IJSRA

International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

Int. J. Sci. Res. Arch.

IJSRA

International Journal of Science and Research Archive

Research Journal Archive, INDIA

(Review Article)

Ransomware Attack Detection: Developing machine learning-based detection models

Aidar Imashev *

Department of Mathematics and Computer Science, Barry University, Miami shores, United States.

International Journal of Science and Research Archive, 2025, 16(01), 901-911

Publication history: Received on 02 June 2025; revised on 08 July 2025; accepted on 11 July 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.16.1.2047

Abstract

Today's cybersecurity infrastructure faces a significant difficulty due to the rise and development of ransomware attacks. Typically, antivirus tools that use signatures cannot identify new and fast-changing ransomware, so changes in detection are required. The piece looks at how machine learning can be used to spot ransomware during attacks. This method relies on feature engineering, where relevant details are removed and picked out from masses of activity, files, and traffic seen on the computer. Both static and dynamic features help identify whether a system is infected with ransomware before any payload is launched. Many machine learning algorithms are studied to find out if they can help model the actions of complex ransomware. Addressing model evaluation metrics such as precision, recall, F1-score, and ROC-AUC explains the limitations of using models in practice. This means the models must quickly identify threats and avoid mistakenly reporting them as false alarms in the real world.

Furthermore, the article mentions issues related to skewed data, bypassing defenses, and growing systems in applications used in real-time. Using models that apply machine learning technology, businesses can enhance their response to threats. Therefore, organizations are prepared to face new ransomware attacks using information from the data they protect.

Keywords: Ransomware Detection; Machine Learning; Feature Engineering; Model Evaluation; Threat Response

1. Introduction

Recently, ransomware has become one of the biggest security threats, attacking both public and private organizations worldwide. Ransomware threats have resulted in millions of dollars lost yearly, as more people fall victim to these attacks, and they grow more advanced. It was reported that between 2020 and 2021, hospital, school, financial, and government organizations saw a 105% growth in ransomware cases globally. As a result, systems are required that can quickly detect and stop new forms of ransomware. Signatures or rules majorly affect how traditional antivirus software and IDS detect threats. These types of systems manage to spot viruses that technicians are aware of; however, they typically miss recently created ransomware that is remodeled to evade detection (Gibert et al., 2020). Because of this problem, old security techniques do not work well today, because attackers utilize camouflage, do their work without leaving files,, and move very swiftly.

Thus, many cybersecurity experts and researchers are adopting machine learning due to its ability to detect unusual behavior. With ML, algorithms can detect threats in system and network data even before knowing their rules, making detection more accurate. This article aims to discuss how to create a ransomware detection model using artificial intelligence. It focuses on the main processes of developing features, choosing algorithms, testing models, and including them in actual threat response software. In this way, it strives to guide the use of data to enhance preemptive security measures.

* Corresponding author: Aidar Imashev

2. Ransomware behavior and detection landscape

Usually, a ransomware attack goes through three phases: the infection occurs first, then the files are all locked, and finally a demand is made for the ransom. At this stage, ransomware gets into a system by tricking users with phishing, sending harmful attachments, or finding ways into RDP services that are left open. After the attackers gain access, the ransomware encrypts your files using symmetric or asymmetric encryption. At this stage, a note is left for the victim, warning them to pay a sum in cryptocurrency to release their system from the ransomware (Scaife et al., 2016; Kharraz et al., 2015).

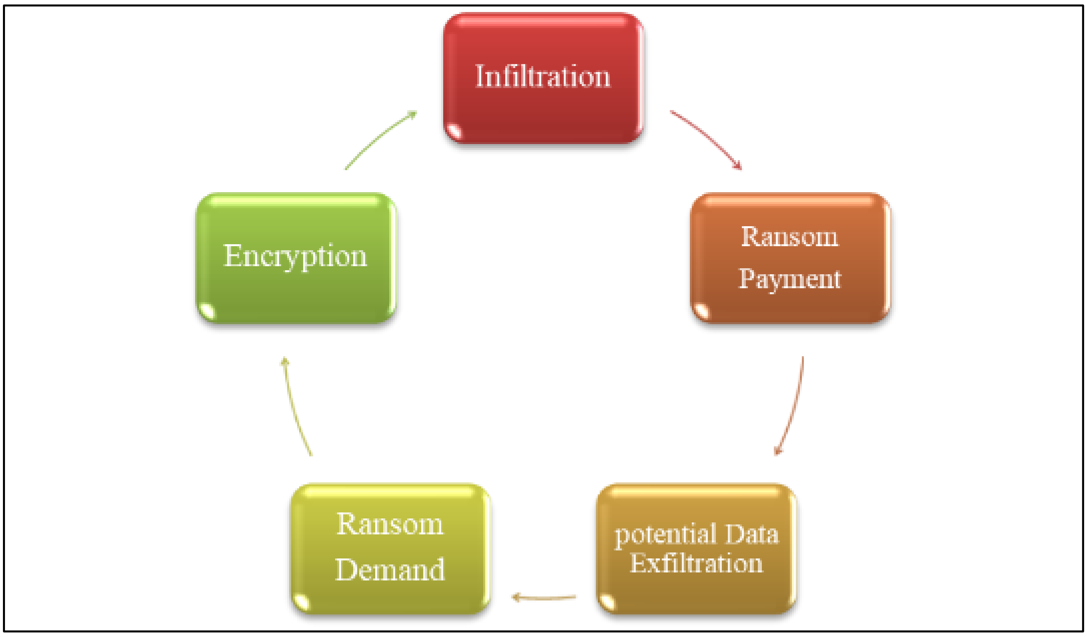

Figure 1 Typical Ransomware lifecycle

Most ransomware is sent to victims through phishing emails that encourage users to carry out harmful actions. FW has also been reported through forceful attacks on RDP, flaws in unpatched applications and harmful drive-by downloads originating from sites that have been attacked (Gibert et al., 2020; Conti et al., 2021). Thanks to botnets and MaaS platforms, it does not take much skill for anyone to use ransomware for attacks. Due to technical evasion methods, it is hard to detect ransomware. Many different ransomware families use encryption on their data and settings to prevent them from being easily recognized using signatures. Because of polymorphism, many versions of ransomware can alter how they are coded to avoid standard antivirus software. Some advanced attacks use memory execution with no visible trace on disk, and powerful, difficult-to-detect PowerShell scripts are added to called functions or DLL injection protocols.

Table 1 Comparison of Ransomware Detection Techniques

Detection Method	Strengths	Limitations
Signature-based Antivirus	Fast detection of known malware	Ineffective against polymorphic or zero-day ransomware
Heuristic Analysis	Can detect previously unseen malware variants	High false favorable rates; relies on predefined rules
Behavioral Analysis	Identifies anomalies in system behavior	Requires extensive profiling; vulnerable to evasion techniques
ML-based Detection	Adaptive, capable of learning new patterns	Needs quality training data; computationally intensive

Current virus detection methods include antivirus protection, intrusion detection systems (IDS), and heuristics. They mainly use preprogrammed signatures, analyze behavior using predefined rules, or only spot variations in traffic patterns. They perform well with known malware, but falter against new and evolving types of ransomware that use fast flux, delayed functions, and methods to avoid being revealed in a sandbox environment. Such tools are inefficient against ransomware today since it changes and evolves quickly.

For this reason, systems that rely on machine learning are necessary to deal with situations as they develop. They help track changes in a system's actions,, allowing for the early detection of ransomware, before the files are encrypted.

3. Feature engineering for ml-based detection

For ransomware detection, the accuracy and consistency of a machine learning model depend mainly on the effective use of relevant data in their development. When dealing with cybersecurity, features are gained from raw system or network data and help you know which parts of that data is suspicious. Before developing features, it is necessary to carefully study how ransomware works and behaves.

3.1. Static Features

Static features can be found in files without running the programs. These include:

- File hash values, for instance SHA-256, are common for recognizing specific types of ransomware, yet do not work with malware that constantly changes its structure.
- Byte entropy measures how dispersed the bytes are in a file. Greater entropy is likely to be found in encrypted or compressed workloads which are commonly used in ransomware infections (Sgandurra et al., 2016).
- Opcode sequences: Contain the machine instructions that the chip can execute. Analyzing the types of opcodes found in software can help identify ransomware errors (Nari & Ghorbani, 2013).
- Because static analysis is straightforward to use, it may not identify unfamiliar or hidden versions of ransomware.

3.2. Dynamic Features

- Dynamic features are identified by trying out a sample in a secure testing environment and tracking its activity.
- Activities like generating unnecessary access to files or putting new processes into the computer can be signs of ransomware.
- Actions in the network: This can show up as using known C2 servers, using the DNS for communication or behaving with abnormal outbound traffic.
- If you notice bulk encryption of files, renaming of extensions and removal of shadow copies, your system may be compromised.
- They become very important, as ransomware will usually do obvious things before encrypting data. Experts found that dynamic analysis can enhance the detection of zero-day attacks and malware that impacts fileless systems (Raff et al., 2018; Ucci et al., 2019).

3.3. Behavioral Features

- Behaviors are not restricted to individual actions but include more wide patterns.
- How users interact with the app: If an app only takings basic inputs from users, it is usually okay; ransomware, however, continues to operate in the background.
- Sudden encryption of several files all at once is a clear indication that something might be wrong.
- To ensure it starts up after a computer reboot, ransomware often depends on using registry keys or timed tasks.
- Higher-level indicators can be used to identify ransomware from other types of malware and harmless issues.

4. Importance of feature selection and domain expertise

Since there are many possible features a model could use, selecting the most important ones is key both to boosting its performance and saving processing time. RFE, score based on mutual information or PCA are ways that help decreases the number of features without decreasing the ability to classify data. Still, it requires some additional manual work as well. Being knowledgeable in that field helps you analyze the system, notice different attack methods and identify reliable features. Even though some features are not statistically significant, a cybersecurity expert could realize that deleting backups plus changes in the registry is likely an indication of ransomware.

Table 2 Comparison of Feature Types for ML-Based Ransomware Detection

Feature Type	Examples	Advantages	Limitations
Static	File hash, opcode sequence, byte entropy	Fast, no execution needed	Ineffective against obfuscation or polymorphism
Dynamic	System calls, file modifications, network I/O	Detects real-time behavior	Requires sandboxing; slower
Behavioral	Encryption rates, persistence, user behavior	Captures high-level intent; robust to evasion	Harder to quantify; needs context

5. Machine learning approaches

Using ML in detecting ransomware means that systems can learn from various data structures and become more prepared to meet new ransomware challenges on their own. Whether one has labeled data and the goals of the model are clear, ML models are usually put into three groups: supervised, unsupervised and deep learning models. The advantages and disadvantages of each approach vary in terms of how accurate, understandable and easy they are to use.

5.1. Supervised Learning

Researchers focus on supervised learning more than other methods due to the abundance of labeled examples. The models use training data of input-output pairs and each item is labeled to show if it is benign or ransomware. Recent studies (Sgandurra et al., 2016; Alzahrani et al., 2022) indicate that RF, SVM, LR and XGBoost are effective in telling malware and ransomware apart.

- Random Forest is accurate because it builds a number of decision trees and then joins their outputs which makes the approach more resistant to overfitting.
- SVM works excellently with many dimensions and handles instances where the data is not simple to separate.
- Even though Logistic Regression is more basic, it offers explanatory results and often serves as the most basic reference model.
- This type of machine learning method known as XGBoost is popular for its ability to work on large data sets and it outmatches several traditional classifier algorithms.
- Even so, these models rely on classified data, something difficult to obtain when dealing with zero-day ransomware, a major problem in very fast-growing areas of cybercrime.

5.2. Unsupervised Learning

Sometimes, if the data is not labeled, unsupervised learning can smartly detect threats we have not seen before. These methods identify behaviors that are not normal for the system it is monitoring.

In the K-Means method, the data is divided into groups that are similar in some features, but this is not useful for data that forms less regular clusters.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) can recognize clusters of any shape and easily separate noise which is often related to malicious behavior.

Approaches such as one-class SVM and Isolation Forests, find out the pattern of healthy behaviors and alert if any attacks deviate from this pattern.

They are efficient in showing new kinds of ransomware, but they may also mistakenly classify safe behavior as dangerous and thus trigger more false alarms (Kesarwani et al., 2018).

5.3. Deep Learning.

Deep learning (DL) helps in discovering non-linear relationships between different pieces of data in a more efficient way. Methods such as CNNs and RNNs have excelled in understanding how ransomware develops over time and space.

Stan et al. took malware code, converted it to grayscale images and trained a CNN to identify if a file was safe or dangerous.

Long Short-Term Memory (LSTM) networks and similar RNNs are useful for real-time monitoring of user actions on systems because they can process logs and documented API behavior.

Autoencoders are a group of neural networks used for learning without labels. They recognize normal behavior and point out when something looks abnormal, which helps detect ransomware attacks.

Deep learning plans do have potential, but they face some issues: They are hard for computers to solve, need vast amounts of data to be effective, and usually cannot be properly understood, which puts them at a disadvantage in being used in production security systems.

Table 3 Comparison of ML Approaches for Ransomware Detection

Learning Type	**Algorithm**	**Strengths**	**Limitations**	**Use Case**
Supervised Learning	Random Forest	High accuracy, ensemble learning	Needs labeled data	Known ransomware classification
	SVM	Effective in high-dimensional spaces	Computationally expensive for large datasets	Binary classification tasks
	Logistic Regression	Simple, interpretable	Limited to linear relationships	Baseline model
	XGBoost	Fast, scalable, high-performance	May overfit on noisy data	High-performance classification
Unsupervised Learning	K-Means	Simple, fast clustering	Assumes spherical clusters	Initial exploratory analysis
	DBSCAN	Detects arbitrarily shaped clusters, noise-tolerant	Parameter sensitivity	Unknown variant detection
	Anomaly Detection	Detects rare events	High false positives	Zero-day ransomware
Deep Learning	CNN	Learns spatial features from binary images	Requires large datasets, less interpretable	Static file analysis
	RNN (LSTM)	Captures sequential behavior over time	High complexity, training instability	Dynamic behavior modeling
	Autoencoder	Unsupervised anomaly detection	Sensitive to reconstruction threshold	Behavioral anomaly detection

6. Model evaluation metrics

To accurately assess ransomware detection with machine learning in R, the focus should be on ensuring minimal mistakes and particularly reduced false negatives. As recently detected, ransomware attacks can destroy systems, so choosing the correct metrics helps to develop strong and reliable systems.

6.1. Accuracy, Precision, Recall, and F1-Score: Learning to Choose Among Competing Values

The accuracy measure may not work well for ransomware because it is often based on far more harmless data than malicious data. While a benign-labeling model might still do well at identifying inputs, it will not spot any threats. The higher the precision, the fewer chances for wrong alarms to be raised. However, recall is about the model knowing when a threat is real ransomware. With the F1-score, you can find an appropriate balance between not reporting potential threats too often and failing to report important ones. With high recall, potential threats are detected, and high precision reduces the number of unnecessary investigations. F1-score is designed to be fair, especially when dealing with security issues.

Table 4 Evaluation Metrics and Their Security Implications

Metric	Formula	Significance
Accuracy	(TP + TN) / (TP + FP + TN + FN)	Can obscure poor performance on the minority (malicious) class
Precision	TP / (TP + FP)	Essential for reducing false positives
Recall	TP / (TP + FN)	Critical for detecting every real ransomware attack
F1-Score	2 × (Precision × Recall) / (Precision + Recall)	Balances detection, sensitivity, and specificity

ROC-AUC and Threshold Tuning

The ROC-AUC can help show how a model can identify ransomware from benign files at varying confidence levels. The model is better at separating classes when the AUC is higher. In actual usage, ROC curves guide security analysts to ensure both threats are captured and that too many notices are not generated.

Provide a multi-line ROC curve image demonstrating the differences between various models in their sensitivity over specificity.

6.2. Confusion Matrix: Ground Truth Visualization

It shows all the classifications made by a model.

- True Positives (TP): These are correctly detected examples of ransomware
- True Negatives (TN): Valid Emails/Files That Are Recognized
- True Negatives (TN): These are correctly labeled as benign files
- False Negatives (FN): Samples of ransomware that were missed by the system

Because a single unnoticed ransomware file could result in a system-wide problem, it is vital to reduce false negatives in cybersecurity.

Table 5 Example Confusion Matrix

	Predicted: Ransomware	Predicted: Benign
Actual: Ransomware	True Positive (TP): 92	False Negative (FN): 8
Actual: Benign	False Positive (FP): 15	True Negative (TN): 885

It is important to focus on recall and FN reduction because, even with a high level of accuracy, missing just 8 ransomware files still poses a high risk.

6.3. Emphasis on Low False Negative Rates

Detecting an attack correctly is more risky than not detecting it. Permitting just one attack to happen can encrypt company data, cause services to stop working, and harm the company's reputation. Therefore, when creating security models, making sure false negatives are extremely rare matters more than having fewer false positives and less manual analysis.

7. Practical challenges and considerations

Even though machine learning is helpful in detecting ransomware, several challenges stop it from being used effectively in practice. Adversarial attacks, imbalanced training sets, difficulties with the real-time functioning of models, and privacy concerns are all reasons why models designed for AI should be handled with care during both development and operation.

7.1. Adversarial learning in machines

One more concern is adversarial machine learning (AML), where hackers purposefully input data so that ML models make unreliable decisions. Here, the attackers might make changes so small that they make it difficult for malware to be found by scanning. They expose the weaknesses of a model by recognizing its decision points and generating examples that are not detected near them. It has been shown that even minor changes to file data or system logs can make it much harder to identify malware when the model's inner functioning is unknown to the attacker (Demetrio et al., 2021). This risk can be reduced by choosing solid training methods, such as enemy training and using ensembles, and monitoring models over time for shifts.

7.2. Imbalanced Datasets

It is also tricky because the number of good events far exceeds the number of ransomware in the sets used to train the model. Many ML models usually struggle to find ransomware since they are meant to highlight the most common class. If 98% of the data labels are labeled as benign, the resulting model could still fail to detect any ransomware. SMOTE (Synthetic Minority Over-sampling Technique), cost-sensitive learning, and undersampling can be used to even out the data. However, they each add some complexity and might cause the model to overfit (Chawla et al., 2002).

7.3. Real-Time Detection

Ransomware must be detected as soon as possible, requiring quick and accurate systems. Yet, the most accurate models require a significant amount of computing power. To avoid loads from being executed, the inference process needs to occur very rapidly, making it necessary to limit how complex and large these models can be (Kolosnjaji et al., 2016). Therefore, the solution is to use lightweight systems, cut down on model components, or choose edge-based apps to handle inference duties as briefly as possible. Also, detection features must be added to host-based agents or SIEM systems, which must be customized for resource-limited systems.

7.4. Protecting and Maintaining Compliance

Table 6 Key Practical Challenges in ML-Based Ransomware Detection

Challenge	**Description**	**Potential Solutions**
Adversarial ML	Malware adapted to fool ML models	Adversarial training, model hardening
Imbalanced datasets	Fewer ransomware samples vs. benign ones	SMOTE, cost-sensitive loss functions, and data augmentation
Real-time detection	High latency in inference for complex models	Model compression, edge inference, lightweight architectures
Data privacy and compliance	Use of sensitive data violates regulatory requirements	Federated learning, differential privacy

Lastly, storing and analyzing big data about people's actions for model building can become a risk for privacy and compliance. In some cases, sensitive data about people or a company may end up in log files or file names, which can be dangerous if the data is not fully anonymized. Regulations such as GDPR and HIPAA require strong measures in handling, processing, and saving sensitive data. To prevent sharing actual data, researchers are starting to use techniques such as federated learning, differential privacy, and secure multi-party computation (Shokri & Shmatikov, 2015). However, using these approaches adds more to the overall complexity of deploying models.

8. Deployment and integration

Machine learning to detect ransomware should be precise and integrated into a company's security systems. The Security Information and Event Management (SIEM) system is mainly targeted as a central point to link and process security alerts, log data, and telemetry from the whole enterprise. Ransomware alerts and automatic handling of threats can be achieved by adding ML models to Splunk, IBM QRadar, or ArcSight SIEM platforms (Almukaynizi et al., 2020). For this to work, APIs or plug-ins are developed so that logs coming in from various systems can be processed by the inference engine, and their outcomes are sent to the SIEM dashboard for analyst review.

8.1. Edge vs. Cloud-Based Inference

Whether an ML inference engine is used on the edge or within a cloud influences how fast the system responds, how much it can grow, and how safe users' data is. Having inference on the edge increases the speed of detecting threats, which is critical to stopping ransomware early in its execution. It should be noted that while models and updates can be centralized in the cloud for inference, some network delays and extra data transfers can occur (Liu et al., 2021).

Table 7 Edge vs. Cloud-Based Inference for Ransomware Detection

Criterion	Edge Inference	Cloud Inference
Latency	Low (real-time response)	Moderate to High
Data Privacy	Higher (data stays local)	Lower (requires secure transmission)
Model Update Complexity	Higher (manual deployment on devices)	Lower (centralized update mechanism)
Scalability	Limited by device resources	Highly scalable with cloud compute
Use Case Fit	Critical systems, OT/IoT environments	Large-scale enterprise monitoring

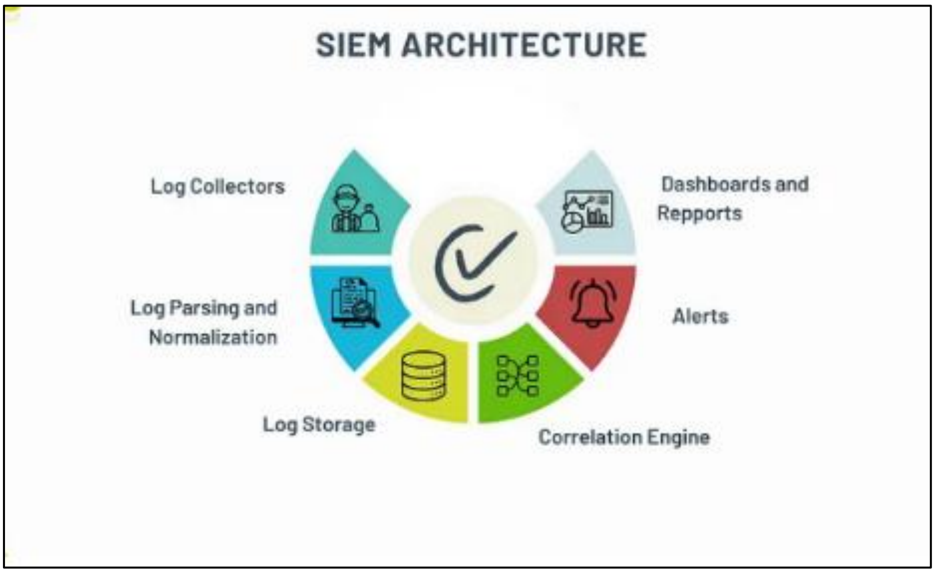

Figure 2 Hybrid Deployment of ML-Based Ransomware Detection in SIEM Architecture

8.2. Continuous Learning and Feedback Loops

Over time, static models may become less effective because attackers in ransomware often develop new strategies. Ensuring there are continuous learning pipelines allows the model to detect new ransomware and their typical behaviors. Security team members must retrain the models by providing fresh data from detection events, threat intelligence sources, and sandbox environments (Saxe & Berlin, 2015). When detection is uncertain or a closed-loop system misses an attack, a person can review these cases. After confirmation, the samples are used for further training, which sharpens the cyber defense if repeated in the future.

8.3. Automation of Incident Response Triggers

ML models used in SIEM help automate how responses are triggered. When high-confidence ransomware is detected, the system can immediately begin the actions listed in the recovery plan. Reasons for intervention may involve separating the host from the network, stopping any dubious processes, contacting specialists, or restoring the system with snapshots. When attacks are detected automatically, ransomware is stopped in seconds and can be contained much more easily (Tegeler et al., 2012).

9. Trends for the future and the best practices

Since ransomware schemes can become more complex in the future, effective detection with machine learning will depend on creative ideas to protect the model, its scaling, and ensure privacy. This section highlights the latest trends and best practices that will help design the best future cybersecurity systems.

9.1. Integrating threats from ML with results from threat intelligence.

Merging ML models and threat intelligence allows for immediate recognition of any risks. Threat intelligence covers IoCs, tactics, techniques, and procedures typical for various known attackers. When these sources are used in ML pipelines, models can find out how a behavior fits a ransomware attack and also find evidence related to the attacker. With the fusion, threat responses are streamlined, and top alerts are distinguished using already known campaigns as a reference (Mittal et al., 2021). MITRE ATT&CK and MISP are platforms that help determine if this approach is practical in improving security processes.

9.2. FedLearning, AI is designed to respect and preserve privacy

Because of concerns about privacy and regulations, raw security logs or telemetry data are seldom shared in hospitals, banks, and smart factories. FL allows several endpoints and organizations to work together to train a single global ML model without sharing their data with a central server (Yang et al., 2019). Clients train the model on their devices and send only the collected gradients to a central place. When used alongside differential privacy and homomorphic encryption, FL makes it possible to fulfill data laws such as GDPR and detect ransomware on many different platforms.

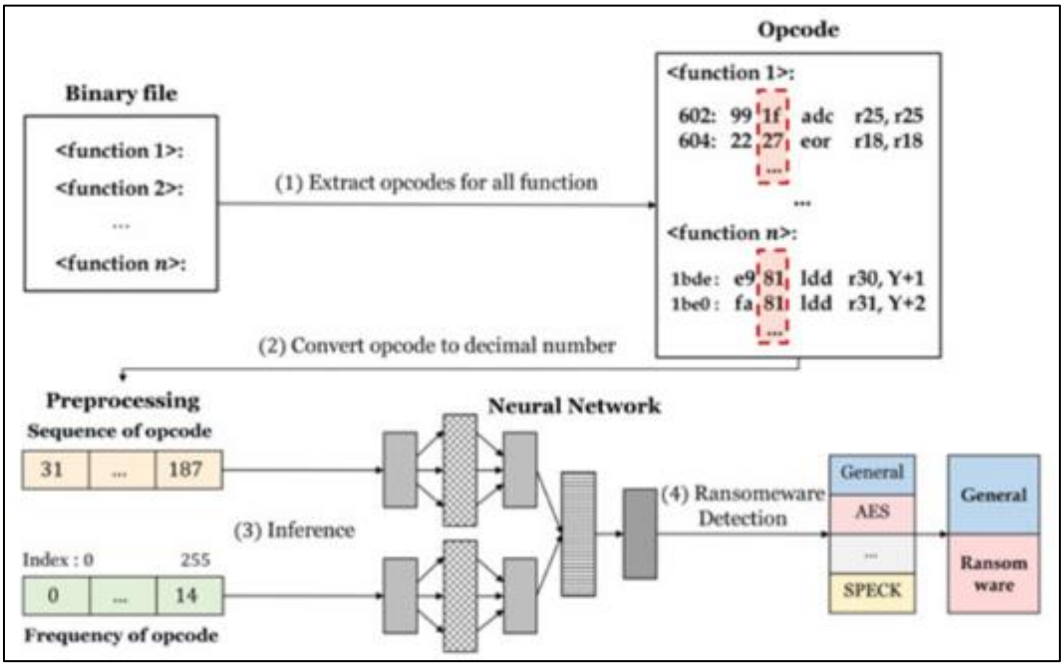

Figure 3 Federated Learning Framework for Ransomware Detection

9.3. Transfer Learning for Adapting to New Attack Families

A critical challenge in ransomware detection is the emergence of novel or obfuscated malware variants that differ significantly from those used during training. Transfer learning mitigates this challenge by allowing models trained on known ransomware families to adapt to unfamiliar threats with limited new data. This approach leverages pretrained representations—often using neural networks trained on extensive behavioral or binary datasets—and fine-tunes them using samples from new campaigns (Kolosnjaji et al., 2016). This reduces the need for large labeled datasets while enabling faster deployment of effective detection models against zero-day threats.

9.4. Use of Synthetic Data for Training Robust Model

The scarcity of labeled ransomware datasets—especially those capturing early-stage behaviors—can hinder model performance and generalizability. Researchers are increasingly leveraging synthetic data generation techniques such as generative adversarial networks (GANs), simulation environments, and controlled malware sandboxes to address this. These tools can create realistic ransomware activity traces, facilitating the training of models on a broader range of behaviors and scenarios (Rigaki & Garcia, 2018). Synthetic data also allows models to be tested against rare edge cases, improving their robustness under adversarial conditions.

Table 8 Use of Synthetic Data for Training Robust Model

Best Practice	**Description**
ML + Threat Intelligence	Fuse behavioral models with IoCs and TTPs for enriched, contextual detection.
Federated Learning	Enable decentralized model training while preserving local data privacy.
Transfer Learning	Adapt pretrained models to new attack families using minimal additional data.
Synthetic Data	Generate diverse ransomware activity for robust model training and evaluation.

10. Conclusion

As ransomware threats evolve in complexity, velocity, and scale, traditional rule- and signature-based detection mechanisms have become increasingly inadequate. The asymmetry between attackers' innovation ability and defenders' reliance on static detection underscores the urgent need for machine learning-based ransomware detection. ML models offer an adaptive, behavior-driven approach that enables early identification of malicious activity, even without known signatures, thereby transforming the cybersecurity landscape from reactive to proactive. The foundation of effective ML-driven detection lies in data quality and the relevance of engineered features. Rich, diverse telemetry—collected from endpoints, network flows, and file systems—enables models to learn subtle patterns associated with ransomware behavior. Carefully designed feature engineering pipelines amplify signal-to-noise ratios, improving classification accuracy and generalizability across environments. Moreover, model interpretability is not a luxury but a necessity; security analysts must understand why a model has flagged a particular event to enable confident, auditable responses and ensure regulatory compliance. The future of ransomware defense depends on organizations' willingness to invest in ML-centric cyber defense ecosystems. This includes not only the deployment of detection models but also the development of feedback-driven learning pipelines, integration with SIEMs, and automation of incident response mechanisms. By committing to data-driven innovation in security, enterprises can build more resilient infrastructures capable of withstanding today’s ransomware challenges—and those of tomorrow.

References

[1] Scaife, N., Carter, H., Traynor, P., & Butler, K. R. B. (2016). Cryptolock (and drop it): Stopping ransomware attacks on user data. 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS). https://doi.org/10.1109/ICDCS.2016.33

[2] Gibert, D., Mateu, C., & Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. Journal of Network and Computer Applications, 153, 102526. https://doi.org/10.1016/j.jnca.2019.102526

[3] Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., & Kirda, E. (2015). Cutting the Gordian knot: A look under the hood of ransomware attacks. DIMVA. https://doi.org/10.1007/978-3-319-20550-2_8

[4] Ugarte-Pedrero, X., Santos, I., Brezo, F., & Bringas, P. G. (2021). Countering polymorphic malware: Formalization, detection and measurement. Computers & Security, 107. https://doi.org/10.1016/j.cose.2021.102301

[5] Sgandurra, D., Muñoz-González, L., Mohsen, R., & Lupu, E. C. (2016). Automated dynamic analysis of ransomware: Benefits, limitations, and use for detection. arXiv preprint. https://doi.org/10.48550/arXiv.1609.03020

[6] Nari, S., & Ghorbani, A. A. (2013). Automated malware classification based on network behavior. Proceedings of the 2013 International Conference on Computing, Networking and Communications (ICNC). https://doi.org/10.1109/ICCNC.2013.6504127

[7] Raff, E., Zak, R., Cox, K. R., Sylvester, J., & McLean, M. (2018). Malware detection by eating a whole EXE. AAAI Workshops. https://arxiv.org/abs/1710.09435

[8] Ucci, D., Aniello, L., & Baldoni, R. (2019). Survey of machine learning techniques for malware analysis. Computers & Security, 81, 123–147. https://doi.org/10.1016/j.cose.2018.11.001

[9] Kesarwani, P., Moser, R., Zeng, Z., & Aiken, A. (2018). Adversarial Examples in Malware Detection. arXiv preprint arXiv: 1802.04528. https://doi.org/10.48550/arXiv.1802.04528

[10] Tobiyama, S., Yamaguchi, Y., Shimada, H., Ikuse, T., & Yagi, T. (2016). Malware detection with the deep neural network using process behavior. Proceedings of the 2016 IEEE Annual Computer Software and Applications Conference. https://doi.org/10.1109/COMPSAC.2016.20

[11] Shibahara, T., Yamaguchi, Y., Shimada, H., & Yagi, T. (2016). Behavior-based malware detection using deep learning. Journal of Information Processing, 24(2), 332–339. https://doi.org/10.2197/ipsjjip.24.332

[12] Gandotra, E., Bansal, D., & Sofat, S. (2014). Malware analysis and classification: A survey. Journal of Information Security, 5(2), 56–64. https://doi.org/10.4236/jis.2014.52006

[13] Gibert, D., Mateu, C., & Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends, and challenges. Journal of Network and Computer Applications, 153, 102526. https://doi.org/10.1016/j.jnca.2019.102526

[14] Kim, G., Lee, S., & Kim, S. (2018). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Future Generation Computer Systems, 79, 940–948. https://doi.org/10.1016/j.future.2018.01.005

[15] Shafiq, M., Gu, Z., Yu, F., & Latif, S. (2020). A survey of deep learning techniques for malware detection. Computers & Security, 101, 101734. https://doi.org/10.1016/j.cose.2020.101734

[16] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

[17] Kolosnjaji, B., Zarras, A., Webster, G., & Eckert, C. (2016). Deep Learning for Classification of Malware System Call Sequences. Australasian Joint Conference on Artificial Intelligence. https://doi.org/10.1007/978-3-319-50127-7_36

[18] Saxe, J., & Berlin, K. (2015). Deep Neural Network-Based Malware Detection Using Two-Dimensional Binary Program Features. 2015 10th International Conference on Malicious and Unwanted Software (MALWARE). https://doi.org/10.1109/MALWARE.2015.7413680

[19] Kolosnjaji, B., Zarras, A., Webster, G., & Eckert, C. (2016). Deep Learning for Classification of Malware System Call Sequences. Australasian Joint Conference on Artificial Intelligence. https://doi.org/10.1007/978-3-319-50127-7_30

[20] Rigaki, M., & Garcia, S. (2018). Bringing a GAN to a Knife-Fight: Adapting Malware Communication to Avoid Detection. 2018 IEEE Security and Privacy Workshops (SPW). https://doi.org/10.1109/SPW.2018.00049