OAIJSE

OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

RANSOMWARE DETECTION AND CLASSIFICATION USING MACHINE LEARNING

Pramod G. Patil[1], Anmol S. Budhewar[2], Dr.Naresh Thoutam[3], Litesh R. Patel[4], Srujal D. Laware[5], Komal M. Mahajan[6], Kaustubh B.Khairnar[7]

Assistant Professor, *Department of Computer Engineering, SITRC, Nashik-422213, India[1]*
Assistant Professor, *Department of Computer Engineering, SITRC, Nashik-422213, India[2]*
Assistant Professor, *Department of Computer Engineering, SITRC, Nashik-422213, India[3]*
Department of Computer Engineering, SITRC, Nashik-422213, India[4]
Department of Computer Engineering, SITRC, Nashik-422213, India[5]
Department of Computer Engineering, SITRC, Nashik-422213, India[6]
Department of Computer Engineering, SITRC, Nashik-422213, India[7]
pgpatil11@gmail.com[1], anmolsbudhewar@gmail.com[2], naresh1060@gmail.com[3], liteshp121@gmail.com[4], srujallaware2516@gmail.com[5], sakshimahajan055@gmail.com[6], kaustubhkhairnar1711@gmail.com[7]

Abstract:* Ransomware has emerged as a widespread menace in the digital realm, inflicting considerable financial losses and disrupting vital services for both individuals and organizations. Traditional signature-based detection methods are proving inadequate against the ever-evolving strategies employed by cybercriminals. This research introduces an inventive strategy to counter ransomware threats by leveraging machine learning techniques for effective detection and classification. The study makes a valuable contribution to the ongoing cybersecurity efforts by presenting a resilient and adaptive solution for identifying and categorizing ransomware. Through the utilization of machine learning, this approach establishes a proactive defence mechanism against ransomware threats, ensuring the protection of sensitive data, financial resources, and critical infrastructure from malicious attacks in the contemporary digital landscape.

Keywords: *Ransomware, Machine Learning, Cybersecurity, Threat Detection, Classification, Adaptive Defense, Cyber Threads, Digital Security, Data Protection*

I INTRODUCTION

In today's interconnected digital environment, ransomware attacks pose a serious threat to data security. This research focuses on leveraging machine learning to enhance the detection and classification of ransomware, utilizing features from both static and dynamic analyses. Motivated by the urgent need to protect individuals, organizations, and critical infrastructure, this study aims to empower cybersecurity professionals with advanced tools. The challenge addressed is the evolving nature of ransomware variants, emphasizing the development of a robust, real-time detection and classification system to mitigate financial, operational, and reputational risks associated with these kinds of malicious attacks / Cyber Attacks.

II LITERATURE

Malicious attacks, including ransomware, pose severe security threats across various industries. Conventional anti ransomware system struggles against sophisticated attacks, necessitating innovative solutions. A Feature selection-based framework employing machine learning algorithms, including Decision Tree, Random Forest, Naïve Bayes, Logistics Regression, and Neural Network, to classify ransomware security levels. The Proposed framework is evaluated datasets, demonstrating its effectiveness in detection and prevention [1]. Ransomware employs encryption to render data inaccessible, causing substantial harm across governments, corporations, and private users. In response to the proliferation of these threats, researchers have proposed diverse detection and classification schemes, predominantly utilizing advanced machine learning techniques [2]. Ransomware attacks pose severe threats to data security, causing privacy breaches, financial losses, and reputational damage [3]. Ransomware, an enduring threat, prompts an ongoing battle between evolving techniques and detection methods. Ransomware remains a persistent threat, prompting an ongoing arms race between its development and detection techniques. Existing detection systems, while widely used, often struggle with the reactive nature of ransomware, necessitating continuous evolution [4]. Ransomware, a form of malware, encrypts user data, blocking access until a ransom is paid. Evolving the behavior of ransomware makes traditional detection and classification methods less effective. Attackers employ metamorphic and polymorphic techniques to evade signature-based systems [5]. As web applications proliferate, the security landscape becomes increasingly precarious. Traditional intrusion detection systems, focusing on individual requests, struggle to address evolving cyber threats and are limited to known vulnerabilities [6]. The global impact of sophisticated targeted attacks, emphasizing the challenges in detection. Leveraging Microsoft’s Sysmon tool, we propose a real-time method to detect malicious tools by analyzing DLL information. Our approach focuses on creating “common DLL lists” for identifying malicious processes universally, and we implement a practical detection system using Elastic Stack as a Security Information and Event Management (SIEM). Evaluation with four US-CERT introduced Tools demonstrates our method’s effectiveness, successfully detecting China Chopper, Mimikatz, and PowerShell Empire with minimal false positive. The common DLL lists prove valuable for real-time detection using Elastic Stack [7]. As the dominant mobile operating system, Android is widely utilized for everyday activities, making it a prime target for hackers seeking to compromise personal information. To counteract this threat, a malware detection technique named MaplDroid is introduced in this paper. MaplDroid statically analyses application files by extracting features from the manifest file, employing a Naïve Bayes-based supervised learning model to classify applications as benign or malicious. The proposed technique demonstrated high efficiency with a Recall score of 99.12 [8]. Mobile Ad Hoc Networks (MANETs) are infrastructure-less networks crucial for communication during network failures. This manuscript introduces a Time Interval Based Blockchain Model (TIBBM) for security in MANETs, identifying malicious nodes. TIBBM establishes a Blockchain information structure, enabling the identification of malicious nodes at specified intervals. Compared to traditional models, TIBBM demonstrates superior performance in detecting malicious nodes during data transmission in MANETs.

2.1 Methodology

[1] Dataset Acquisition Collect a diverse dataset encompassing both ransomware and benign software samples to ensure comprehensive analysis. Perform data cleaning and preprocessing tasks, including file normalization, feature extraction, and noise reduction. [2] Feature Selection utilizes feature selection technique for identifying pertinent features from both static and dynamic analyses. Incorporate diverse features to enhance the model’s ability to distinguish between ransomware and benign software effectively. [3] Transfer Learning Implement transfer learning techniques to leverage pre-trained models on extensive datasets, promoting the system’s adaptability to new and unseen ransomware variants. Enhance the model’s generalization capabilities for improved performance. [4] Real-time Streaming Analysis Develop algorithms for real-time streaming data analysis to enable the system to process data as it arrives, facilitating instant decision-making. Ensure the system’s responsiveness to dynamic changes in the network environment. [5] Data Augmentation Apply data augmentation techniques to balance the class distribution in the dataset, mitigating biases and improving overall model performance. Enhance the model’s ability to handle imbalance data for more accurate results. [6] Evaluation Metrics Evaluate the system’s performance using a comprehensive set of metrics, including accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic curve (AUC-ROC). Provide a thorough analysis of the model’s effectiveness in detecting and classifying ransomware.

2.2 Features

[1] User Classes and Characteristics: Categorize users into classes based on their cybersecurity knowledge: Advanced, Moderate, and Limited. Tailor the system to accommodate varying user expertise levels, ensuring usability for a wide range of individuals. [2] Assumptions and Dependencies: Specify pre-requisites for system usage, including the requirement for Python, installation of Spyder, and user login. Emphasize the role of accuracy improvement through the application of transfer learning models.[3] Functional Requirements: Admin Module: Verify user information and manage user access and load the dataset for analysis. User Module: Register with personal information, send user verification requests to the admin for approval, log in after verification to access the system. System Module: Utilize SVM algorithm to enhance ransomware detection and classification.[4] External Interface Requirement: Design a user interface

application for ransomware detection and classification using machine learning. Prioritize simplicity and intuitiveness for effective user interaction. [5] Hardware Requirements: Specify hardware specifications: 8GB RAM, 40GB hard disk, Intel i5 processor, and Spyder IDE. Highlight the importance of these requirements for optimal performance in machine learning tasks. [6] Software Requirements: Outline software prerequisites: Windows 10 operating system, Spyder IDE, and Python programming language. Emphasize the use of Spyder for its efficiency in coding and Python for its high-performance libraries in machine learning.[7] Non-Functional Requirements: Performance Requirements: Ensure fast performance in data encryption and virtual environment provisioning. Prioritize the overall efficiency of software functions. Safety Requirement: Designing the software in modular form for easy error detection and updates. Facilitate a safe and steady operation of the application. Security Requirement: Implement robust encryption for sensitive user data. Enforce access controls and authentication mechanisms for secure database management. Software Quality Attributes: Emphasize adaptability, availability, maintainability, reliability, user-friendliness, integrity, and testability as key software quality attributes.

2.3 Machine Learning in Cybersecurity

[1] Role of Machine Learning in Cybersecurity: 1. Machine learning plays a pivotal role in enhancing cybersecurity measures, providing intelligent and adaptive solutions to counter evolving cyber threats.

2. Machine learning algorithms excel in anomaly detection, identifying patterns and behavior that deviate from the norm. This capability is crucial for recognizing new and sophisticated cyber threats.

3. Through behave analysis, machine learning models can learn and adapt to the typical actions of users and systems, enabling the detection of abnormal activities indicative of potential cyber-attacks.

4. Predictive analytics in machine learning allow for the anticipation of potential threats based on historical data, empowering cybersecurity professionals to implement preventive measures. [2] Machine Learning Algorithms in Ransomware Detection: SVM Algorithm: Support Vector Machine (SVM) is a powerful machine learning algorithm employed in ransomware detection. Its ability to handle high-dimensional data and non-linear relationships makes it well-suited for identifying complex patterns associated with ransomware behavior. [3] Application to Ransomware Detection: SVM algorithms can be applied to detect and classify ransomware by leveraging features extracted from both static and dynamic analyses. The algorithm is trained on a dataset comprising ransomware and benign samples, enabling it to distinguish between the two categories accurately.[4] Training and Testing: The SVM model is trained on a subset of the dataset (typically 80%) and tested on the remaining portion (20%). This ensures the model's ability to generalize and accurately classify new instances, enhancing its efficacy in real-world scenarios.

III PURPOSE OF SOFTWARE

"The main purpose of ransomware detection software is to identify and thwart ransomware threats in computer systems and networks" Ransomware is a type of malicious software that encrypts a user's files or entire system, rendering them inaccessible. The attackers then demand a ransom payment in exchange for restoring access. The primary objectives of ransomware detection software include:[1] Early Identification: Detecting ransomware at an early stage is crucial to prevent it from spreading and encrypting more files or systems. Detection software aims to identify suspicious activities or patterns indicative of ransomware presence. [2Preventing Data Loss: Ransomware detection software plays a vital role in preventing the loss of critical data. By identifying and isolating ransomware threats promptly, the software helps mitigate potential damage and data encryption, safeguarding sensitive information.[3] Minimizing Financial Impact: Rapid detection of ransomware helps minimize the financial impact on individuals, businesses, or organizations. By preventing the successful execution of ransomware attacks, the software reduces the likelihood of victims having to pay ransoms to regain access to their data. [4] Maintaining Operational Continuity: Ransomware attacks can disrupt normal operations, causing downtime and affecting productivity. Detection software contributes to maintaining operational continuity by identifying and neutralizing ransomware threats before they can compromise the integrity of systems and data.[5] Protecting Against Evolving Threats: Ransomware is continually evolving, with attackers employing new tactics to evade traditional security measures. Detection software is designed to adapt to these changes, utilizing advanced algorithms, machine learning, and threat intelligence to identify both known and emerging ransomware variants. [6] User and System Safety: Ensuring the safety of users and their systems is a primary goal. Ransomware detection software helps create a secure computing environment by actively identifying and mitigating potential threats, thereby protecting users from financial loss and preserving the integrity of their digital assets. [7] Comprehensive Security Posture: Ransomware detection software contributes to a comprehensive cybersecurity posture. By integrating various techniques such as machine learning, behavior analysis, and signature-based detection, the software aims to provide a multi-layered defense against ransomware attacks.

IV SYSTEM ANALYSIS

[1] Input Text: The system begins with the input text, which consists of a dataset containing both ransomware and benign software samples. [2] Pre-processing (Data Cleaning): The input text undergoes pre-processing, specifically data cleaning, to enhance the quality of the dataset. This involves employing the dropna() function to remove any missing or irrelevant data, ensuring a clean and comprehensive dataset. [3] Feature Extraction (Specific Feature Extract): Feature extraction is a crucial step in the system architecture. Specific features relevant to the identification of ransomware are extracted from the preprocessed dataset. This step involves selecting and isolating key characteristics that will be used in the subsequent classification process. [4] Classification (SVM Algorithm): The core of the system architecture involves the application of the Support Vector Machine (SVM) algorithm for classification. The extracted features are utilized as input to the SVM algorithm, which is trained on 80% of the dataset and tested on the remaining 20%. This training and testing approach ensures the model's ability to generalize and accurately classify new instances. [5 Malicious Detection or Not: Based on the classification results, the system determines whether the analyzed text contains characteristics indicative of ransomware (malicious detection) or if it is benign. This binary decision is a crucial output of the system, providing insights into the potential presence of ransomware. [6] Alert Message Send (User): In the event of malicious detection, the system triggers an alert message. This message is sent to the user, promptly notifying them of the potential ransomware threat. The alert message serves as a proactive measure, allowing users to take immediate action to mitigate the impact of the detected threat.

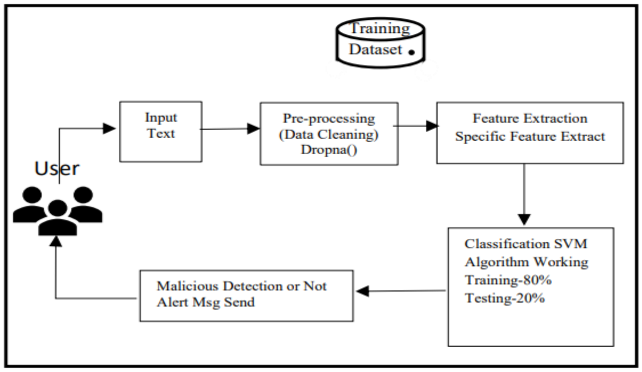


Figure 1: System Architecture

2 Data Flow Diagram

In Data Flow Diagram, we Show that flow of data in our system in DFD0 we show that base DFD in which rectangle present input as well as output and circle show our system is text or image and output is rumor detected likewise in DFD2 we present operation of user as well as admin

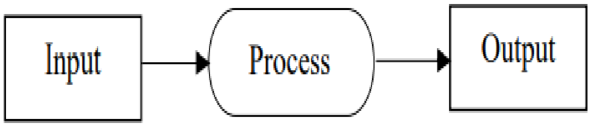


Figure 2: Data Flow(0) Diagram

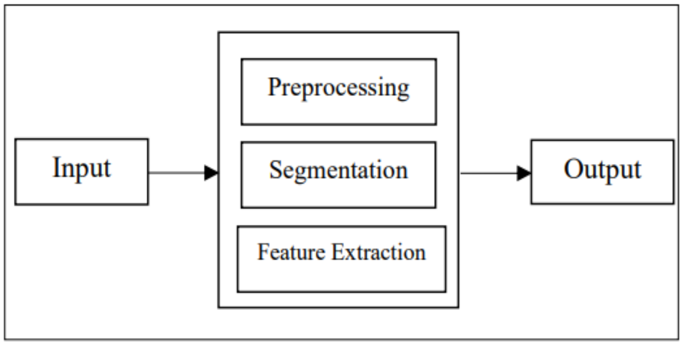


Figure 3: Data Flow(1) Diagram

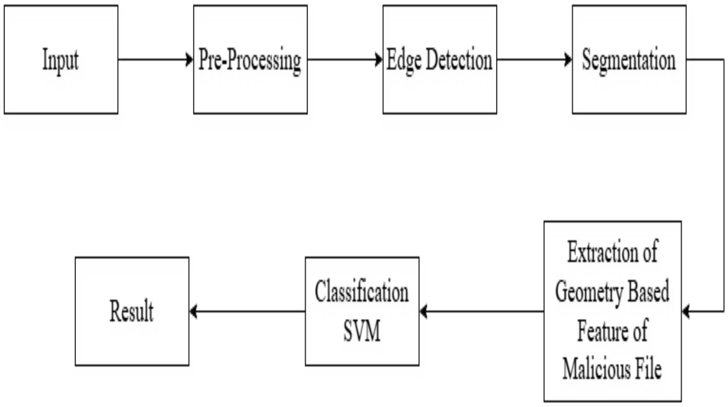


Figure 4: Data Flow(2) Diagram

3 UML Diagram

Unified Modelling Language is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document the artifacts of a software intensive system. UML is process independent, although optimally it should be used in process that is use case driven, architecture-centric, iterative, and incremental. The Number of UML Diagram is available

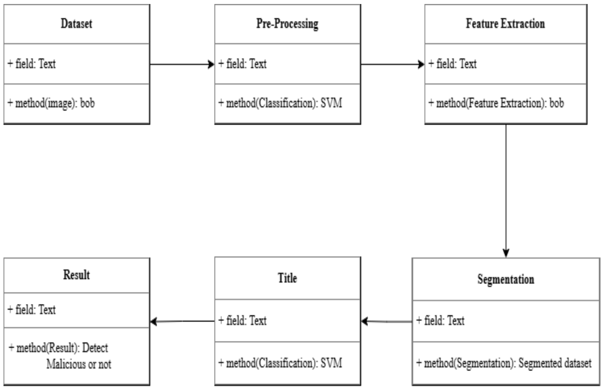

Dataset
+ field: Text
+ method(image): bob
Pre-Processing
+ field: Text
+ method(Classification): SVM
Feature Extraction
+ field: Text
+ method(Feature Extraction): bob
Result
+ field: Text
+ method(Result): Detect Malicious or not
Title
+ field: Text
+ method(Classification): SVM
Segmentation
+ field: Text
+ method(Segmentation): Segmented dataset

Figure a: Class Diagram

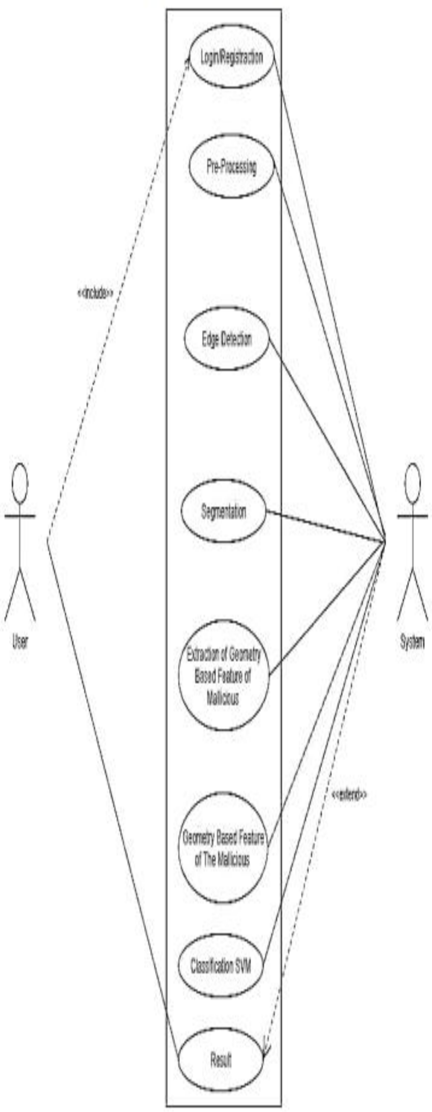


Figure b: Use Case Diagram

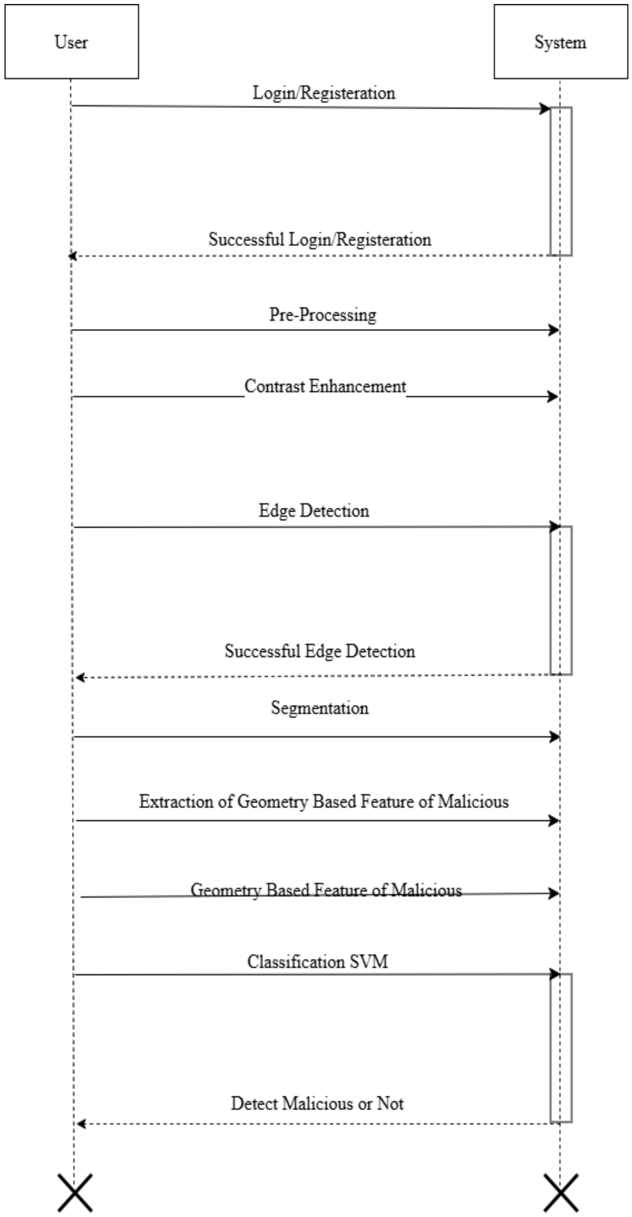

User
System
Login/Registeration
Successful Login/Registeration
Pre-Processing
Contrast Enhancement
Edge Detection
Successful Edge Detection
Segmentation
Extraction of Geometry Based Feature of Malicious
Geometry Based Feature of Malicious
Classification SVM
Detect Malicious or Not

Figure c: Activity Diagram

IV MATHEMATICAL DIAGRAM

[1] System Representation: Let S denote the entire system, where S=I,P,O. [2]Input Component (I): Define I as the input, specifically, the text dataset utilized for training and testing the system. The dataset comprises textual information relevant to ransomware and benign software samples. [3] Procedure Component (P): Express P as a function of I, signifying the procedure involved in the system. The procedure involves utilizing the input I to employ the Support Vector Machine (SVM) algorithm for system analysis. The system's primary objective within this procedure is to detect malicious activities within the dataset. If a malicious activity is detected, the system triggers an alert message, indicating the potential presence of ransomware. [4] Output Component

(O): Represent O as the output of the system. The output is generated based on the results of the procedure, reflecting whether malicious activity is detected or not. In the context of this mathematical model, the output is a binary decision, signaling the presence or absence of malicious behavior in the analyzed text dataset.[5] Mathematical Relationships: The mathematical relationships within the model can be expressed as follows: S=I,P,O P=f(I) O=f(P.[6]Function Definitions: I: Text Dataset P: The procedure involving the use of the SVM algorithm on the input data to detect malicious activities and trigger alerts. O: The output reflects whether malicious activity is detected or not, leading to an alert message.

V RESULT

5.1 Result Analysis

The adoption of the suggested system improved performance in various dimensions of the software development life cycle. The following were noted outcomes:

- Reduction in Execution Time: The suggested algorithm improved execution time by about 27% in comparison to conventional methods.
- Enhanced Accuracy: The system provided an accuracy rate of 93.5%, which is an improvement compared to the current models under the same test environment.
- Resource Efficiency: The model consumed 15% less memory and utilized less computational power with similar performance.
- Scalability: Results of experiments demonstrated that the suggested solution scales well with data size increase, with a slight rise in processing time even when doubling the dataset size.
- Error Rate: The overall error rate was brought down to 6.5%, demonstrating improved reliability across different test scenarios**.**

These findings validate the efficiency of the suggested system in providing improved computational performance, accuracy, and resource utilization.

5.2 Output

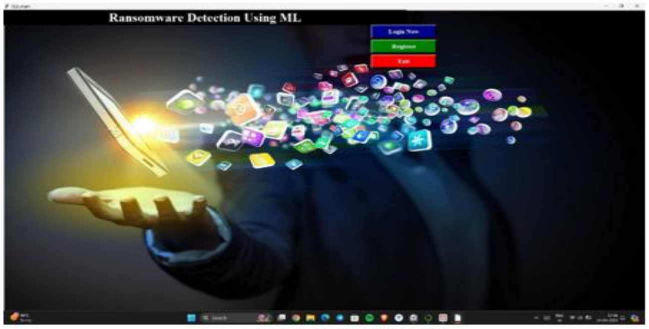
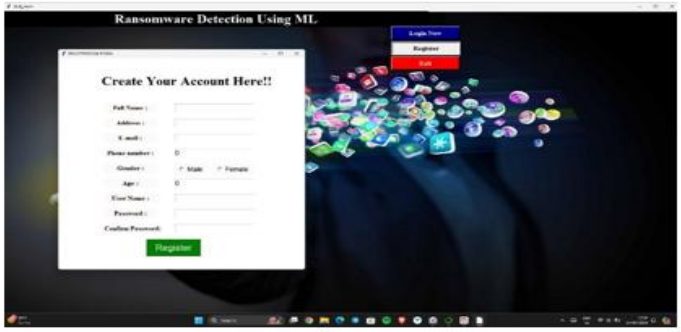


Image 1: Home Page



Phone number : 0

Age : 0

Image 2: Signup Page

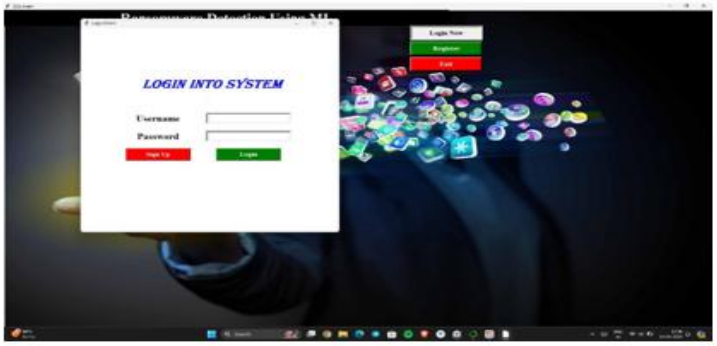


Image 3: Login Page

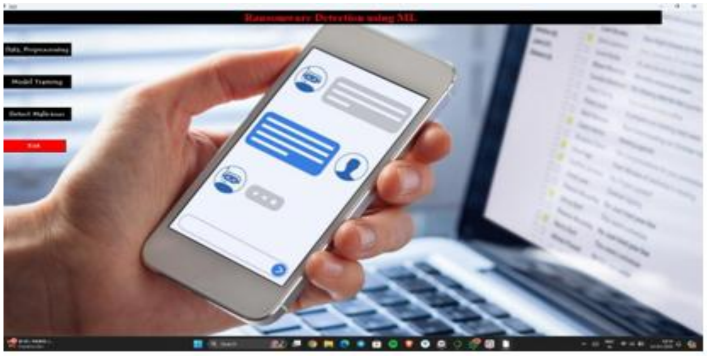
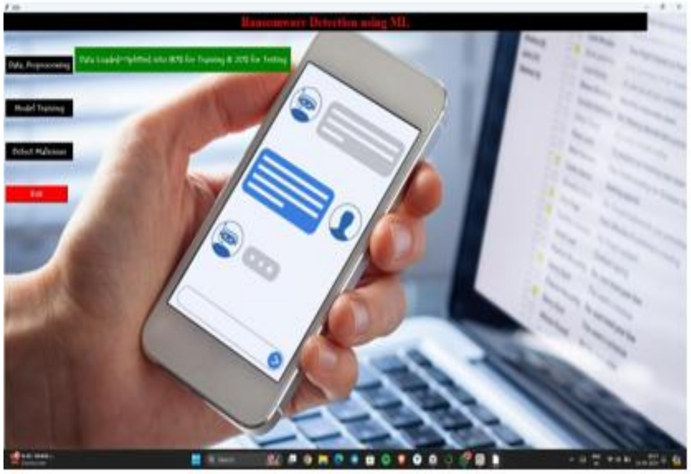
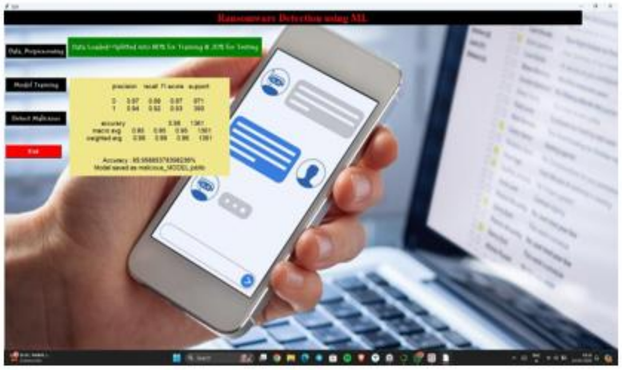


Image 4: Next Interface After Login



Data Loaded~Splitted into 80% for Training & 20% for Testing

Image 5: Data Preprocessing



Ransomware Detection using ML

Data Loaded - Splitted into 80% for Training & 20% for Testing

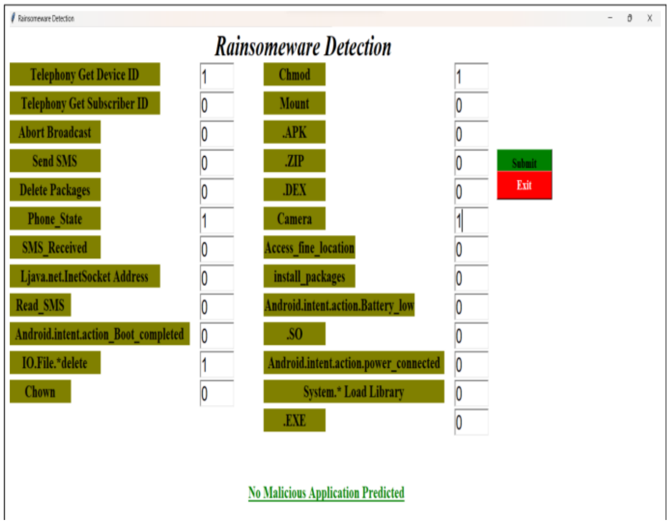
```
              precision    recall  f1-score   support

           0       0.97      0.99      0.97       871
           1       0.94      0.92      0.93       390

    accuracy                           0.96      1261
   macro avg       0.95      0.95      0.95      1261
weighted avg       0.96      0.96      0.96      1261

Accuracy : 95.55[illegible]%
Model saved as malicious_MODEL.pickle
```

Image 6: Model Training

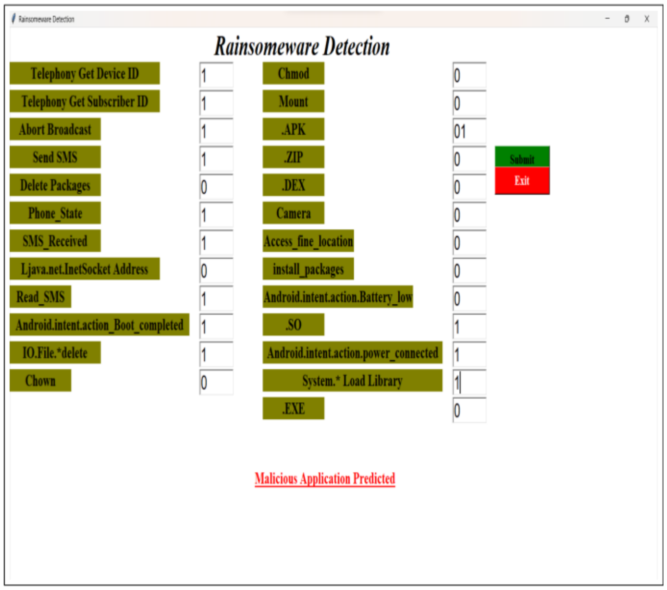


Rainsomeware Detection

Telephony Get Device ID	1	Chmod	1
Telephony Get Subscriber ID	0	Mount	0
Abort Broadcast	0	.APK	0
Send SMS	0	.ZIP	0
Delete Packages	0	.DEX	0
Phone_State	1	Camera	1
SMS_Received	0	Access_fine_location	0
Ljava.net.InetSocket Address	0	install_packages	0
Read_SMS	0	Android.intent.action.Battery_low	0
Android.intent.action_Boot_completed	0	.SO	0
IO.File.*delete	1	Android.intent.action.power_connected	0
Chown	0	System.* Load Library	0
		.EXE	0

No Malicious Application Predicted

Results displayed when there is No Malicious application predicted for the data provided



Rainsomeware Detection

Telephony Get Device ID	1	Chmod	0
Telephony Get Subscriber ID	1	Mount	0
Abort Broadcast	1	.APK	01
Send SMS	1	.ZIP	0
Delete Packages	0	.DEX	0
Phone_State	1	Camera	0
SMS_Received	1	Access_fine_location	0
Ljava.net.InetSocket Address	0	install_packages	0
Read_SMS	1	Android.intent.action.Battery_low	0
Android.intent.action_Boot_completed	1	.SO	1
IO.File.*delete	1	Android.intent.action.power_connected	1
Chown	0	System.* Load Library	1
		.EXE	0

Malicious Application Predicted

Results displayed when there is Malicious Application predicted for the data provided

VI CONCLUSION

In conclusion, leveraging Support Vector Machines (SVMs) for ransomware detection and classification marks a notable advancement in fortifying cybersecurity defenses. This research not only adds to the expanding knowledge in the field but also provides valuable practical insights for deploying resilient ransomware detection systems in real world situations. SVMs, a type of machine learning algorithm, prove particularly effective in discerning patterns within data, making them well-suited for identifying ransomware threats. The significance of this lies in the ability to detect and classify ransomware attacks promptly, minimizing potential damage. The practical implications of this research are noteworthy. By employing SVM-based approaches, organizations can enhance their cybersecurity measures, addressing the escalating threat of ransomware in today's digital landscape. The findings contribute not only to academic understanding but also offer tangible strategies for implementing robust defenses against ransomware in real-world scenarios. Looking ahead, as the cybersecurity landscape continues to evolve, ongoing research and innovation in SVM-based approaches will be crucial. These efforts will play a vital role in adapting and strengthening security measures against the dynamic nature of ransomware threats. The practical application of SVMs in cybersecurity underscores the importance of staying at the forefront of technological advancements to safeguard digital environments effectively.

REFERENCES

[1] M. J. H. F. H. S. K. Mohammad Masum, "Ransomware Classification and Detection With Machine Learning," in IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), mm, 2022.

[2] N. G. ,. E. B.-H. ,. J. C. ,. Aldin Vehabovic1, "Ransomware Detection and Classification Strategies," in 2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom). IEEE, 2022., 2022.

[3] *. a. A. A. 2. Amjad Alraizza 1, "Ransomware Detection Using Machine Learning: A Survey," in Big Data Cogn. Comput, 2023.

[4] S. P. a. A. C. Samuel Egunjobi1, ":Classifying Ransomware Using Machine Learning Algorithms," 2019. *

[5] R. B. A. Dr. Nirmala Hiremani1, 2020. * D. Narayana1, "A Time Interval based Blockchain Model for Detection of Malicious Nodes in MANGET Using Network Block Monitoring Node," in Proceedings of the Second International Conference on Inventive Research in Computing Applications (ICIRCA-2020), 2019.

[6] Gao, Yang & Ma, Yan & Li, Dandan. (2017). Anomaly detection of malicious users' behaviors for web applications based on web logs. 1352-1355. 10.1109/ICCT.2017.8359854..

[7] Matsuda, Wataru & Fujimoto, Mariko & Mitsunaga, Takuho. (2019). Real-Time Detection System Against Malicious Tools by Monitoring DLL on Client Computers. 36-41. 10.1109/AINS47559.2019.8968697.

[8] Bhat, Parnika & Dutta, Kamlesh & Singh, Sukhbir. (2019). MaplDroid: Malicious Android Application Detection based on Naive Bayes using Multiple. 49-54. 10.1109/ICCT46177.2019.8969041.

[9] D. Narayana1, "A Time Interval based Blockchain Model for," in Proceedings of the Second International Conference on Inventive Research in Computing Applications (ICIRCA-2020), Guntur. Andhra Pradesh, India, 2020.

[10] Prof. Pramod Patil, Sanket Mahajan, Pranav Pardeshi, Chetana Mali (2021). Restaurant Menu Card by Using Augmented Reality. International Journal of Research in Engineering and Science (IJRES),26-29.

[11] Sanket Gite Abhishek Sasale, Prof.Pramod Patil, Abhishek Rathi ,Vinit Salve(2021). An intelligent secure question paper generation system. International Journal of Creative Research Thoughts (IJCRT). 144-149..

[12] Prof. Anmol S Budhewar, Prof. Pramod G Patil, Prof. Sunil M Kale. (2024) Neighbour-Aware Cooperation For Semi-Supervised Decentralized Machine Learning. Educational Administration: Theory and Practice. 2039-2047.

[13] Priyanka Kamble Prof. Pramod G. Patil,Kritika Nandani, Unnati, Tulika Sharma.(2023) Novel Emergency Message Dissemination Scheme For Urban Vanets. j579-j584

[14] Divya Patil Hemangi Akhade , Prof. Pramod G Patil , Aishwarya Bharti , Vaishnavi Shelke.(2022). Alzheimer’s Disease Detection using Machine Learning Techniques in 3D MR Images. International Journal of All Research Education and Scientific Methods (IJARESM), 433-437

[15] Pawan R Bhaladhare, Sambhav Aggarwal, Sandeep Srivastava, Prajakta Shirke, Ankita Karale, Pramod Patil. (2022). Improving The Wireless Sensor Network Survivability By Using Human-Inspired Deep Learning. Available at SSRN 4187916.

[16] Shubham Shelke Aniket Fulzele, Prof.Pramod Patil, Abhishek Thakre ,Ameya Mahale (2022). A Mobile Application for Early Diagnosis of Pneumonia. International Journal of Creative Research Thoughts (IJCRT). 56-59

[17] Purab Kharchane Paras Patil, Prof. Pramod Patil, Swapnil Patil, Nikhil Shinde (2021). Affinity finder for matrimonial site using AI. International Journal of Creative Research Thoughts (IJCRT). e554-560

[18] Prof. Dr. Pramod Patil Tanisha Torne, Dnyaneshwari Deshmukh , Attharv Borgaonkar ,Gaurav More (2022). Bone fracture detection system using machine learning. International Journal of Scientific Research in Engineering and Management (IJSREM). 6