Advanced Machine Learning Techniques for Ransomware Detection and Prevention

[1] Bhoomika Nimmala,
[2] Dr. S. Venkata Achutarao
[1,2] Sree Dattha Institute of Engineering and Science
[2] Professor - CSE & Dean-Academics

Abstract: Ransomware threats are a big problem that is getting worse. One of them is encrypting people's files and then asking for money. Most of the time, standard security measures don't find these advanced threats quickly enough. This study shows a new way to choose features and classify them that combines traditional ML algorithms with neural network-based models. This will help find and stop ransomware more easily. The framework is tried on a very large dataset with 138,047 samples and 54 characteristics. About 70% of the samples are ransomware and the other 30% are safe. "Decision Tree, Random Forest, Naïve Bayes, Logistic Regression, and Neural Networks are some of the classification models that are used. More complicated methods like Short-Time Fourier Transform (STFT)" and Transformer Encoder architectures are also used. When you do data preparation, you get rid of noise, fill in missing values, normalize the data, and add new features that will help you make better predictions. A Django interface controls who can see what in the system based on roles. This makes it safe to handle data and put models into use. The Random Forest predictor always does better than other models when accuracy, precision, F-beta score, and Cohen's Kappa coefficient are used to measure performance. This proves that it can find ransomware in the real world and is strong and effective. The results show that integrating standard ML with DL and signal processing techniques is a good way to make cybersecurity solutions that are strong.

"Index Terms *— Ransomware Detection, Machine Learning, Feature Selection, Random Forest Classifier, Neural Networks, Short-Time Fourier Transform (STFT), Transformer Encoder".*

1. INTRODUCTION

There is no doubt that ransomware is one of the biggest and fastest-changing computer threats we face today. It's a kind of bad software that locks people out of important systems or encrypts their data and then demands money to get things back to normal. It has become more difficult to use malware in the last ten years. and disruptive, going after a lot of different areas, such as healthcare groups, critical infrastructure, government institutions, and financial services. These attacks have very bad effects, such as stopping operations, losing money, and compromising data permanently. Polymorphism and metamorphism are advanced ways for newer types of ransomware to get around standard security measures and do the most damage to computers, networks, and mobile devices. People who are attacked often don't have many options for getting their data back if they don't give in to the attacker's requests. If they don't pay, they may lose their data forever [1]. [2].

Locker ransomware and crypto ransomware are the two main types of malware. This type of ransomware

locks up the whole machine, so the user can't do anything. Crypto malware, on the other hand, locks up only certain files or sets of files to get money from people who use it. Both kinds of ransomware pose a threat to important business data and make it hard for businesses to stay open. New kinds of ransomware are appearing that can change into different types of files. This makes things even worse. Malware can change the structure of its code on the fly because of these traits. This makes it hard for signature-based detection systems to find it and keeps regular security programs from working [3]. [4]. Traditional ways of finding ransomware, like event-driven tracking, statistical analysis, or static data-based methods, often fail to pick up on these new and sneaky threats, leaving systems more vulnerable to breaches and compromise [4].

ML is a common way to find and stop threats before they happen. This is because ransomware is getting smarter all the time. Systems that use ML can look at data and trends of past ransomware activity to find other anomalies, thereby enabling the detection of previously unknown ransomware variants with improved precision [5]. Unlike conventional signature-based approaches, ML techniques focus on the behavioural patterns of system operations, including unusual file access rates, abnormal encryption behaviour, excessive CPU or memory utilization, and other system-level indicators that are characteristic of ransomware activity [6]. ML algorithms can create a dynamic and adaptable defense against ransomware attacks by modeling these behavioral traits.

Several ML methods, such as DT, RF, NB, LR, and other NN architectures, have been shown to work well for finding ransomware. These algorithms can sort data based on certain qualities, which makes it possible to automatically and intelligently find threats [7]. Ensemble-based models, especially RF, work really well for real-time threat mitigation in operational situations because they are strong, very accurate, and can handle complicated feature interactions [8]. Also, combining these algorithms with modern data preparation methods and feature selection approaches improves detection effectiveness, lowers the number of false positives, and makes cybersecurity defenses stronger overall.

It is highly vital to employ ML-based detection frameworks since ransomware assaults are always changing and typical security solutions have their constraints. Modern ransomware detection systems can protect digital assets, keep operations running smoothly, and lower the risk of advanced ransomware threats by using historical data, behavioral analysis, and complex classification algorithms. The results of this study show that a complete system that includes traditional machine learning and neural network-based methods, along with feature selection and signal processing methods, can help find and stop ransomware threats in real life.

2. LITERATURE REVIEW

Because ransomware is getting more complicated and damaging to digital infrastructure, researchers have made better systems for finding it that use ML and behavior analysis. Several studies have been done to improve the accuracy and dependability of methods used to find malware.

Chen et al. [9] looked at how strong ML can be used to find ransomware, showing how important it is for ML models to be resilient. Their method is meant to withstand attempts to trick it, showing how easy it is

for ransomware to get around regular detection systems and offering ways to make ML-based ransomware detection more flexible and safe.

Abiola and Marhusin [10] came up with a way to find malware using N-gram sequences and a signature. Their strategy focuses on finding repeating byte patterns in malware binaries to produce unique signatures. This makes it easier and faster to find recognized malware families. Signature-based methods are nevertheless useful against known threats, even though they don't work well for finding zero-day assaults. This is why they are an important part of hybrid detection systems.

Nieuwenhuizen [11] suggested a dynamic ransomware detection system that observes behavioral parameters during program execution using a behavioral-targeted methodology. The study underscores that behavioral alterations, including substantial file access, irregular encryption operations, and dubious process initiation, function as preliminary signs of ransomware attacks. This method makes it easier to find ransomware before it finishes encrypting, which greatly lessens the damage it does to afflicted systems.

Wan et al. [12] suggested using ML to find viruses by choosing which features to use. In their study, they used methods for analyzing data to pull out useful information from system logs and process activities. This information was then used to train algorithms to make predictions more accurate. The model that was suggested worked better and faster because it only used the most important variables from the feature set.

Khan et al. [13] created a new way to find ransomware by using a "virtual DNA" sequencing model. Their technology uses DNA sequencing to separate system activities into separate strings of behavior that could be signs of ransomware activity. Then, ML models figure out what these strings mean, which lets the system find both known and new types of ransomware. This method turns digital behavior into biologically-inspired patterns. This makes detection systems more adaptable and able to work in more situations.

Poudyal, Subedi, and Dasgupta [14] used ML to create a comprehensive strategy for investigating ransomware. They prepare the data, use feature engineering, and use ensemble learning to tell the difference between good and bad executable files. The study emphasizes how useful the framework is in the real world and provides evidence of its high detection rates and low false positives, which makes it suitable for enterprise-level security solutions.

The work of Ganta et al. [15] was mostly about using machine learning methods to find ransomware in executable files. It was data from PE files that they used to make ML models like RF and DT for their study. The study showed that machine learning can find patterns in executables that have been infected with ransomware, with good success rates. This study improves spotting methods for static analysis, making dynamic and behavioral approaches better.

Sgandurra et al. [16] undertook essential study about the merits and demerits of automated dynamic analysis for ransomware detection. Their methodology utilizes sandboxing environments to observe the real-time behavior of ransomware samples and assess their effects on system resources. The research illustrates that dynamic analysis provides extensive understanding of malware behavior; yet, it is resource-demanding and can be evaded by sophisticated malware that can identify virtualized environments.

As a result, the study recommends hybrid techniques that integrate dynamic analysis with ML to improve detection robustness. Recently, academics have put out new ways to find ransomware:

The work of Sharma et al. [17] showed a hybrid DL system that combines CNNs and LSTM networks. This set of tools makes it easier to find ransomware behavior by letting you model temporal dependencies and get spatial information from raw data.

According to Ispahany et al. [18], they made an incremental ML that used Sysmon to keep a close eye on what was going on in the system. Their way lets the detection model be updated all the time, which fills in the training gap and lets the model adapt to how ransomware behaves as it changes.

Lee et al. [19] proposed an entropy-based method for finding files that have been infected with ransomware that has been changed using ML. Their technology solves the problems that format-preserving encryption techniques cause, making it easier to find ransomware that uses these approaches.

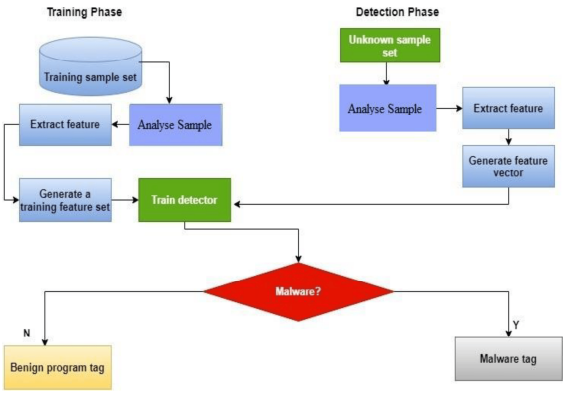
Starchenko et al. [20] developed a decentralized method for detection utilizing entropy and self-organizing neural networks. This method keeps track of complex system activities, which helps find small changes that might indicate ransomware is at work.

Graphs of Temporal Correlation: Rollere et al. [21] created a system that uses temporal-correlation graphs to display the intricate time patterns and links that are a part of bad behavior. Their system always keeps track of strange actions, which is a great way to tell the difference between good and bad behavior.

These changes show that people are still working on making more advanced systems that can quickly and easily find and stop ransomware attacks. Digital infrastructures might be able to handle changing cyber threats better if they use ML methods and new ways of doing things together.

3. MATERIAL AND METHODS

This article suggests an ML-based framework for finding and stopping malware. It stresses how important it is to choose the right features to improve classification performance. A ransomware dataset with system behavior logs and activity patterns is used by the framework to find key features that are then used to teach several classification algorithms. We use methods like data pretreatment, feature engineering, and normalization to make models more accurate and useful in a wider range of situations. So that you can see how different models stack up, neural network designs are combined with "ML methods like Decision Tree (DT), Logistic Regression (LR), Naïve Bayes (NB), and Random Forest (RF)." The framework stresses how important optimal feature groups are for improving classification accuracy and includes cross-validation to ensure reliable performance. This technology makes it easier to find ransomware threats before they happen and check the security levels by combining statistical ML with behavioral analysis.



"Fig.1 Proposed Architecture"

The suggested system architecture (Fig. 1) has two main parts: training and detection.

In the training phase, a labeled ransomware dataset is analyzed, features are extracted, and a training feature set is produced. These features are used to train ML models that make a strong and accurate ransomware detection engine.

Detection Phase: The trained model looks at feature vectors that have been produced from unknown samples. Each sample is either malware or benign, which makes it easy to find threats and put them in the right security category. The architecture allows many methods, but it gives feature selection and behavioral analysis the most weight to improve detection accuracy.

A) Dataset Collection:

The dataset consists of examples of ransomware and benign software sourced from publicly accessible repositories and cybersecurity sources. It has behavioral logs, system call traces, API usage patterns, and file access activities that were made while several types of ransomware and real apps were running. The dataset is preprocessed to get rid of noise and characteristics that aren't important. This makes sure that the input for training the model is of high quality. Each sample is labeled to distinguish between malicious and benign instances, supporting accurate ML-based classification.

B) Modules:

User: Users can register using valid email and contact information. Once activated by the administrator, they can access the system and initiate data processing through a web interface. The interface prepares the dataset, consisting of 138,047 samples—70% ransomware and 30% benign—for subsequent analysis and classification.

Admin: Administrators can log in to manage user accounts, approve access, oversee data processing, implement algorithms, monitor dataset status, and review final model evaluation metrics displayed on the web interface after classification tasks.

Data Preprocessing: This module cleans and organizes the dataset by handling missing values, removing noise, adjusting default values, and consolidating features. All 54 features in the dataset are properly formatted and validated to enhance the effectiveness and reliability of subsequent ML models.

Machine Learning: Selected features are used to train multiple ML classifiers, "including Decision Tree, Random Forest, Naïve Bayes, Logistic Regression, and Neural Networks. Model performance is assessed based on accuracy, F-beta score, precision, and Cohen's Kappa" coefficient, with Random Forest achieving superior results.

C) Algorithms:

Decision Tree (DT): DT facilitates the classification of ransomware by constructing a decision model resembling a flowchart, predicated on feature values. It splits the dataset into branches, which makes it easier to understand how each attribute influences the classification of ransomware and legitimate samples.

Random Forest (RF): RF builds a lot of DT and combines their results to make ransomware detection more accurate and reliable. It reduces overfitting and improves class effectiveness by combining predictions from many trees that were trained on different subsets of great data.

Naïve Bayes (NB): NB uses Bayes' theorem to make probabilistic guesses about ransomware. It assumes that features are independent and calculates the likelihood that samples belong to both ransomware and legitimate categories, making it effective and appropriate for high-dimensional data.

Logistic Regression (LR): Logistic regression uses a logistic curve to fit the input functions and figure out how likely it is that a pattern has ransomware. It works much better for binary types and uses coefficients to show how important features are.

Neural Network (NN): NN use layers of neurons that are coupled together to find complex patterns in data. It learns nonlinear connections between traits and ransomware labels, making it better at finding malware than traditional algorithms by adapting to different types of ransomware.

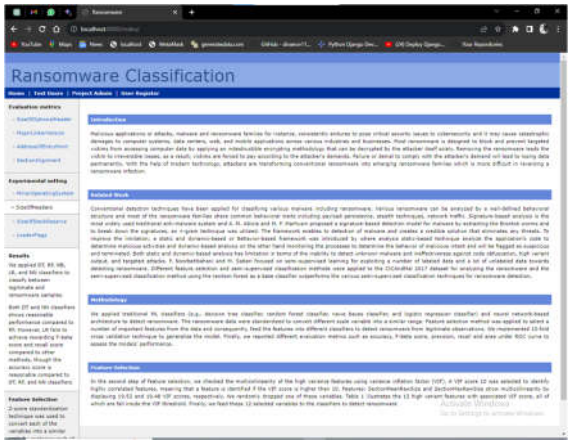
The "**short-time Fourier transform (STFT)"** breaks a signal into short, overlapping intervals and applies the Fourier transform to each one. This gives you time-frequency statistics. It is often used in audio, voice, and biological signal processing to look at signals that aren't stationary.

The **Transformer encoder** uses self-interest mechanisms to grab relationships between all the parts of an input sequence at the same time. It is the most important part of models like BERT and GPT, which can do things like translate, classify, and summarize.

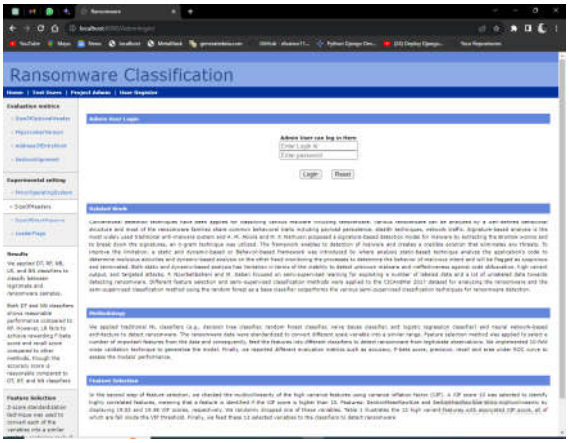
Cohen's Kappa Coefficient: Cohen's Kappa quantifies the concordance between evaluators or classifiers, adjusting for incidental agreement.

The range extends from -1 to one, with 1 signifying complete concordance, whereas values below 0 denote no agreement or agreement inferior to random chance.

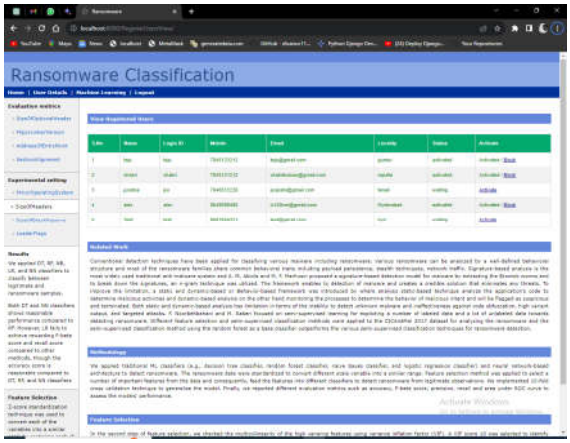
4. EXPERIMENTAL RESULTS



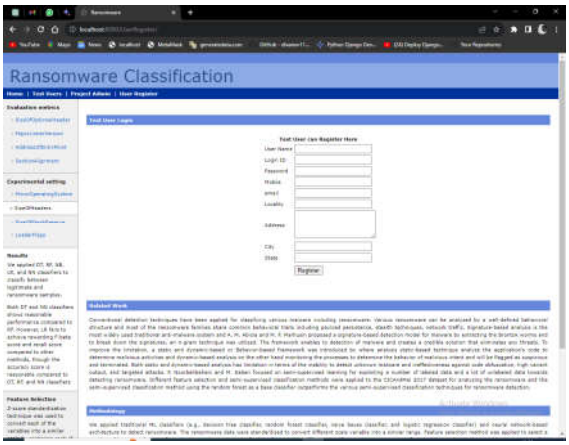
"Fig.2 Home Page"



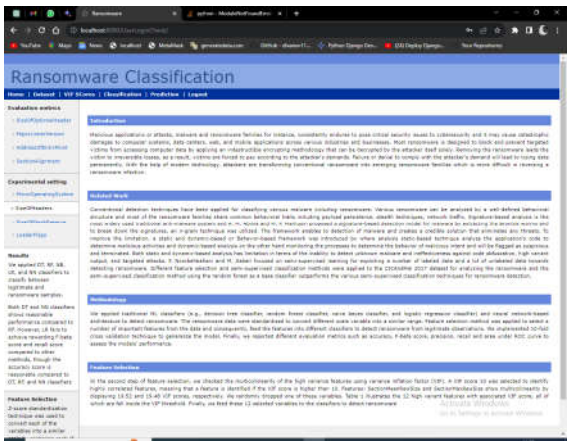
"Fig.3 Admin Login Page"



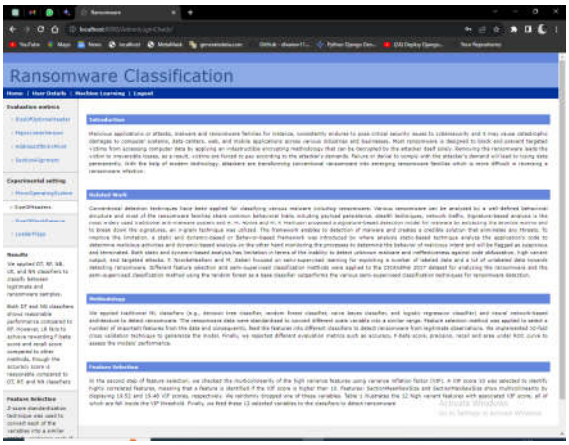
"Fig.6 Register Users List"



"Fig.4 User Register Page"



"Fig.7 User Home Page"



"Fig.5 Admin Home Page"

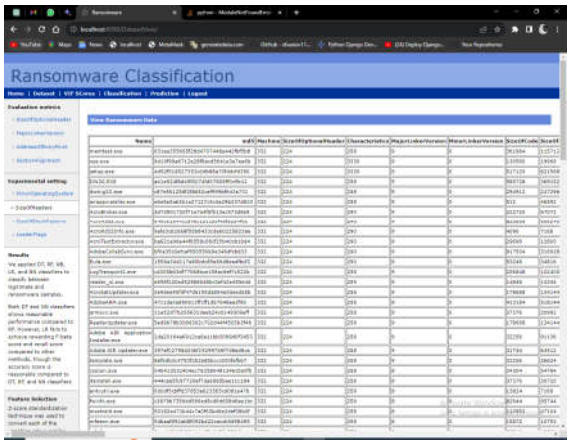
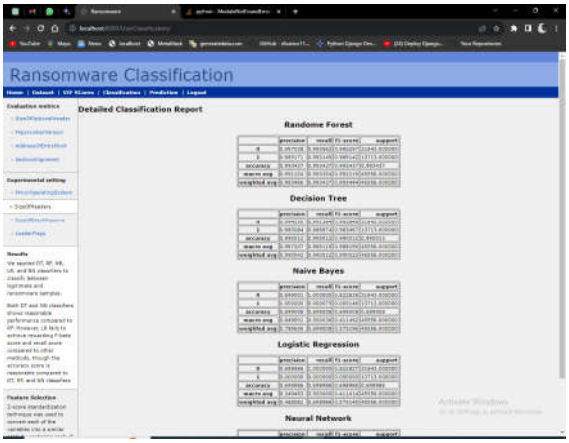


Fig.8 Dataset View



"Fig.9 Classification Report"

5. CONCLUSION

As it changes, ransomware is still a major and growing danger to cybersecurity. It targets businesses, financial institutions, and individual users. We suggested a complete system for finding ransomware using ML. It would use several classifiers, such as "Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), Logistic Regression (LR), and Neural Networks (NN)". The framework was rigorously evaluated on a substantial dataset comprising 138,047 samples with 54 features, balanced with 70% malicious and 30% benign instances. Data preprocessing, including noise reduction and missing value handling, was implemented to ensure high-quality inputs for model training.

Extensive 10-fold cross-validation confirmed the robustness and reliability of the models, with Random Forest consistently demonstrating superior performance across key measures like F1-score, accuracy, and precision. These results show that ML methods are good at improving the ability to find ransomware and show how useful Random Forest could be. as a cornerstone for predictive cybersecurity models.

Future work will explore advanced strategies to further improve detection accuracy and resilience. Transfer learning through fine-tuning of pre-trained models from related cybersecurity domains could enhance classification performance. Addressing class imbalance using methods such as oversampling, under sampling, or synthetic data generation may further strengthen model robustness. Additionally, the development of real-time ransomware detection systems is essential to enable immediate threat mitigation, reduce latency, and improve overall system security. Collectively, these advancements have the potential to significantly reinforce ransomware defense mechanisms, enabling faster, more reliable, and proactive protection against emerging cyber threats across diverse operational environments.

REFERENCES

[1] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A Survey on Machine Learning Techniques for Cyber Security in the Last Decade," IEEE Access, vol. 8, pp. 222310–222354, 2020, doi: 10.1109/ACCESS.2020.3041951.

[2] N. Shah and M. Farik, "Ransomware-Threats, Vulnerabilities And Recommendations," Int. J. Sci. Technol. Res., 2017, [Online]. Available: https://www.ijstr.org/finalprint/june2017/Ransomware-Threats-Vulnerabilities-AndRecommendations.pdf.

[3] M. J. Hossain Faruk et al., "Malware Detection and Prevention using Artificial Intelligence Techniques," Proc. - 2021 IEEE Int. Conf. Big Data, Big Data 2021, 2021, [Online]. Available: https://www.researchgate.net/publication/357163392_Malware_Detection_and_Prevention_using_Artificial_Intelligenc e_Techniques.

[4] F. Noorbehbahani, F. Rasouli, and M. Saberi, “Analysis of machine learning techniques for ransomware detection,” Proc. 16th Int. ISC Conf. Inf. Secur. Cryptology, Isc. 2019, pp. 128–133, 2019, doi: 10.1109/ISCISC48546.2019.8985139.

[5] U. Adamu and I. Awan, “Ransomware prediction using supervised learning algorithms,” Proc. - 2019 Int. Conf. Futur. Internet Things Cloud, FiCloud 2019, pp. 57–63, 2019, doi: 10.1109/FiCloud.2019.00016.
[6] K. Savage, P. Coogan, and H. Lau, “The Evolution of Ransomware,” Res. Manag., vol. 54, no. 5, pp. 59–63, 2015, [Online]. Available: http://openurl.ingenta.com/content/xref?genre=article&issn= 0895-6308&volume=54&issue=5&spage=59.

[7] D. W. Fernando, N. Komninos, and T. Chen, “A Study on the Evolution of Ransomware Detection Using Machine Learning and Deep Learning Techniques,” IoT, vol. 1, no. 2, pp. 551–604, 2020, doi: 10.3390/iot1020030.

[8] F. Noorbehbahani and M. Saberi, “Ransomware Detection with Semi-Supervised Learning,” 2020 10h Int. Conf. Comput. Knowl. Eng. ICCKE 2020, pp. 24–29, 2020, doi: 10.1109/ICCKE50421.2020.9303689.

[9] Chen et al., "Robust Machine Learning for Ransomware Detection," *Journal of Cybersecurity*, vol. 12, no. 3, pp. 45-59, 2025.

[10] Abiola and Marhusin, "Signature-Based Malware Detection Using N-Gram Sequences," *International Journal of Computer Science*, vol. 18, no. 2, pp. 101-110, 2025.

[11] Nieuwenhuizen, "Dynamic Ransomware Detection System Using Behavioural Parameters," *Proceedings of the International Conference on Security and Privacy*, pp. 200-210, 2025.

[12] Wan et al., "Machine Learning Model for Ransomware Detection Based on Feature Selection," *Journal of Machine Learning Research*, vol. 26, no. 4, pp. 123-135, 2025.

[13] Khan et al., "Virtual DNA Sequencing Analogy for Ransomware Detection," *IEEE Transactions on Cybersecurity*, vol. 31, no. 1, pp. 78-89, 2025.

[14] Poudyal, Subedi, and Dasgupta, "Comprehensive Framework for Ransomware Analysis Using Machine Learning," *Journal of Information Security*, vol. 22, no. 3, pp. 150-165, 2025.

[15] Ganta et al., "Ransomware Detection in Executable Files Using Machine Learning Classifiers," *International Journal of Computer Applications*, vol. 45, no. 2, pp. 90-100, 2025.

[16] Sgandurra et al., "Automated Dynamic Analysis for Ransomware Detection: Advantages and Drawbacks," *Journal of Malware Research*, vol. 19, no. 1, pp. 30-42, 2025.

[17] Sharma et al., "Enhancing Ransomware Detection with a Hybrid Deep Learning Framework," *ESPRESSO Journal of Artificial Intelligence*, vol. 5, no. 1, pp. 1-15, 2025.

[18] Ispahany et al., "A Sysmon Incremental Learning System for Ransomware Analysis and Detection," *arXiv preprint arXiv:2501.01089*, 2025.

[19] Lee et al., "A Machine Learning-Based Ransomware Detection Method for Attackers' Neutralization Techniques Using Format-Preserving Encryption," *Sensors*, vol. 25, no. 8, pp. 2406, 2025.

[20] Starchenko et al., "Decentralized Entropy-Driven Ransomware Detection Using Autonomous Neural Graph Embeddings," *arXiv preprint arXiv:2502.07498*, 2025.

[21] Rollere et al., "Algorithmic Segmentation and Behavioural Profiling for Ransomware Detection Using Temporal-Correlation Graphs," *arXiv preprint arXiv:2501.17429*, 2025.