasets is unnecessary. It means the original datasets are used for further steps of the pipeline.

4.2. Feature Engineering

Feature engineering mainly includes feature construction (or extraction) and feature selection. We do not construct new features for both scanner modules in this work. Here, we work only on the feature selection. We perform feature selection based on the feature importance and feature correlation. We use one of the popular methods, i.e., ExtraTreesClassifier, to find the importance of the feature and select the best feature based on the selection criteria. After the feature importance, we use the correlation method to find the correlation between the features so we may drop the highly correlated features from the feature set.

4.3. ML Algorithms

This one is the crucial step in the ML pipeline because selecting the correct ML model from the dozens of models and tuning their hyperparameter is very difficult. So, to overcome this problem, we use a TPOT AutoML tool [19, 20], which reports the best ML pipeline for the data configuration after executing the configured ML models over the different hyperparameters. We measure the performance of the reported model based on the Balanced-Accuracy, Accuracy, Precision, Recall, and F1-score statical measures. After identifying the best ML classifier/model, we incorporate the reported model into our proposed Antidote suite. Here, we execute TPOT independently for both scanners over the respective datasets.

In addition to proactive features, Antidote incorporates non-proactive features using OS commands. We innovatively use OS commands to achieve superior performance. The detail of each non-feature feature is:

4.4. RAM Booster

Random Access Memory (RAM) is pivotal in ensuring optimal system performance. In the current state of work, we use a manual intervention mechanism for RAM booster. This approach first searches the top 5 processes with the highest RAM consumption and then kills these processes. In future, we are planning to introduce a pioneering approach for RAM Booster, which has artificial intelligence (AI) capabilities. That approach will use an AI-driven process termination instead of a manual intervention mechanism for RAM booster. We will use the 'kill' command with the AI-enabled process ID to kill the processes.

4.5. Junk Cleaner

Antidote offers a comprehensive solution to identify and eliminate the unnecessary files that can accumulate over time. We present a dual-pronged strategy to clean junk files. It identifies duplicate files within a specified directory by calculating unique hash values using the MD5 algorithm. This process ensures a meticulous examination of the file structure, reducing redundant data. Secondly, the feature targets and removes files deemed as 'junk', characterized by specific file extensions such as '.tmp', '.bak', '.log', and '.swp'. This approach empowers users to maintain a lean and organized digital workspace.

4.6. Overall System Health Checker

Our health checker tool is equipped with diverse health-check functionalities that provide real-time system insights. It incorporates distinct functionalities, each focusing on a specific facet of system health. It includes hardware, operating system, network connectivity, application functionality, and resource utilization metrics. It also facilitates checking for critical updates, ensuring the system remains current and secure. The associated OS commands with the OSHC are 'subprocess', 'platform', and 'psutil'.

5. EVALUATION AND RESULTS

We perform our evaluation and development over the Kali Linux within the Python v3.11.4 environment with supporting libraries like NumPy v1.24.3, Pandas v2.0.1, and Scikit-learn v1.3.0. The hardware configuration of the test machine is Intel® Core(TM) i7-11800H 11th Gen CPU with 4.00 GB NVIDIA GeForce RTX 3050 Ti and 16.00 GB RAM.

5.1. Dataset

We use the open-source and publicly available Malware dataset [17] for the PE scanner and the Phishing website dataset [18] for the URL scanner. The malware dataset has 138047 unique records of the malware with 55 features and labels. Similarly, the phishing dataset has 11054 unique URL records with 30 features and labels.

5.2. Data Analysis

We analyse both datasets to understand the insights of data. Our analysis identifies that the malware dataset has 96724 malware and 41323 benign malware records, and similarly, the phishing dataset has 6157 malware and 4897 benign URLs. It implies approximately 70% malware instances and 30% benign instances in the malware dataset, approximately 56% phishing URLs and 44% legitimate URLs in the phishing dataset. We follow the 80-20 rule to segregate both datasets into training and testing data.

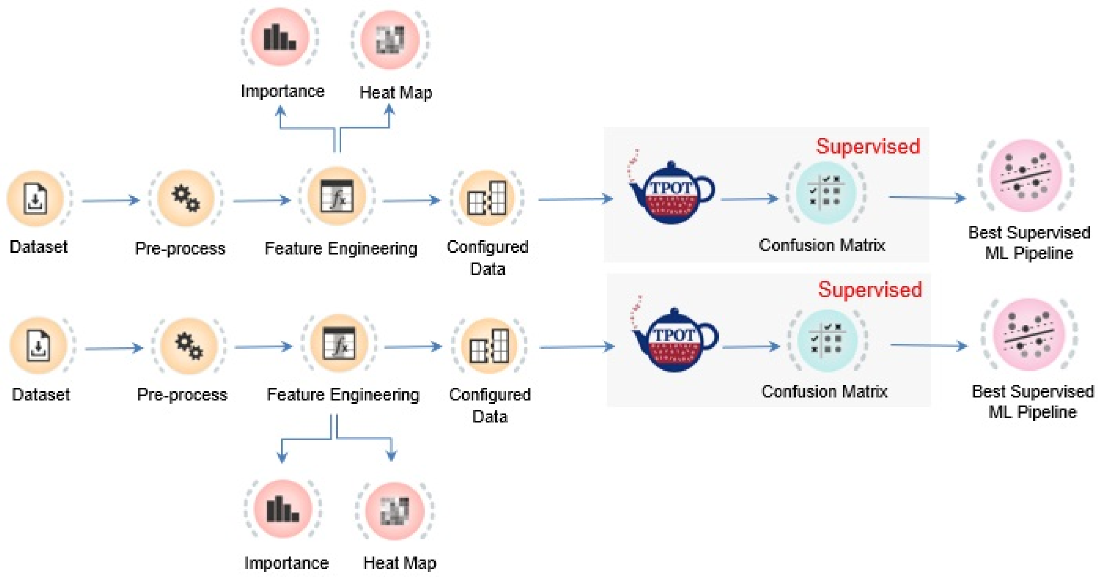


FIGURE 3: The approach for proactive features of Antidote.

5.3. Feature Engineering

We apply *ExtraTreesClassifier* with default hyperparameters for the feature selection over the malware dataset containing 55 features. The computed feature importance of all 55 features is plotted in Figure 4. We use *ExtraTreesClassifier* due to its ease of use.

Similarly, we compute the feature importance of all 30 features of the phishing dataset, and the same is plotted in Figure 5.

After calculating the importance of the features, we selected features more significant than 2.0% for the malware dataset. This selection selects the 13 features out of 55 features. After determining their importance, we compute and check the correlation between them. The heatmap of the correlation matrix is plotted in Figure 6.

Similarly, due to their uniqueness, we consider all 30 features for the feature correlation for the phishing dataset. The heatmap of the computed correlation matrix is plotted in Figure 7.

Both heatmaps illustrate that there is very little correlation between the selected features. It signifies the importance and significance of all features. Based on that, our feature engineering process identifies 13 features for the malware dataset and 30 for the phishing dataset. All these features are used in TPOT to identify the best ML algorithm. All the features obtained after the feature engineering are listed in Appendix.

5.4. Results

As mentioned earlier, we use the AutoML tool called TPOT to identify the best ML algorithm for supervised learning. We use TPOT due to its easy-in-use functionality. We run the TPOT classifier with the default configuration and default hyperparameters. We also report the Balanced-Accuracy, Accuracy, Precision, Recall, and F1 score for the best-reported ML algorithm in terms of accuracy.

TPOT reports the pipeline of *XGBClassifier* and *LinearSVC* to achieve an accuracy of 98.69% as the best classifier for the malware dataset or PE scanner. The hyperparameters of the *LinearSVC* are C=1.0, dual=False, loss=squared_hinge, penalty=12, tol=le-05 and hyperparameters of the *XGBClassfier* are learning_rate=1.0, max_depth=1, min_child_weight=17, n_estimators=100, n_jobs=1, subsample=1.0, verbosity=0.

Similarly, TPOT reports *GradientBoostingClassifier* with learning_rate=0.1, max_depth=9, max_features=0.5, min_samples_leaf=3, min_samples_split=14, n_estimators=100, subsample=0.3 hyperparameters to achieve accuracy of 96.92% as the best classifier for the phishing dataset or URL scanner. All other hyperparameters have default values and reported performance metrics are listed in Table 2 for both scanners. The reported confusion matrix is plotted in Figure 8 respectively.

After the ML phase, we associate both identified ML classifiers with the front end of the respective functionality of the Antidote suite. The user interfaces of the Antidote suite are shown in Figures 9-13.

6. CONCLUSION AND FUTURE SCOPE

This work introduces Antidote, an all-encompassing cybersecurity suite featuring proactive and non-proactive components. The suite comprises two proactive features, a PE Scanner and a URL Scanner, alongside three non-proactive features: a RAM Booster, a Junk Cleaner, and an Overall System Health Checker.

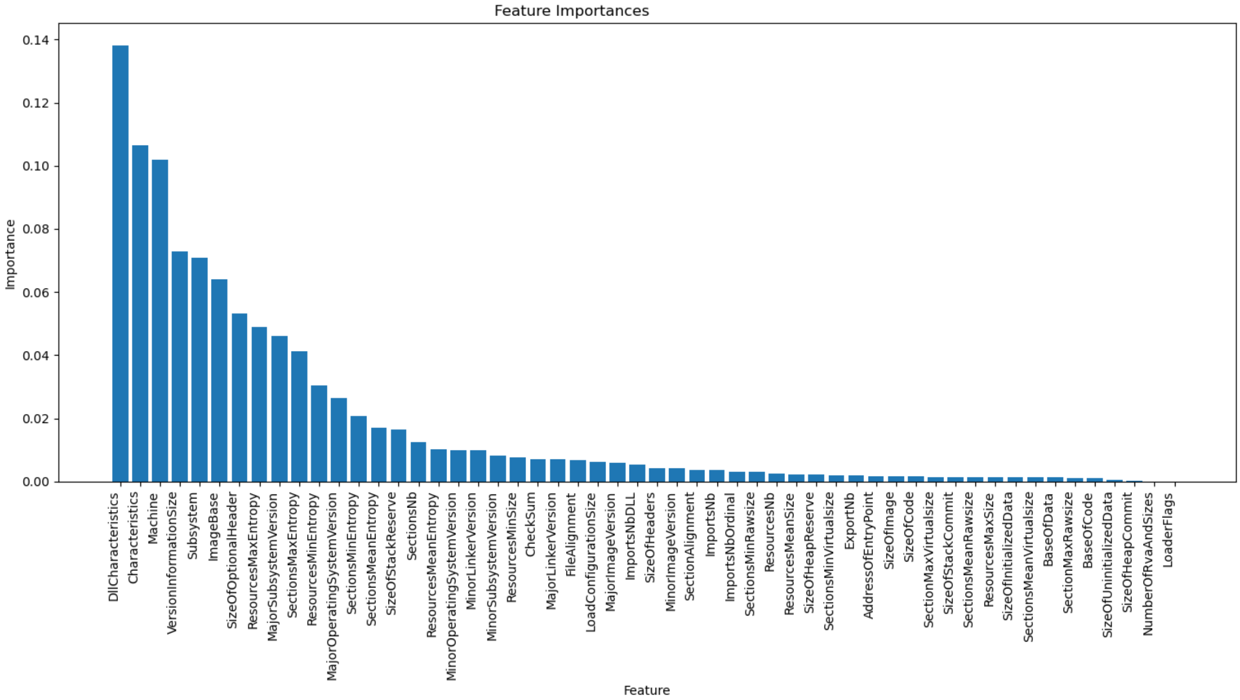


FIGURE 4: Feature importance of all 55 features of malware dataset.

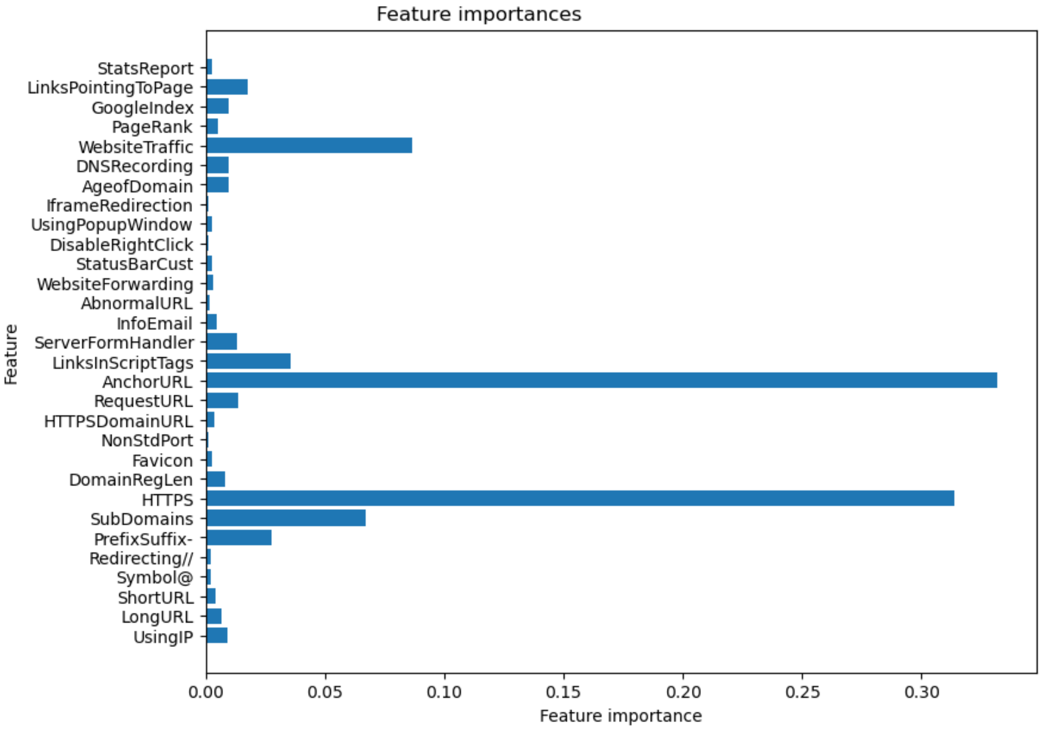


FIGURE 5: Feature importance of all 30 features of phishing dataset.

TABLE 2: Reported performance by best classifiers.

Feature	Dataset	Classifiers	Balance-Accuracy (%)	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
PE Scanner	Malware	XGBClassifier and LinearSVC	98.89	99.02	98.18	98.56	98.37
URL Scanner	Phishing URL	GradientBoostingClassifier	96.78	96.92	96.57	97.98	97.27

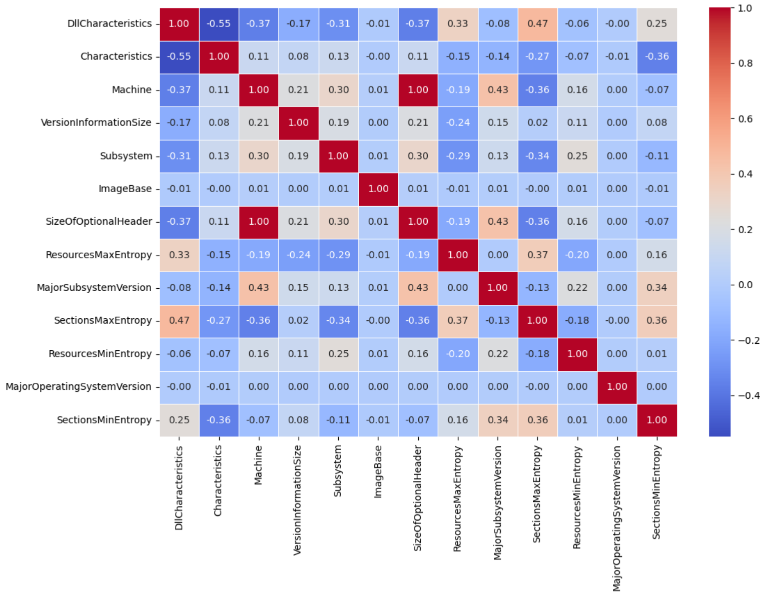


FIGURE 6: Heatmap of the feature correlation of 13 features of malware dataset.

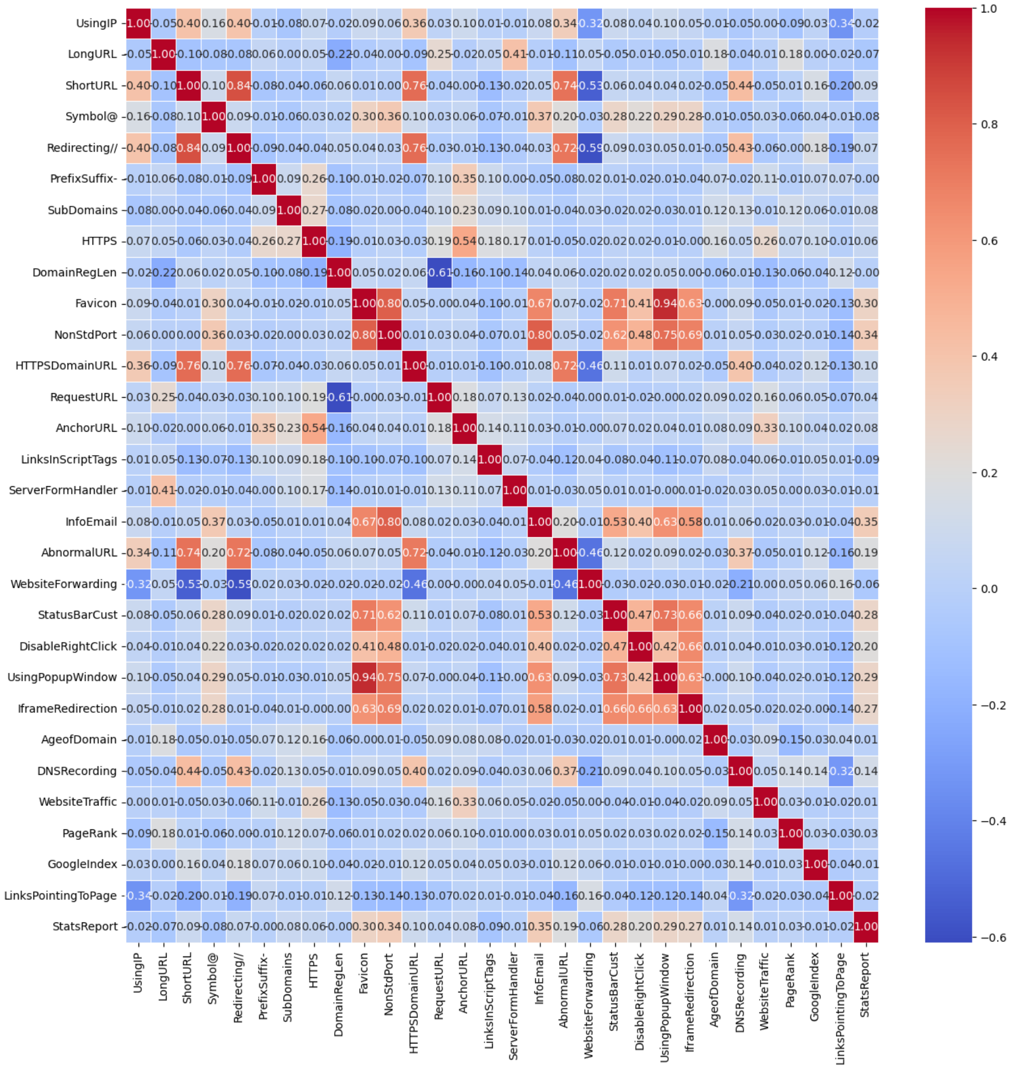
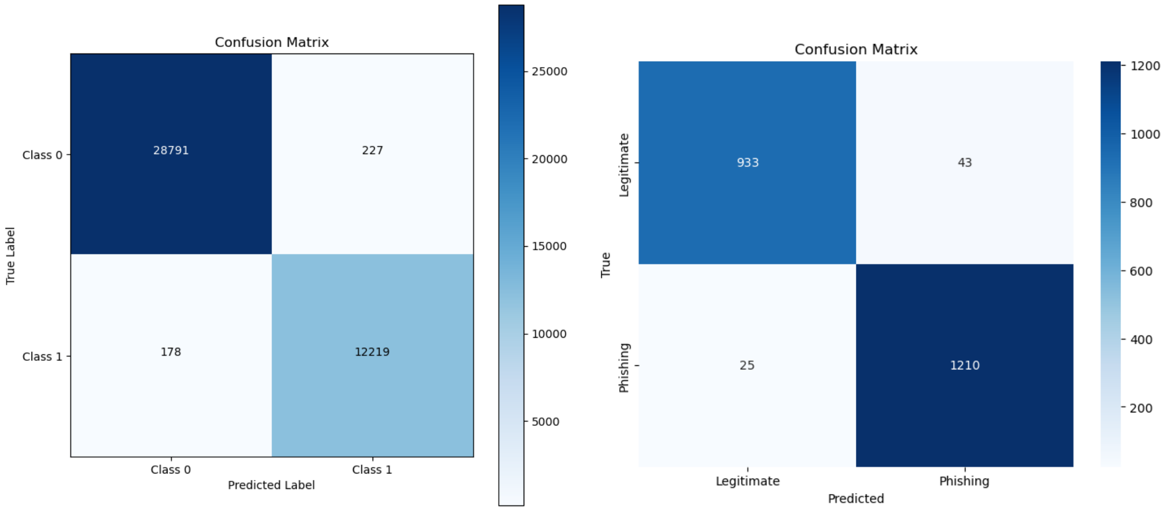


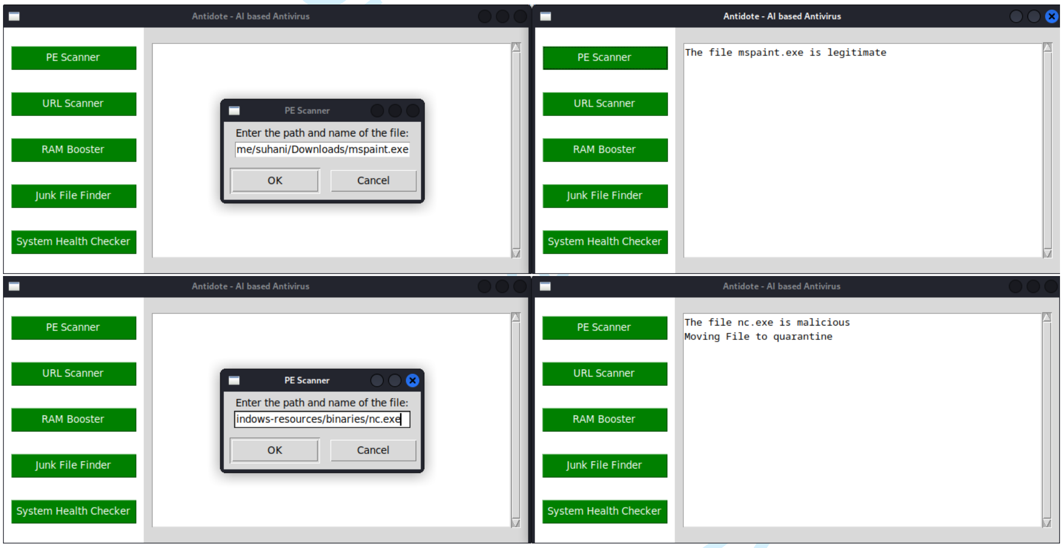
FIGURE 7: Heatmap of the feature correlation of 30 features of phishing dataset.



(a) PE scanner.

(b) URL scanner.

FIGURE 8: Confusion matrix.



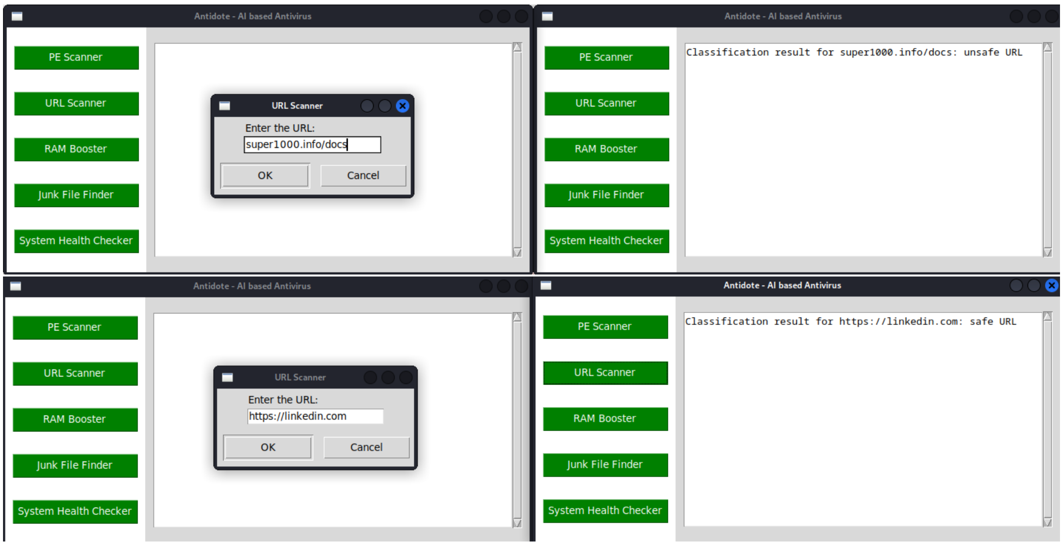
Enter the path and name of the file:
me/suhani/Downloads/mspaint.exe

```
The file mspaint.exe is legitimate
```

Enter the path and name of the file:
indows-resources/binaries/nc.exe

```
The file nc.exe is malicious
Moving File to quarantine
```

FIGURE 9: PE Scanner Interface.



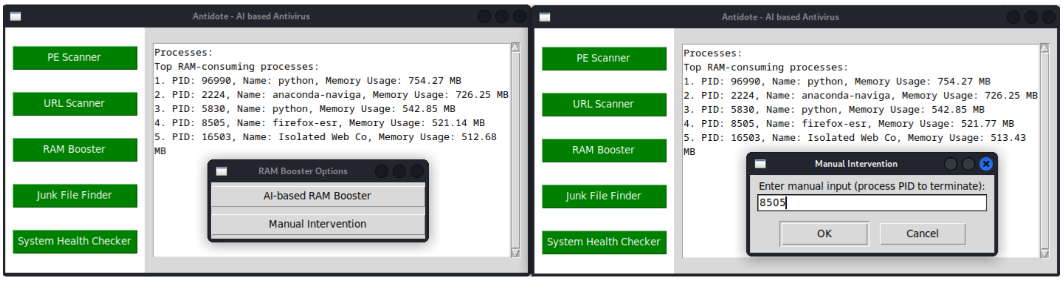
Enter the URL:
super1000.info/docs

```
Classification result for super1000.info/docs: unsafe URL
```

Enter the URL:
https://linkedin.com

```
Classification result for https://linkedin.com: safe URL
```

FIGURE 10: URL Scanner Interface.

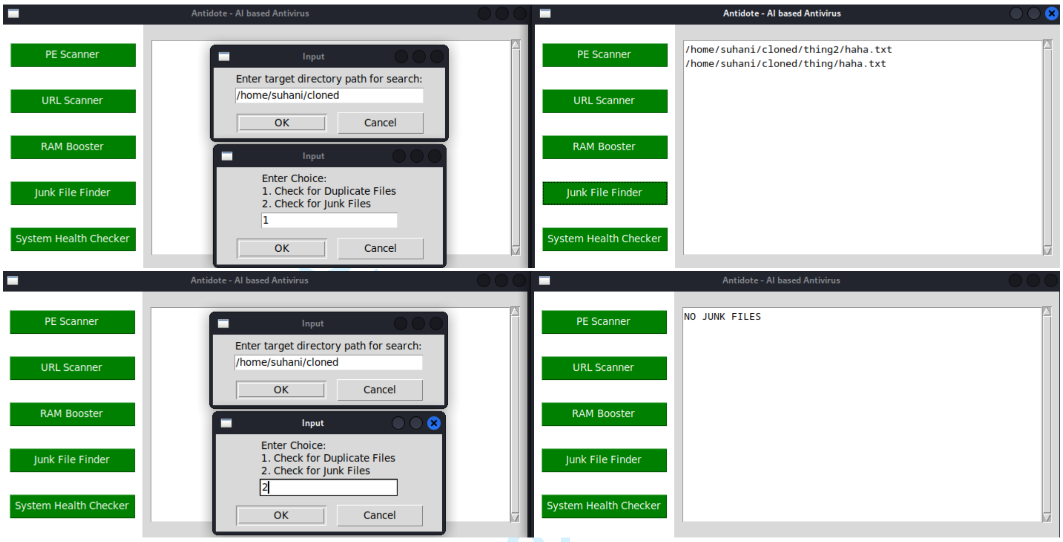


```
Processes:
Top RAM-consuming processes:
1. PID: 96990, Name: python, Memory Usage: 754.27 MB
2. PID: 2224, Name: anaconda-naviga, Memory Usage: 726.25 MB
3. PID: 5830, Name: python, Memory Usage: 542.85 MB
4. PID: 8505, Name: firefox-esr, Memory Usage: 521.14 MB
5. PID: 16503, Name: Isolated Web Co, Memory Usage: 512.68
MB
```

```
Processes:
Top RAM-consuming processes:
1. PID: 96990, Name: python, Memory Usage: 754.27 MB
2. PID: 2224, Name: anaconda-naviga, Memory Usage: 726.25 MB
3. PID: 5830, Name: python, Memory Usage: 542.85 MB
4. PID: 8505, Name: firefox-esr, Memory Usage: 521.77 MB
5. PID: 16503, Name: Isolated Web Co, Memory Usage: 513.43
MB
```

Enter manual input (process PID to terminate): 8505

FIGURE 11: RAM Booster Interface.



Enter target directory path for search: /home/suhani/cloned

Enter Choice:
1. Check for Duplicate Files
2. Check for Junk Files

1

```
/home/suhani/cloned/thing2/haha.txt
/home/suhani/cloned/thing/haha.txt
```

Enter target directory path for search: /home/suhani/cloned

Enter Choice:
1. Check for Duplicate Files
2. Check for Junk Files

2

```
NO JUNK FILES
```

FIGURE 12: Junk File Cleaner Interface.

Proactivity is instilled in the system through the integration of machine learning, while non-proactive features utilize operating system commands.

The TPOT AutoML tool is employed for the proactive features to identify the most suitable pipeline from a set of configured classifiers and their corresponding hyperparameters. Leveraging the best-reported pipeline, we achieved a balance-accuracy of 98.89% for the PE malware scanner and 96.78% for the phishing scanner.

In the future, we aim to enhance the proposed suite by integrating a more comprehensive dataset, thereby improving its robustness and accuracy. Additionally, our plans include introducing proactivity to the non-proactive features. Furthermore, we are incorporating intrusion detection capabilities into the suite, transforming it into a more comprehensive cybersecurity suite.

APPENDIX

List of features for PE Scanner: 13

DllCharacteristics, Characteristics, Machine, VersionInformationSize, Subsystem, ImageBase, SizeOfOptionalHeader, ResourcesMaxEntropy, MajorSubsystemVersion, SectionsMaxEntropy, ResourcesMinEntropy, MajorOperatingSystemVersion, SectionsMinEntropy.

List of features for URL Scanner: 30

Using IP, LongURL, ShortURL, Symbol@, Redirecting//, PrefixSuffix-, SubDomains, HTTPS, DomainRegLen, Favicon, NonStdPort, HTTPSDomainURL, RequestURL, AnchorURL, InScriptTags, ServerFormHandler, InfoEmail, AbnormalURL, WebsiteForwarding, StatusBarCust, DisableRightClick, UsingPopupwindow, IframeRedirection, AgeofDomain, DNSRecording, WebsiteTraffic, PageRank, GoogleIndex, LinksPointingToPage, StatsReport.

DECLARATIONS

Funding No Funding of any sort if received for this manuscript.

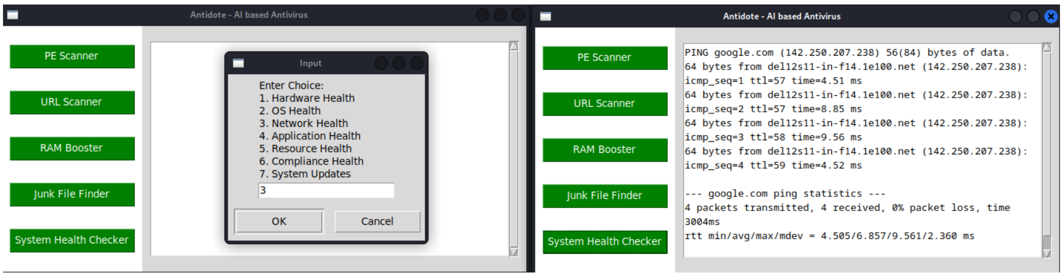
Disclosure of potential conflicts of interest Authors declare that no conflicts of interests of any sort is associated with the manuscript.

Research involving human participants and/or animals. Not Applicable.

Data availability Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Code availability. Not applicable to this article.

Acknowledgment. No acknowledgement available to



```
Enter Choice:
1. Hardware Health
2. OS Health
3. Network Health
4. Application Health
5. Resource Health
6. Compliance Health
7. System Updates
3
```

```
PING google.com (142.250.207.238) 56(84) bytes of data.
64 bytes from del12s11-in-f14.1e100.net (142.250.207.238):
icmp_seq=1 ttl=57 time=4.51 ms
64 bytes from del12s11-in-f14.1e100.net (142.250.207.238):
icmp_seq=2 ttl=57 time=8.85 ms
64 bytes from del12s11-in-f14.1e100.net (142.250.207.238):
icmp_seq=3 ttl=58 time=9.56 ms
64 bytes from del12s11-in-f14.1e100.net (142.250.207.238):
icmp_seq=4 ttl=59 time=4.52 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time
3004ms
rtt min/avg/max/mdev = 4.505/6.857/9.561/2.360 ms
```

FIGURE 13: Overall System Health Checker Interface.

this article.

Informed consent. All the authors have confirmed the consent to submit the paper to the Journal.

Authors' contributions. The authors confirm contribution to the paper as follows:
• **Rohit Kumar Sachan:** Study conception and design, Analysis and interpretation of results, Verification of results, Draft manuscript preparation. Critically revised the work, Supervised the finding of the work. • **Suhani Choudhary:** Study conception and design, Data collection, Analysis and interpretation of results. All authors reviewed the paper and approved the final version of the manuscript.

REFERENCES

[1] Data Security Council of India (DSCI) and SEQRITE. India cyber threat report 2023. Accessed: 15/01/2024.

[2] Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., and Xu, M. A survey on machine learning techniques for cyber security in the last decade. *IEEE access*, **8**, 222310–222354.

[3] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., and Wang, C. Machine learning and deep learning methods for cybersecurity. *Ieee access*, **6**, 35365–35381.

[4] Sarker, I. H., Kayes, A., Badsha, S., Alqahtani, H., Watters, P., and Ng, A. Cybersecurity data science: an overview from machine learning perspective. *Journal of Big data*, **7**, 1–29.

[5] Ye, Y., Wang, D., Li, T., and Ye, D. IMDS: Intelligent malware detection system. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1043–1047.

[6] Ye, Y., Wang, D., Li, T., Ye, D., and Jiang, Q. An intelligent PE-malware detection system based on association mining. *Journal in computer virology*, **4**, 323–334.

[7] Belaoued, M. and Mazouzi, S. A Chi-Square-Based Decision for Real-Time Malware Detection Using PE-File Features. *J. Inf. Process. Syst.*, **12**, 644–660.

[8] Rezaei, T., Manavi, F., and Hamzeh, A. A PE header-based method for malware detection using clustering and deep embedding techniques. *Journal of Information Security and Applications*, **60**, 102876.

[9] Usman, N., Usman, S., Khan, F., Jan, M. A., Sajid, A., Alazab, M., and Watters, P. Intelligent dynamic malware detection using machine learning in IP reputation for forensics data analytics. *Future Generation Computer Systems*, **118**, 124–141.

[10] Shaukat, K., Luo, S., and Varadharajan, V. A novel deep learning-based approach for malware detection. *Engineering Applications of Artificial Intelligence*, **122**, 106030.

[11] Brown, A., Gupta, M., and Abdelsalam, M. Automated machine learning for deep learning based malware detection. *Computers & Security*, **137**, 103582.

[12] Ahmed, A. A. and Abdullah, N. A. Real time detection of phishing websites. *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 1–6. IEEE.

[13] Rao, R. S., Vaishnavi, T., and Pais, A. R. CatchPhish: detection of phishing websites by inspecting URLs. *Journal of Ambient Intelligence and Humanized Computing*, **11**, 813–825.

[14] Kumar, J., Santhanavijayan, A., Janet, B., Rajendran, B., and Bindhumadhava, B. Phishing website classification and detection using machine learning. *2020 international conference on computer communication and informatics (ICCCI)*, pp. 1–6. IEEE.

[15] Bahaghighat, M., Ghasemi, M., and Ozen, F. A high-accuracy phishing website detection method based on machine learning. *Journal of Information Security and Applications*, **77**, 103553.

[16] Jovanovic, L., Jovanovic, D., Antonijevic, M., Nikolic, B., Bacanin, N., Zivkovic, M., and Strumberger, I. Improving phishing website detection using a hybrid two-level framework for feature selection and xgboost tuning. *Journal of Web Engineering*, **22**, 543–574.

[17] Kaggle. Malware. Accessed: 06/09/2023.

[18] Kaggle. Phishing website Detector. Accessed: 15/09/2023.

[19] TOPT. Python Automated Machine Learning tool (TPOT). Accessed: 24/11/2023.

[20] Le, T. T., Fu, W., and Moore, J. H. Scaling tree-based automated machine learning to biomedical big data with a feature set selector. *Bioinformatics*, **36**, 250–256.