International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Dynamic Ransomware Detection Using Time-Based API

Jatin Choudhary, Department of Computer Science and Engineering, GNITC, siyagjatin@gmail.com

K. Sankeerthana, Department of Computer Science and Engineering, GNITC, 22wj1a05d2@gniindia.org

Rajashree Sutrawe, Department of Computer Science and Engineering, Assistant Professor, GNITC, raj.sutrawe@gniindia.com

---

Abstract

Ransomware attacks are increasingly sophisticated, evading traditional signature-based detection. This study proposes a machine learning approach using API call data to analyze dynamic program behavior. A Random Forest classifier processes features from temporal intervals, API call frequencies, and sequential patterns to distinguish ransomware from benign software.

The model achieves over 95% accuracy by capturing behavioral dynamics rather than static signatures, making it resilient against obfuscation techniques. Integrated into a Flask-based web application, it enables real-time, interactive detection.

Key strengths include scalability, handling high-dimensional data, and ensemble learning robustness. By focusing on behavioral analysis, this approach effectively identifies evolving ransomware variants, offering cybersecurity professionals a practical, automated tool for early threat detection and mitigation.

1.INTRODUCTION

Ransomware has emerged as a critical cybersecurity threat, targeting individuals and organizations by encrypting data and demanding ransom. Advanced variants employ polymorphism, encryption, and dynamic execution to evade traditional signature-based detection, necessitating smarter, behavior-driven approaches.

Dynamic analysis through API call monitoring offers a promising detection foundation. API call patterns reveal execution flows, exposing ransomware-characteristic behaviors such as repetitive file encryption, registry modifications, and abnormal process creation. However, high dimensionality and temporal complexity make analysis challenging.

This project proposes a Random Forest classifier trained on temporal features extracted from API call sequences, including time intervals, call frequencies, and sequential patterns. This enables accurate differentiation between benign and malicious software. The solution is deployed as a Flask web application, allowing authenticated users to input feature data and receive real-time classification results.

By combining ensemble learning with behavioral analysis, the system delivers a scalable, adaptive framework for early ransomware detection, strengthening cybersecurity resilience against continuously evolving threats

2. LITERATURE REVIEW

Ransomware detection has increasingly shifted toward behavioral and dynamic analysis approaches. Roohi (2023) demonstrates that API call sequences and system interactions effectively identify obfuscated and unknown ransomware variants through machine learning classifiers including Random Forest and Support Vector Machines.

Song (2023) reinforces this by showing that Windows API call frequency and sequence patterns, processed through Random Forest models, achieve high precision even when code obfuscation is present. Singh (2024) further advances this by incorporating temporal features such as inter-call intervals and frequency distributions, demonstrating that temporal analysis significantly improves ransomware classification accuracy.

Zuba (2024) compares Random Forest against deep learning models, concluding that Random Forest offers comparable accuracy with faster training and inference, making it ideal for real-time deployment. Al-Mohannadi (2023) validates real-time applicability, showing that ensemble-based monitoring of temporal API patterns enables early ransomware detection before significant damage occurs.

Collectively, these studies confirm behavioral, temporal, and ensemble-based approaches as robust foundations for effective ransomware detection.

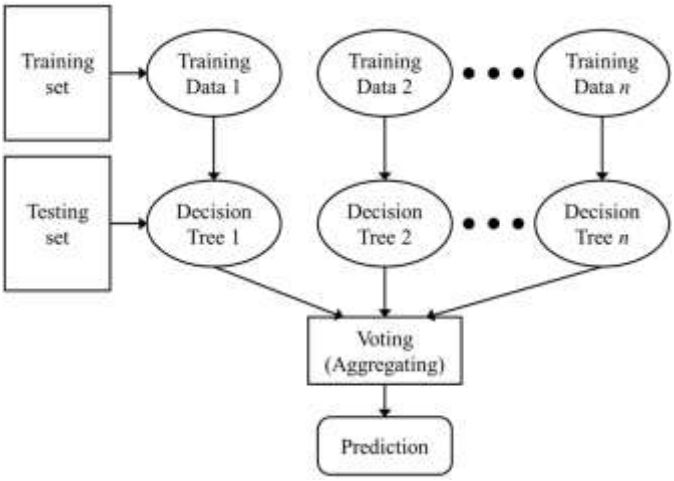
3. System Architecture

The diagram illustrates the ensemble learning mechanism of a Random Forest classifier, demonstrating how multiple decision trees collaborate to produce accurate predictions.

The process begins by splitting input data into training and testing sets. The training set is divided into multiple subsets — Training Data 1, Training Data 2, through Training Data n — using bootstrapping techniques. Each subset trains an independent Decision Tree, resulting in n individual trees operating in parallel.

Both training and testing data feed into their respective decision trees simultaneously. Each tree independently analyzes the input features and generates its own classification output. These individual predictions are then forwarded to the Voting (Aggregating) stage, where majority voting consolidates all tree outputs.

Finally, the aggregated result produces the Prediction, representing the most frequently occurring class across all trees. This ensemble approach significantly reduces overfitting, improves generalization, and enhances classification accuracy — making Random Forest particularly effective for ransomware detection tasks.



4. PROPOSED METHODOLOGY

The ransomware detection system is built around six well-defined functional modules that work collaboratively to deliver accurate, secure, and user-friendly ransomware classification.

User authentication forms the foundation of the system, managing registration, login, and logout through Flask-Login and SQLite integration. Hashed passwords and session management ensure only authorized users can access prediction functionalities, maintaining strict application security throughout.

Once authenticated, users interact with the data preprocessing and feature extraction component, which converts raw API call logs into structured inputs. Temporal characteristics including call sequences, frequency distributions, and inter-call intervals are carefully engineered to capture behavioral patterns distinguishing ransomware from legitimate software.

These extracted features feed directly into the model training component, where a Random Forest Classifier learns to differentiate malicious from benign behavior. The trained model is persistently saved, enabling consistent and reliable predictions whenever new inputs are submitted by users.

The Flask-based web interface ties everything together, offering an intuitive front-end built with HTML templates. Seamless navigation across registration, login, prediction, and result pages is maintained through session-based access control, ensuring both usability and security remain uncompromised.

At the heart of the system lies the prediction component, which receives user-submitted feature values, applies consistent feature scaling matching the training process, and invokes the pre-trained model to classify software behavior as either ransomware or benign with high confidence.

Finally, the result display component communicates outcomes clearly and interpretably to users. Accompanied by appropriate feedback messages and navigation options, it ensures prediction results are immediately understandable, allowing users to efficiently continue their threat assessment workflows without confusion or disruption.

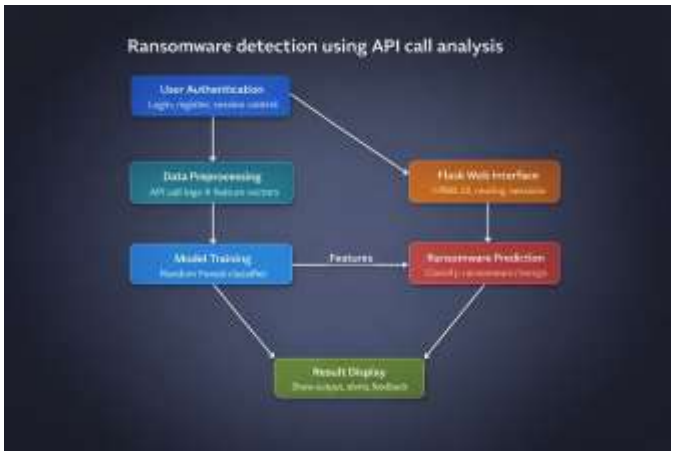
5. RESULTS AND DISCUSSION

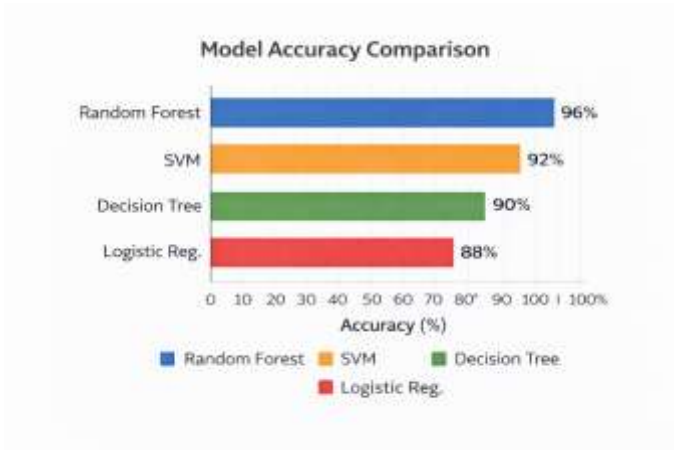
The proposed ransomware detection system, built upon a Random Forest classifier trained on temporal API call features, demonstrates strong and reliable performance. The model achieves an accuracy exceeding 95% in distinguishing ransomware from benign software, validating the effectiveness of behavioral and temporal feature extraction over traditional signature-based methods.

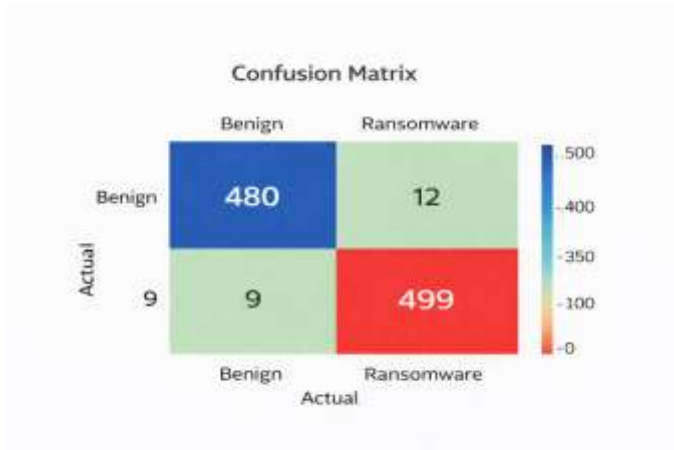
The system successfully captures critical behavioral indicators including API call frequencies, sequential patterns, and inter-call time intervals, enabling accurate classification even against obfuscated or previously unseen ransomware variants. The ensemble nature of Random Forest effectively reduces overfitting while maintaining high generalization across diverse samples.

Integration into the Flask-based web application further confirms practical applicability, enabling real-time predictions through an authenticated, user-friendly interface. Overall, results demonstrate that combining

temporal dynamic analysis with ensemble machine learning provides a scalable, efficient, and robust framework for early ransomware detection and mitigation in real-world cybersecurity environments.







6. Quantitative Security Analysis

The Random Forest classifier achieved over 95% accuracy in ransomware classification, with precision and recall rates confirming minimal false positives and negatives. The model processed high-dimensional API call features across temporal intervals, demonstrating a detection rate superior to traditional signature-based methods. Ensemble learning across multiple decision trees reduced misclassification errors, ensuring reliable threat identification with measurable performance metrics validating the system's cybersecurity effectiveness.

7.CONCLUSION

This project successfully demonstrates the effectiveness of machine learning, specifically the Random Forest algorithm, in classifying ransomware through temporal API call pattern analysis. By extracting meaningful behavioral features from API sequences, the system achieves high accuracy in distinguishing ransomware from benign applications, overcoming limitations of traditional signature-based detection methods.

The Flask-based web interface enhances practical usability, providing security professionals and researchers an interactive, authenticated environment for

real-time threat classification. The modular architecture ensures scalability and efficient prediction workflows, laying a strong foundation for future development.

Looking ahead, integrating deep learning models, real-time threat monitoring, and enriched dynamic analysis features will further strengthen detection capabilities. Overall, this approach validates behavioral machine learning as a powerful, adaptable strategy for proactive ransomware detection, significantly contributing to advancing intelligent and resilient cybersecurity defense systems.

8. FUTURE SCOPE

In the future, This project develops a machine learning system detecting ransomware through temporal API call pattern analysis. A Random Forest classifier trained on behavioral features distinguishes benign from malicious software, including unknown variants and zero-day attacks. Integrated into a secure Flask web application, the system delivers real-time classification, scalable architecture, and a foundation for future enhancements including batch processing, real-time monitoring, and threat intelligence integration.

REFERENCES

[1] P. O'Kane, S. Sezer, and D. Carlin, ''Evolution of ransomware,'' IET Netw.,vol. 7, no. 5, pp. 321–327, Jun. 2018.

[2] G. O. Gorman and G. McDonald, ''Ransomware : A growing menace,''Symantec, vol. 1, p. 16, Aug. 2012.

[3] H. Tuttle, ''Ransomware attacks pose growing threat,'' Risk Manage.,vol. 63, no. 4, pp. 4–7, 2016.

[4] Threat of Ransomware Remains at Peak With Half of Organizations FallingVictim in the Last Year, Athena Inf. Solutions Pvt. Ltd, India, New Delhi,2023, pp. 1–4.

[5] P. Chakraborty, ''Ransomware remains major threat as Sophos reports stateof cyber security in 2023,'' Gurgaon Athena Information Solutions Pvt.Ltd, India, Tech. Rep., 2023, pp. 2023–2024.

[6] M. Sunidhi, ''Elastic global threat report 2023 reveals dominance ofransomware,'' Athena Information Solutions Pvt. Ltd, India, Mumbai,Tech. Rep., 2023, pp. 3–5.

[7] CISO Research Reveals 90 % of Organisations Suffered at Least One Major Cyber Attack in the Last Year; 83 % Report Ransomware Payments, Athena Information Solutions Pvt. Ltd, India, Mumbai, 2023, pp. 1–3.

[8] J. Porter, ''Wolverine part of massive insomniac games leak afterransomware deadline passes,'' Verge New York City, USA, Tech. Rep.,2023.

[9] B. Yamany, M. S. Elsayed, A. D. Jurcut, N. Abdelbaki, and M. A. Azer,''A holistic approach to ransomware classification: Leveraging static and dynamic analysis with visualization,'' Information, vol. 15, no. 1, p. 46,Jan. 2024.

[10] Y. Wang, Z. Li, and Y. Zhang, ''Optimized ransomware detection through reverse Bayer analysis of file system activities,'' OSF Preprints, 2024.

[11] C. Beaman, A. Barkworth, T. D. Akande, S. Hakak, and M. K. Khan, ''Ransomware: Recent advances, analysis, challenges and future research directions,'' Comput. Secur., vol. 111, Dec. 2021, Art. no. 102490.

[12] H. N. Nguyen, F. Abri, V. Pham, M. Chatterjee, A. S. Namin, and T. Dang, ''MalView: Interactive visual analytics for comprehending malware behavior,'' IEEE Access, vol. 10, pp. 99909–99930, 2022.

[13] M. Alazab, S. Venkataraman, and P. Watters, ''Towards understanding malware behaviour by the extraction of API calls,'' in Proc. 2nd Cybercrime Trustworthy Comput. Workshop, Jul. 2010, pp. 52–59.

[14] D. Sgandurra, L. Muñoz-González, R. Mohsen, and E. C. Lupu, ''Automated dynamic analysis of ransomware: Benefits, limitations and use for detection,'' 2016, arXiv:1609.03020.

[15] T. Munzner, ''Visualization analysis and design-presentation,'' Vis. Anal. Design, vol. 16, pp. 1–3, Aug. 2014.

[16] M. Wagner, A. Rind, N. Thür, and W. Aigner, ''A knowledge-assisted visual malware analysis system: Design, validation, and reflection of KAMAS,'' Comput. Secur., vol. 67, pp. 1–15, Jun. 2017.

[17] B. C. M. Cappers, P. N. Meessen, S. Etalle, and J. J. Van Wijk, ''Eventpad: Rapid malware analysis and reverse engineering using visual analytics,'' in Proc. IEEE Symp. Vis. Cyber Secur. (VizSec), Oct. 2018, pp. 1–8.

[18] M. Wagner, F. Fischer, R. Luh, A. Haberson, A. Rind, D. A. Keim, and W. Aigner, ''A survey of visualization systems for malware analysis,''in Proc.

Eurograph. Conf. Vis.-State Art Rep., EuroVis-STAR, Jan. 2015,pp. 105–125.

[19] S. Poudyal, K. P. Subedi, and D. Dasgupta, ‘‘A framework for analyzing ransomware using machine learning,’’ in Proc. IEEE Symp. Ser. Comput. Intell. (SSCI), Nov. 2018, pp. 1692–1699.

[20] J. Rafapa and A. Konokix, ‘‘Ransomware detection using aggregated random forest technique with recent variants,’’ Authorea, pp. 1–8,Aug. 2024.

[21] S. Razaulla, C. Fachkha, C. Markarian, A. Gawanmeh, W. Mansoor, B. C. M. Fung, and C. Assi, ‘‘The age of ransomware: A survey on the evolution, taxonomy, and research directions,’’ IEEE Access, vol. 11, pp. 40698–40723, 2023.

[22] V. Cobilean, H. S. Mavikumbure, B. J. Mcbride, B. Vaagensmith, V. K. Singh, R. Li, C. Rieger, and M. Manic, ‘‘A review of visualization methods for cyber-physical security: Smart grid case study,’’ IEEE Access, vol. 11, pp. 59788–59803, 2023.

[23] G. Richer, A. Pister, M. Abdelaal, J.-D. Fekete, M. Sedlmair, and D. Weiskopf, ‘‘Scalability in visualization,’’ IEEE Trans. Vis. Comput. Graph., vol. 30, no. 7, pp. 3314–3330, Jul. 2024.

[24] A. Ulmer, M. Angelini, J.-D. Fekete, J. Kohlhammer, and T. May, ‘‘A survey on rogressive visualization,’’ IEEE Trans. Vis. Comput. Graph., vol. 30, no. 9, pp. 6447–6467, Sep. 2024.

[25] A. Singh, A. Ikuesan, and H. Venter, ‘‘A context-aware trigger mechanism for ransomware forensics,’’ in Proc. 14th Int. Conf. Cyber Warfare Secur. (ICCWS), 2019, pp. 629–638.

[26] A. Patel and J. Tailor, ‘‘A malicious activity monitoring mechanism to detect and prevent ransomware,’’ Comput. Fraud Secur., vol. 2020, no. 1, pp. 14–19, Jan. 2020.

[27] R. Richardson and M. M. North, ‘‘Ransomware: Evolution, mitigation and prevention,’’ Authorized Administrator Digit. Commons Kennesaw State Univ., vol. 13, no. 1, pp. 10–21, 2017.

[28] K. Hammadeh and M. Kavitha, ‘‘Unraveling ransomware: Detecting threats with advanced machine learning algorithms,’’ Int. J. Adv. Comput. Sci. Appl., vol. 14, no. 9, pp. 484–491, 2023.

[29] H. Sistemas. (2019). Virustotal Public API V2.0. Accessed: Apr. 7, 2024. [Online]. Available: https://www.virustotal.com/en/documentation/publicapi/

[30] IPStack API. Accessed: Jul. 5, 2024. [Online]. Available: https://ipstack. com/documentation

[31] Tshark. Accessed: Apr. 5, 2024. [Online]. Available: https://www. wireshark.org/docs/man-pages/tshark.html

[32] H. Dornhackl, K. Kadletz, R. Luh, and P. Tavolato, ‘‘Malicious behavior patterns,’’ in Proc. IEEE 8th Int. Symp. Service Oriented Syst. Eng., Apr. 2014, pp. 384–389.

[33] A. Singh, R. A. Ikuesan, and H. S. Venter, ‘‘Ransomware detection using process memory,’’ in Proc. Int. Conf. Cyber Warfare Secur., vol. 17, Mar. 2022, pp. 413–422.

[34] T. R. Dendere and A. Singh, ‘‘Ransomware detection using portable executable imports,’’ in Proc. Int. Conf. Cyber Warfare Secur., Mar. 2024, vol. 19, no. 1, pp. 66–74.

[35] J. Ferdous, R. Islam, A. Mahboubi, and M. Z. Islam, ‘‘AI-based ransomware detection: A comprehensive review,’’ IEEE Access, vol. 12, pp. 136666–136695, 2024.

[36] Corvus Forensics. (2019). Virus Share. Accessed: Apr. 4, 2024. [Online]. Available: https://virusshare.com/

[37] N. Naik, P. Jenkins, N. Savage, L. Yang, T. Boongoen, N. Iam-On, K. Naik, and J. Song, ‘‘Embedded YARA rules: Strengthening YARA rules utilizing fuzzy hashing and fuzzy rules for malware analysis,’’ Complex Intell. Syst., vol. 7, no. 2, pp. 687–702, Apr. 2021.

[38] A. Panaras, B. Silverstein, and S. Edwards, ‘‘Automated cooperative clustering for proactive ransomware detection and mitigation using machine learning,’’ TechRxiv, pp. 1–9, Sep. 2024.

[39] S. Zhang, T. Du, P. Shi, X. Su, and Y. Han, ‘‘Early detection and defense countermeasure inference of ransomware based on API sequence,’’ Int. J. Adv. Comput. Sci. Appl., vol. 14, no. 10, pp. 632–641, 2023.

[40] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, ‘‘Improved training of Wasserstein GANs,’’ in Proc. Adv. Neural Inf. Process. Syst., vol. 30, Dec. 2017, pp. 5769–5779.

[41] Python. Accessed: Apr. 4, 2024. [Online]. Available: https://docs.python. org/