IJIRIS: International Journal of Innovative Research in Information Security E-ISSN: 2349-7017
Volume 12, Issue 04, April 2026 P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2026.v1204.03

Dynamic Ransomware Detection Using Temporal API Call Analysis

Dr.B.Ranjitha
Assistant Professor, Department of CSE
Guru Nanak Institute of Technology, Hyderabad, Telangana, India
https://orcid.org/0009-0000-6299-7991
Merugu Ashwitha Reddy, Nancharla Sushmitha, Kamera Krupa Sagar
UG Students, Department of CSE,
Guru Nanak Institute of Technology, Hyderabad, Telangana, India

Publication History
Manuscript Reference No: IJIRIS/RS/Vol.12/Issue04/ISAP26.ISAP10082
Research Article | Open Access | Double-Blind Peer-Reviewed| ArticleID: IJIRAE/RS/Vol.12/Issue04/ISAP26.ISAP10082
Received:02,March 2026, Revised: 29, March 2026, Accepted: 10,April 2026,Published Online: 22,April 2026.
https://www.ijiris.com/volumes/Vol12/iss-04/03.ISAP26.ISAP10082.pdf
Article Citation: Dr.Ranjitha,Merugu,Nancharla,Kamera(2026), Dynamic Ransomware Detection Using Temporal API Call Analysis. IJIRIS: International Journal of Innovative Research in Information Security, Volume 12, Issue 04 of 2026 pages 272-279 **Doi:>** https://doi.org/10.26562/ijiris.2026.v1204.03 **BibTeX Key: Dr.Ranjitha@2026Dynamic**
IJIRIS papers should be cited as IJIRIS (International Journal of Innovative Research in Information Security, AM Publications, India 2026, ISSN 2349-7017, https://doi.org/10.26562/ijiris.2026.v1204.03 The journal's official abbreviation is IJIRIS. Orcid: https://orcid.org/0009-0004-9398-7488

Abstract: Ransomware has become one of the most hazardous cyber security threats, where sensitive user data are encrypted and ransom is required to be recovered. Conventional detection schemes which rely on static signatures are becoming less effective against contemporary ransomware which relies on obfuscation and polymorphic strategies. The project introduces a machine learning-driven dynamic ransomware identification system concentrating on analyzing the dynamic behavior of the machine by utilizing time-based API Call patterns. With the frequency of API calls, their order, and time, useful behavioral characteristics are identified and trained as a Random Forest classifier. The system, which is implemented with a Flask web interface, offers real-time detection and visual insights, with accuracy over 95% and allows proactive ransomware protection.

Keywords: Ransomware Detection, Dynamic Analysis, API Call Monitoring, Temporal Features, Machine Learning, Random Forest.

I. INTRODUCTION

Ransomware is currently among the most severe and fastest-evolving cybersecurity threats, as it attacks the computer systems by encrypting the important user data and requires ransom to decrypt it. The attacks have led to huge financial losses and data breaches of individuals, organizations and even government infrastructures in the recent years. Obfuscation and the frequent changes of the code prevent the detection of modern ransomware, which is growing more challenging to tackle by traditional security mechanisms based predominately on fixed signatures and heuristic rules. This project aims to overcome this challenge by concentrating on behavior based detection through the analysis of interaction of programs with the operating system as the programs are executed. When tracking API calls and analyzing their order, frequency, and time, the system will be able to detect ransomware specific behavioral patterns. The proposed system uses a Random Forest classifier that is trained on these time based features and is useful in correctly classifying benign and malicious programs. The solution is practical and effective as a Flask based web interface can offer real time detection results and intuitive visualizations to facilitate easy use.

II. LITERATURE SURVEY

Singh et al. (2023) suggested a dynamic ransomware detection model that is based on the API calls sequence analysis when executing a program. The method focuses on behavioral characteristics like the frequency of API calls, execution sequence, and time delays to differentiate between ransomware and clean software. To enhance efficiency and minimize redundancy, feature selection techniques, such as correlation analysis and PCA are used. Random Forest and Gradient Boosting classifiers have a high accuracy of more than 94% especially in the initial stages of pre-encryption. The framework, however, is based on sandbox environments and can be potentially finely tuned with regards to real time deployment. Park and Kim (2024) presented a time series informed ransomware detection method that concentrates on the time separation between subsequent API requests. The technique identifies behavioral patterns distinct to ransomware in the encryption stages by examining timing periods and call orderings. Classification is done with random Fore stand LSTM models, with an accuracy ofover95%.Thepaper draws the focus on better resistance against code obfuscation with the use of temporal profiling. Although the LSTM based model achieves good performance, it requires more computational resources that can be limiting in resource constrained setting. Lietal.(2024 introduced a dynamic ransomware detect or that uses machine learning and behavioral presented a time series informed ransomware detection method that concentrates on the time separation between subsequent API requests.

The technique identifies behavioral patterns distinct to ransomware in the encryption stages by examining timing periods and call orderings. Classification is done with random Forest and LSTM models, with an accuracy of over 95%. The paper draws the focus on better resistance against code obfuscation with the use of temporal profiling. Al though the LSTM based model achieves good performance, it requires more computational resources that can be limiting in resource constrained setting. Li et al. (2024) suggested a framework of ransomware behavior analysis through visualization inspired that presents API call sequences in graph based forms. The API calls are represented as nodes with edges that reflect logical and temporal connections and allow the analysts to visualize the suspicious patterns of execution. The model combines visual analytics and a Random Forest classifier to aid the automated detection process and improve interpretability. The strategy enhances openness and recognition of rogue ransomware forms, though, visual examination can be added overhead and necessitate professional input to be interpreted efficiently. Gupta and Lee (2022) introduced a dynamic ransomware detector that uses machine learning and behavioral analysis. The system dissects API traces, file operations, and changes in the registry to detect ransomware as it runs. Several classifiers are tested and the best of the mis the Random Forest that has the highest accuracy of 93. A log parsing system that is automated enhances scalability and manual effort. Although useful in early detection, the framework requires controlled execution environments and might be difficult with large volume real time system logs. Zhao and Chen(2023) suggested a hybrid ransomware detection model which is an integration of deep learning and conventional machine learning models. The API call logs are converted into temporal feature maps that record both sequence and timing information and processing with CNN to extract the features. Finally, a Random Forest classifier is used which enhances interpretability and strength. The system has a high accuracy of 97 percent, and low false positive rates. Nevertheless, the hybrid design makes the system more complex, and must be optimized with care to deploy in real time.

III. EXISTING SYSTEM

The available ransomware detection systems largely rely on signature based or heuristic techniques whereby they rely on predefined rules and known malware patterns. These systems check on suspicious programs and match the code with a database of known ransomware signature. When a match occurs, the program is labeled as malicious and otherwise it is considered to be safe. Al though this approach can be moderately efficient in the case of familiar ransomware families, new or modified attacks, as well as zero days, are challenging. Contemporary ransomware often employs obfuscation and encryption to conceal its actions, making signature updates low and detection slow, greatly augmenting the likelihood of successful infection.

A. Disadvantages of Existing Systems

- Fails to detect new or unknown ransomware are not present in the signature database.
- Uses purely the static analysis of the code, disregarding the dynamic behavior.
- Frequent signature updates, which raise maintenance effort.
- Simple by pass of the ransomware with obfuscation or encrypted payloads.

B. Proposed System

The suggested system uses a dynamic approach to ransomware detection based on machine learning to monitor the behavior of the running programs, instead of using fixed signatures. It constantly checks API calls that are performed by applications to the operating system with an emphasis on file access, encryption, and registry changes. The analysisof these API calls is determined by the sequence, frequency and timing in order to extract meaningful behavioral patterns. A random forest classifier is then employed to classify the benign and malicious behavior correctly. It also comprises of Flask based web interface, which enables users to upload logs and see intuitive visualizations, making it possible to detect ransomware quickly, reliably, and flexibly.

C. Advantages of the Proposed System

- Identifies ransomware dynamically than through the use of predefined code signatures.
- Able to detect new and unfamiliar types of ransom ware well.
- Minimizes false positives and false negatives with machine learning.
- Offers the real-time tracking and prevents encryption of data before it is encrypted.

IV. SYSTEM ARCHITECTURE

The suggested Dynamic Ransomware Detection architecture comprises of six main components interacting via secure and clearly defined interfaces to aid in real time ransomware detection and analysis. The design is scalable and modular, allowing behavioral evaluation to be efficiently evaluated with the help of machine learning. User Interface (UI):User Interface gives the primary interface between the users and the ransomware detection system. It enables uploading of executable files or API call logs, secure authentication and displaying detection results. This interface has controlled access and facilitates easy and user friendly launching of the ransomware analysis process. Flask Backend API: Flask Backend API serves as the hub of the system. It handles the requests of users, authentication, input validation and communication among various modules. The backend facilitates real time, smooth and secure data flow between the components of the system. Data Preprocessing Module: The module handles raw API call logs and eliminates noise and normalizes the timing information, and translates the sequences of executions to structured numerical data. Raw data is then pre-processed and stored in the database ready to be extracted and analyzed to obtain accurate features.

Detection & Analysis Layer: Detection and Analysis Layer is the decision pipeline of the system. It combines processed data with the machine learning model to examine patterns of behavior at runtime and identify possible ransomware actions.

IJIRIS: International Journal of Innovative Research in Information Security E-ISSN: 2349-7017
Volume 12, Issue 04, April 2026 P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2026.v1204.03

Machine Learning Model (Random Forest): The system uses a Random Forest classifier to differentiate between normal and malicious processes. It examines the characteristics including the frequency of API calls, execution sequence, and time intervals. The ensemble learning method enhances the detection accuracy, robustness, and the adaptation to novel ransomware variants. Visualization & Final Output Module: This module displays the results of detection in transparent charts and visual analytics. It assists users to understand classification results with ease by showing them whether the process under analysis is benign or ransomware as well as behavioral insights.

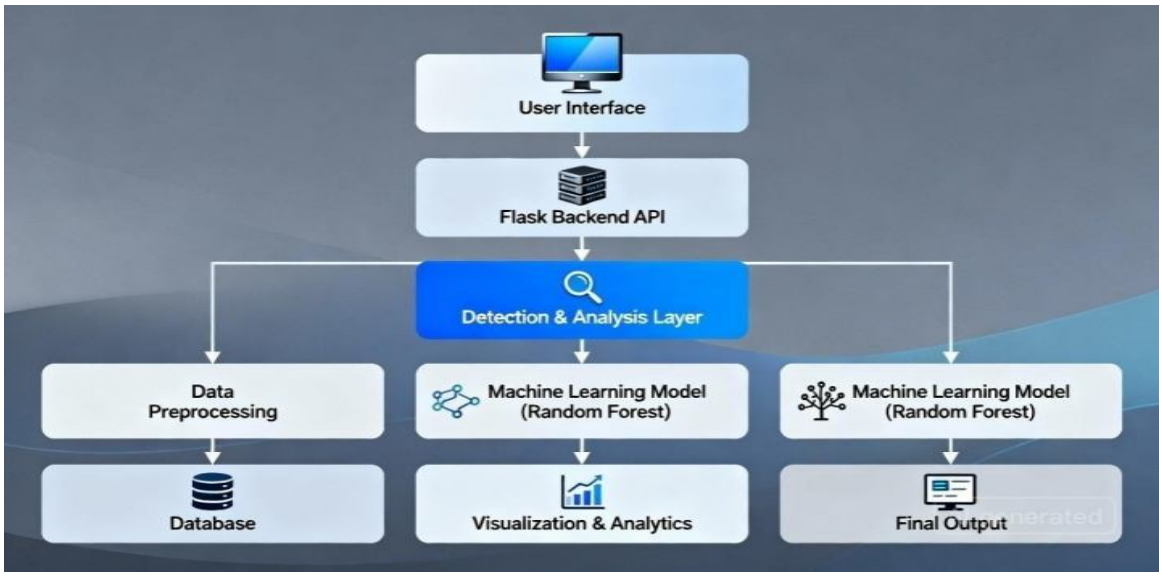

Fig. 1. This system takes uploaded files as input and uses a Random Forest model by analyzing the behavior of API calls based on time to identify the files, and visualizes the results to allow accurate and real-time ransomware detection.

A. Methodology

Algorithm: Dynamic Ransomware Detection Using Temporal API Call Analysis

Step 1. When the user is successfully authenticated, he/she uploads an executable file or API call log via the Flask based web interface.

Step 2. The backend server authenticates the input uploaded and routes it to the dynamic analysis module to be executed in a secure manner or inspected by the logs.

Step 3. The dynamic analysis module logs the program execution in a controlled environment and traces sequences of API calls and their timestamps.

Step 4. Logs of API calls are captured and sent to the preprocessing module, where noise can be filtered out, timestamps can be normalized, and raw sequences can be transformed into structured numerical data.

Step 5. The process of feature extraction yields time based behavioral features, including frequency of API calls, sequence of calls and wait time between calls.

Step 6. Detection is done by giving the resulting feature vector to the Random Forest classifier that compares it with the predicted behavioral pattern.

Step 7. The Random Forest model will consider the features and either declare the program as Benign or Ransomware depending on the learned temporal patterns.

Step 8. The result of the classification, including the confidence scores, is safely stored in the database to track the results and analyze them in the future.

Step 9. The visualization module creates the graphical illustrations of behavior and detection results of API, i.e. charts and metrics.

Step 10. The end result and visual outputs are presented to the user via the web interface.

To enhance the efficiency of detection, the system uses temporal feature aggregation on the sequence of API calls. With an execution log L and a timeline T, inter call intervals are calculated as $\Delta ti = T i - T i k$. Abnormal or non-regular time series patterns of feature vectors are highlighted, which allows detecting ransomware behavior early and minimizes false positives

B. Module Names

- User Interface(Frontend)
- Backend(Flask Framework)
- Dynamic Analysis Module
- Machine Learning Model
- Visualization & Output Layer

1.User Interface (Frontend): The user interface is the main interface between the users and the ransomware detection system.

IJIRIS: International Journal of Innovative Research in Information Security E-ISSN: 2349-7017
Volume 12, Issue 04, April 2026 P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2026.v1204.03

It is created in the form of HTML, CSS and JavaScript to assure of a clean, responsive, and aesthetically pleasing design. Using this interface, users can conveniently upload executable files, or API call logs and see the detection results immediately. The interactive design facilitates easy navigation and easy usability features so that even untrained users and cybersecurity experts can use the system without the need to possess technical know-how.

2.Backend (Flask Framework): The system has a backend that is based on the Flask framework and is in charge of application logic and data flow management. It safely processes file uploads, checks input data and preprocesses before analysis. Flask is effective in directing user requests to the dynamic analysis and machine learning modules. to ensure smooth communication between modules. Also, the backend deals with the production of results and the coordination of data storage and retrieval to ensure stable and reliable system performance.

3.Dynamic Analysis Module: The dynamic analysis module is used to track the behavior of a program by running the files uploaded to the system in a controlled sandbox. It logs API call sequences, or interactions between the program and the operating system, during execution.

4.Machine Learning Model: The machine learning model is important in categorizing programs into benign and ransomware. It examines behavioral characteristics based on API call sequences, with an algorithm like Random Forest and Gradient Boosting which is applicable in exploiting non linear and intricate data patterns. The model is trained with dynamical execution data and is evaluated with performance metrics such as accuracy, the true positive rate, and false positive rate to guarantee the consistency and reliability of the detection.

5.Visualization & Output Layer: Visualization and output layer displays the results of detection in a clear, easy to comprehend format. It presents the results of classification and levels of confidence using visual means like charts, progress indicators and alert messages. This module can assist in real time analysis, reporting, and informed decision making in cybersecurity monitoring by visually displaying API call behavior and detection status.

V. IMPLEMENTATION

The ransomware detection system proposed is deployed with a modular scalable architecture that distinctly separates the machine learning back-end with the user friendly web interface. The system is dynamically used to track the behavior of API calls when executing a program and categorises applications as benign or ransomware in real time. The logs of the API calls are gathered, purged, and converted into structured data, and behavioural attributes (frequency, sequence, and timing intervals) are mined. A Random Forest classifier is trained to use labelled datasets and embedded into the backend to allow real time predictions to be made correctly.

Algorithm Used:

Existing Algorithm:

Current ransomware detection method uses primarily heuristic rules and matching of the static signature. It matchestheprogramcodewithadatabaseofknownmalwaresignaturestodetectthreats.Althoughitworkswell with ransomware that have been identified before, it is notable to analyse runtime behaviour.

Proposed Algorithm:

The algorithm suggested is a dynamic detection method based on machine learning that examines time-based API call patterns when running a program. Random Forest classifier is used to extract and process behavioral characteristics.

VI. EXPERIMENTAL RESULTS

The system is tested with dynamic execution datasets that have benign and ransomware samples. It is more accurate in detecting malicious behavior by analyzing API calls over time, displays the results on multiple pages of the interface and performance metrics, which is more accurate and can be detected faster than traditional ones.

Fig 1: Home Page

The home page serves as the entry point of the system, providing navigation options for registration and login while presenting an overview of the ransomware detection platform.

IJIRIS: International Journal of Innovative Research in Information Security
Volume 12, Issue 04, April 2026
E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2026.v1204.03

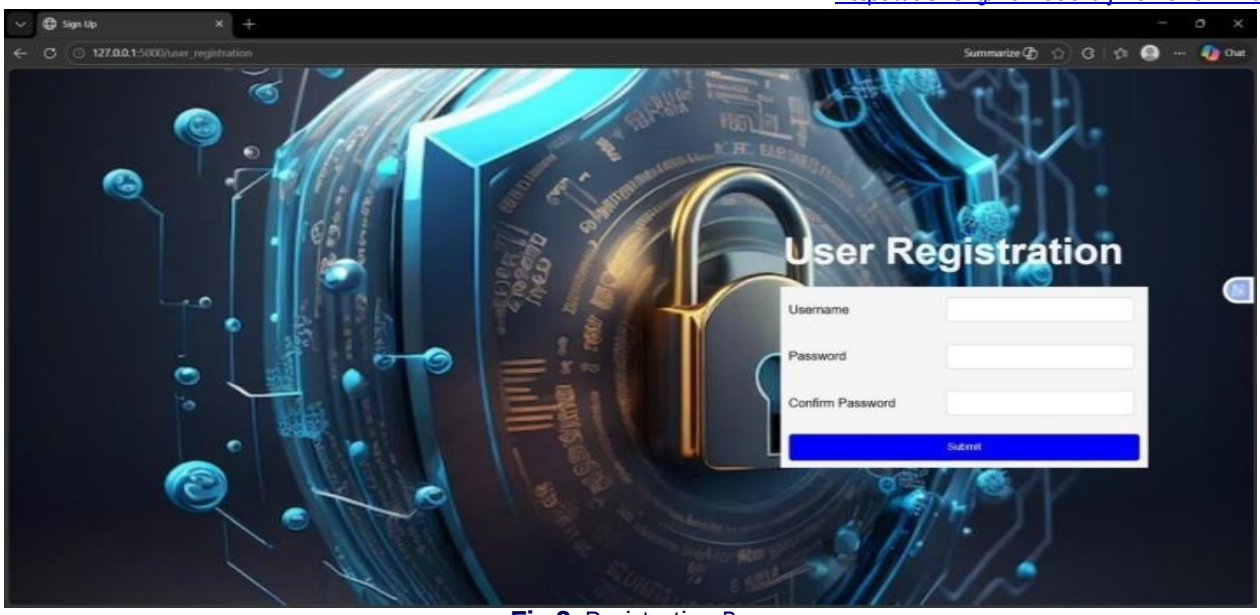

Fig 2: Registration Page

The registration page allows new users to create an account by entering basic credentials, ensuring authorized access and securely storing user details for future authentication.

Fig 3: Login Page

The login page enables registered users to securely authenticate using their credentials and access the ransomware detection dashboard for further analysis and operations.

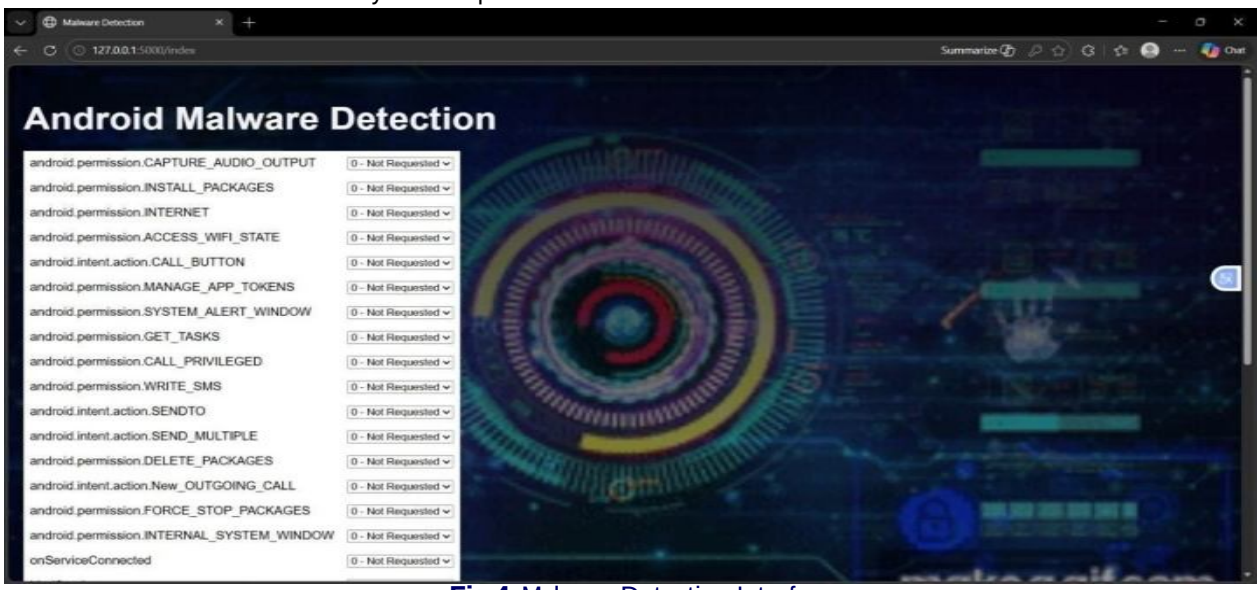

Fig 4: Malware Detection Interface

IJIRIS: International Journal of Innovative Research in Information Security E-ISSN: 2349-7017
Volume 12, Issue 04, April 2026 P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2026.v1204.03

This figure illustrates the malware detection interface where the users interface the analysis module. It shows API call features and parameters of malicious behavior in which the user can easily initiate detection and analyse application behavior.

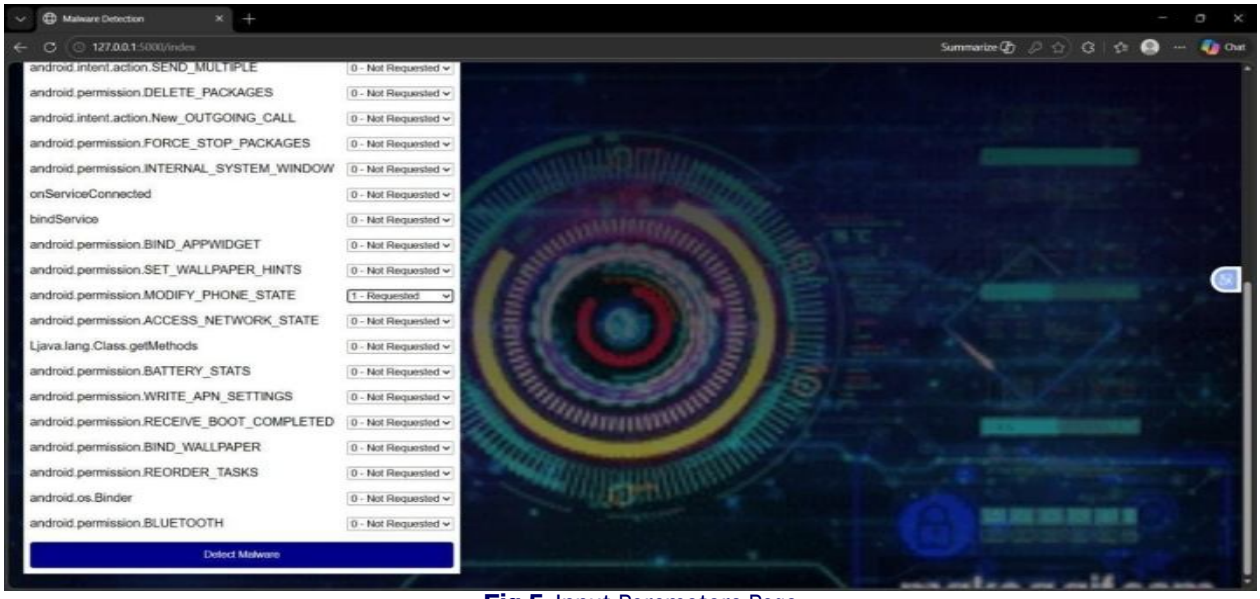

Fig 5: Input Parameters Page

The detection page allows users to upload executable files or API call logs, initiate analysis, and provide required input parameters for ransomware behavior evaluation.

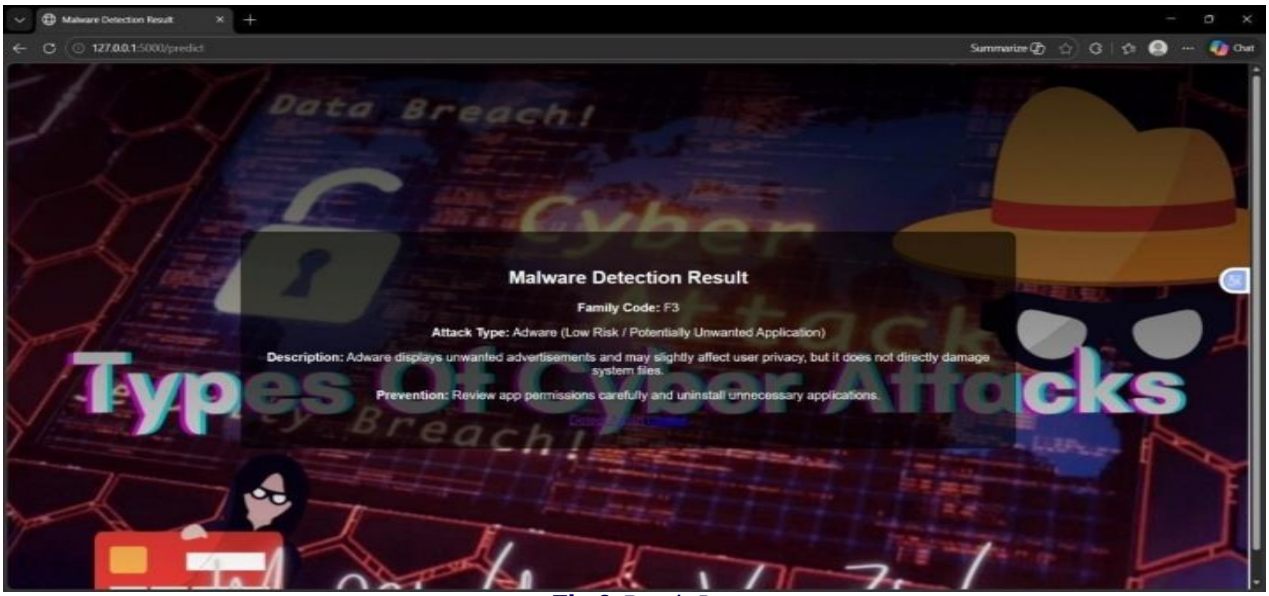

Fig 6: Result Page

The result page displays the final classification output, showing whether the application is benign or ransomware, along with confidence levels and visual insights for easy interpretation.

VII. CONCLUSION

The proposed Dynamic Ransomware Detection System is a logical and useful application of machine learning and easy to use web interface that can detect ransomware activity in real time. The system identifies malicious processes by correctly classifying the behavioral features of API calls, frequency and timing, through the use of a Random Forest classifier. Its interactive visualizations and modular Flask based design guarantee transparency of the results, early threat detection, low levels of data loss, and high levels of adaptability, which makes the solution reliable in both research and real world cybersecurity.

VIII. FUTURE ENHANCEMENT

The existing system offers a powerful and smart way to identify ransomware based on machine learning and time based API analysis. Their effectiveness can be enhanced in the future through such features as realtime monitoring of API, in corporating improved deep learning models such as LSTM or CNN, and automated alerts and detailed reporting. These enhancements would add accuracy, scalability, and practicality in real world cyber security settings.

REFERENCES

1. Zhang,S.,Wu,J.,Zhang,M.,& Yang,W.(2024).Dynamic malware analysis based on API sequence semantic fusion. School of Cyber Science and Engineering, Southeast University,Nanjing,China.
2. Wang,S.Xu,J.,Dong,F.,Wang,H.,&Yang,H.(2024).CanCal:Towards real-time and lightweight ransomware detection and response in industrial environments. arXiv preprint arXiv:2408.16515v1 [cs.CR].
3. Alhaidari,F.,AbuShaib,N.,Alsafi,M.,Alharbi,H.,Alawami,M.,Aljindan,R.,Rahman,A.,&Zagrouba,R.(2025).Ze Vigilante: Detecting zero-day malware using machine learning and sandboxing analysis techniques. Journal of Cyber Security and Digital Forensics.
4. Aljabri, M. (2024). Ransomware detection based on machine learning using memory dumps to detect ransomware with high accuracy. Computers & Security, Elsevier.
5. Cen,M.(2024). A zero-day ransomware early detection method based on zero-shot learning (ZRS). Computers & Security,Elsevier.
6. Chew,C.J.W.,Kumar,V.,Patros,P.,&Malik,R.(2024).Real-time system call-based ransomware detection. International Journal of Information Security, Springer.
7. Alhuwayshil,S.,Alrashed,A.,& Khan,M.(2025).Enhancing ransomware threat detection: Risk-aware monitoring of functional API calls. MDPI Cybersecurity.
8. Gulmez,S.,& Erdem, E.(2024). XRan: Explainable deep learning-based ransomware detection. Computers & Security,Elsevier.
9. Ahmed, R., & Lee,Y. (2023). Dynamic analysis and machine learning-based detection of ransomware. Journal of Information Security and Digital Forensics.
10. Zhao,L., & Chen,H.(2023). Hybrid deep learning model for real-time ransomware detection using API monitoring.Neural Computing and Applications, Springer.
11. Kaur,A., & Joshi,N.(2024).Intelligent ransomware detection using ensemble machine learning models. MDPI Electronics.
12. Ahmed,M.,& Rahman,T.(2023).Dynamic ransomware detection framework using API call dependency graphs. IEEE Transactions on Information Forensics and Security.
13. Wang, L., & Lin, H.(2022).Comparative analysis of static and dynamic detection techniques for ransomware identification. IEEE Security & Privacy.
14. Singh,S.,Kumar,R., & Sharma, M. (2023). Ransomware detection using machine learning on dynamic API call sequences. IEEE Access.
15. Park,S.,& Kim,H.(2024).Behavioral profiling of ransomware via API call time series analysis. Computers & Security, Elsevier.
16. Li,X.,Chen,J., & Zhang,Y.(2024).Visualization framework for malware and ransomware behavior analysis. ACM Transactions on Privacy and Security.
17. Gupta, R.,& Lee,Y.(2022).Behavior-based machine learning approach for detecting ransomware attacks. Journal of Network and Computer Applications, Elsevier.
18. Zhao, Q.,&Huang,Z.(2024).Real-time detection of obfuscated ransomware using graph neural networks. IEEE Transactions on Dependable and Secure Computing.
19. Li,H.,Wang,J.,& Yang,F.(2023). A multi-layered machine learning approach for dynamic ransomware detection. Springer Journal of Cybersecurity Research.
20. Luo,C.,Zhang,Y.,& Xu,H.(2024). Detecting unknown ransomware variants through API call correlation analysis. IEEE Access.
21. Tang,W.,& Zhou,K.(2023).Behavioral analysis-based zero-day ransomware detection using Random Forest. MDPI Applied Sciences.
22. Joshi,Y.,Totad,S.G.,Geeta,R.B.,& Prasad Reddy,P.V.G.D.(2018). Mobile agent-based frequent pattern mining for distributed databases. In S. Bhalla, V. Bhateja, A. Chandavale, A. Hiwale, & S. Satapathy (Eds.), Intelligent computing and information and communication (Vol. 673). Springer. https://doi.org/10.1007/978-981-10-7245-1_9
23. Bharamagoudar,G.R.,Totad,S.G.,Prasad Reddy,P.,&Shobha,R.B.(2015).Zealous leadership paradigms. International Journal of Globalisation and Small Business, 7(1), 92–106. https://doi.org/10.1504/IJGSB.2015.069033
24. Geeta,R.B.,Totad,S.G.,Prasad Reddy,P.,&Shobha,R.B.(2015).Big data structure and usage mining coalition. International Journal of Services Technology and Management, 21(4/5/6), 252–271. https://doi.org/10.1504/IJSTM.2015.073930