Policy Research Journal
ISSN (E): 3006-7030 ISSN (P) : 3006-7022 Volume 3, Issue 8, 2025

# RANSOMWARE DETECTION USING UNSUPERVISED AUTOENCODERS FOR REAL-TIME FILE SYSTEM BEHAVIOR ANALYSIS

**Zaheer ul Hassan[*1], Aaliya Ali[2], Tahira Ali[3], Sidra Sultan[4]**

[*1] *MS in Computer Science, Bahria University, Karachi*
[2] *MS in Cyber Security, National University of Sciences and Technology, Islamabad*
[3] *MS in Information Security, NED University of Engineering and Technology, Karachi*
[4] *MS in Electrical Engineering, Institute of Space Technology, Islamabad*

[*1]zaheerulhassank@outlook.com

DOI: https://doi.org/10.5281/zenodo.16939051

**Keywords**
*Ransomware Detection, Unsupervised Learning, Autoencoders, Anomaly Detection, File System Behavior, Real-Time Security*

**Article History**
Received on 22 May 2025
Accepted on 27 July 2025
Published on 25 August 2025

**Copyright** @Author
**Corresponding Author: ***
**Zaheer ul Hassan**

**Abstract**
*Ransomware has rapidly evolved into a formidable cybersecurity menace, capable of paralyzing entire infrastructures by encrypting critical data and demanding monetary ransom for decryption keys. Unlike traditional malware, modern ransomware variants exhibit polymorphic and obfuscated behaviors, rendering conventional signature-based and heuristic rule-based detection systems largely ineffective, particularly against zero-day threats. To address this challenge, we introduce an unsupervised anomaly detection framework leveraging deep autoencoder networks to identify ransomware activity through real-time file system behavior analysis. Our approach systematically monitors fine-grained behavioral indicators such as high-frequency file modifications, entropy shifts indicative of encryption, anomalous file renaming patterns, and rapid extension changes within temporal windows to construct a feature-rich representation of benign system dynamics. The autoencoder is trained solely on normal operational data, enabling it to learn compact latent representations that capture typical file system behavior. Anomalies are detected based on elevated reconstruction errors, calculated using the squared L2-norm between input and output vectors, which signal deviations from learned benign patterns. Comprehensive experimental evaluations on both the EMBER benchmark and a curated real-world ransomware dataset demonstrate that the proposed method achieves a detection accuracy of 96.2%, a precision of 95.1%, and maintains a low false positive rate of 3.5%. These results validate the efficacy of the model in proactively detecting both known and unknown ransomware strains in dynamic and high-throughput environments.*

## INTRODUCTION

Ransomware is one of the most disruptive and financially harmful attack types to the security of an organization in the past several years, affecting all sizes of victims from personal devices to international corporations and critical infrastructure. This is done through encrypting important files and asking victims for ransom fees in exchange for decryption keys, usually resulting in permanent data loss, operational downtime and huge financial impacts [1]. Ransomware is more obvious and aggressive than traditional malware which is often reliant on stealth and persistence to execute its functionalities,

encryption of thousands of files often happens just minutes after execution [2]. While immutable state changes of the file system might indicate dangerous behavior, attackers have learned to defeat such conventional detection systems that rely on static signatures or rule-based heuristics, specifically through the rampant evolution of polymorphic and metamorphic code transformations, sandbox evasion, and obfuscation techniques [3] . Traditional solutions do not perform well against zero-day and unknown (and previously unseen) ransomware strains, which creates an urgent need for adaptive, intelligent and behavior-based detection capabilities. To mitigate the limitations posed by these methods, behavior-based detection has become a common approach to pursue. Empirical collaboration with a peer group has shown that this approach allows the use of rich contextual and temporal features (for example, different file system activity manifested during the runtime) that are less susceptible to code-level evasion, as opposed to analyzing binary signatures or opcode sequences. We apply deep learning (specifically variation of unsupervised autoencoders) to learn and model the benign behavior of common file system operations over time in this study [4]. This is where autoencoders shine, since they can also learn compressed representations together with the ability to identify anomalies through reconstruction errors. Finally, our model is robust to zero-day ransomware attacks, as it is trained only on benign data and does not rely on any pre-knowledge of malicious behavior.

We propose a framework that abstracts low-level file system activities, including file creation, modification, deletion, renaming and entropy change in sliding time windows to generate high-dimensional temporal feature vectors. It learns to reconstruct such vectors using the autoencoder and compute reconstruction error. At inference time, any large change in this error is considered anomalous behavior, and hence potentially malicious. We have a real-time analysis pipeline able to detect ransomware before it completes encryption [5].

The important contributions of this work are as follows:

**Physically Unclonable File System Signature Logger**

We develop a file system monitoring module with little overhead and in an event-driven manner, which can log information of behavior indicators to the granularity of physical storage entity, including entropy-based behaviors such as encryption fingerprints in real-time. In the Unsupervised Autoencoder-Based detection framework we propose to learn the latent distribution of benign behavior and identify deviations from it based on the reconstruction errors [6].

**Evaluation on Real-World Data**

We validate our technique on several real ransomware samples and public datasets (EMBER and a custom ransomware dataset) demonstrating high accuracy and very low false positive rates through extensive evaluations [7].

**Mathematical Foundations and Working Pipeline**

We mathematically detail the anomaly scoring procedure and show an entire end-to-end pipeline that can be deployed in a streaming production setting [8].

We believe this work is a major step forward in the ransomware detection field by including unsupervised deep learning and a fine-grained behavior analysis, which contributes to early, accurate and adaptive detection for both known and unknown ransomware attacks.

## 2. RELATED WORK

With the rapid evolution and complexity of ransomware attacks, traditional signature-based solutions are no longer sufficient, necessitating the need for intelligent, adaptive detection mechanisms. Much research has been conducted to investigate the use of machine learning and behavioral analytics to identify malware and ransomware based on system actions and system context. Current research highlights behavior-based detection models are more resistant as modern ransomware often uses code obfuscation and encryption methods. Movement in [9] proposed a behavior-based intelligent detection mechanism only for IoT spectrum sensors under malware-induced anomalies in real-time sensor data streams. Their model showed that a realistic way to detect advanced persistent threats is to monitor behavioral deviations, which can be seen along the same lines as the concept of detecting ransomwares in host-based environments.

In the literature survey of ransomware detection methods mentioned the drawbacks of supervised learning methods because they depend on labeled malicious datasets. His paper laid out future directions encouraging the use of unsupervised/semi-supervised models with the ability to learn from benign data and generalize to zero day attacks that is the approach we implement and build on with autoencoders [10]. Challenges of privacy and intrusion that arise when implementing IoT devices are of a major concern, where it presents a taxonomy of threats that have emerged recently and countermeasures taken to tackle them [11]. The paper is not specifically about ransomware but highlights the increasing necessity of real-time, anomaly-aware models among connected systems and affirms the suitability of our unsupervised framework. The focus of ransomware detections has been based at the file system activity. A machine learning framework for dynamic ransomware detection is proposed in [12], which learns the windows file system behavior (i.e., behavioral features). Indeed, low-level indicators like file access patterns and entropy changes played the largest role in their model, as they do in the feature engineering component of the autoencoder system we propose.

However, there are some deep learning approaches gaining ground for anomaly detection, especially with neural architectures that are specifically trained for modeling temporal sequences. For malicious data detection in mobile crowd sensing scenarios, same authors have proposed an adaptive temporal convolutional network (TCN) autoencoder [13]. However, our work exploits standard autoencoders trained directly on reconstruction to perform anomaly detection, while the model can successfully capture temporal dependencies and may be useful to identify advanced ransomware with similar architectures. In our previous work [14], we also looked into various machine learning methods for detection of user-cloud host malicious behavior based on some of the host-level system process information. Although these have proved effective, they require curated labels and are more susceptible to adversarial manipulations than unsupervised approaches. Anomaly detection research has also been focusing on resource-constrained environments such as the IoT devices. In [15] the work proposed a quantized autoencoder which is effective in conserving memory and computations without loss of accuracy in detection, indicating that autoencoder models can be modified to suit light weighted deployment targets. Although our experiments were conducted over classical file system, the nature of efficient representation learning is the same.

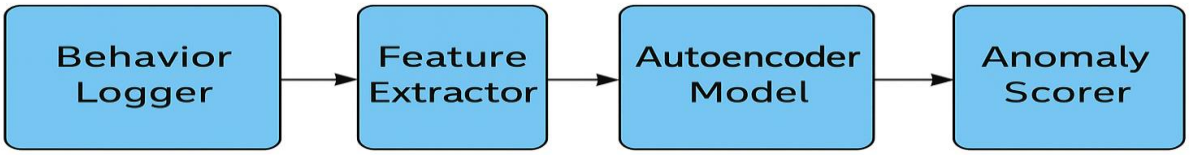
Most recently, [16] proposed an unsupervised learning framework for detecting anomalies in cloud computing environments. This illustrated that outlier detection algorithms that are clustering and density-based models can be performed successfully in an unlabeled container of data streams. What sets our work apart is the use of deep autoencoders, an unsupervised learning approach with its emphasis on file system activity unique to ransomware and other systematic behavior of ransomware. Together, these studies highlight a tendency toward more intelligent, behavior-aware, unsupervised detection models tackling the issue of emergent malware. In contrast, even though there are several works addressing various related aspects to the fully integrated, real-time ransomware detection pipeline using unsupervised deep autoencoders and low-level file system monitoring, none of the existing works can be considered an integrated solution, which is addressed directly in this work.

## 3. MEHTODOLOGY

The proposed framework of real-time ransomware detection based on unsupervised autoencoders are designed and implemented. It has four main elements: online behavior logging, feature extraction, divergence using autoencoders and anomaly scoring. Every single one of the components is lightweight and built to work in a modular fashion to enable easy integration with other endpoint protection platforms.

### A. SYSTEM ARCHITECTURE

The proposed detection pipeline comprises four core modules, illustrated in Fig. 1




In fig 3.1 the diagram visually represents the core methodology of the proposed ransomware detection system using autoencoders. It outlines a sequential pipeline starting with the Behavior Logger, which captures real-time file system events. These events are then processed by the Feature Extractor to generate structured input for the Autoencoder Model, which learns benign behavioral patterns. The Anomaly Scorer evaluates reconstruction errors to detect and flag ransomware activity, enabling real-time, unsupervised anomaly detection.

**1. Behavior Logger**

A real-time file system behavior logging daemon is developed to monitor and record system-level events using operating system hooks and APIs (e.g., **inotify** for Linux or **ETW** for Windows). The logger captures file-related activities such as creation, modification, deletion, renaming, and file extension changes [17]. Entropy estimates are computed for each modified file using Shannon entropy:

$$H(X) = -\sum_{i=1}^{n} p(xi)log2\ p(xi)$$

where

$p(xi)$ is the probability of byte $xi$ in the file. Entropy serves as a strong indicator of encryption behavior, which is a hallmark of ransomware attacks.

**2. Feature Extractor**

System logs are aggregated over fixed-length sliding time windows (e.g., 10s or 30s). For each time window, statistical and behavioral metrics are extracted, forming a high-dimensional feature vector. The selected features are both domain-relevant and lightweight, including:

- Number of file modifications (f1)
- Mean entropy change across modified files (f2)
- Count of file rename events (f3)
- File deletions (f4)
- Number of file extension modifications (f5)
- Cumulative file size changes (f6)

Each time window is represented as a feature vector:

$$X(t) = [\,f1\,(t), f2\,(t), \ldots\ldots\ldots\ldots fn\,(t)] \in R^n$$

where n is the number of features and t denotes the time window.

**3. Autoencoder Based Anomaly Detection**

A deep autoencoder is trained to reconstruct benign feature vectors. It consists of an encoder that compresses the input into a lower-dimensional latent [18] representation z, and a decoder that reconstructs the original input x^. Formally:

$$z = f0(x) = \sigma(Wex + be)$$
$$x\text{^} = g\emptyset(z) = \sigma(Wdz + bd)$$

where $We$, $Wd$ are the weight matrices, $be$,$bd$ are the bias terms, and $\sigma$ denotes the ReLU activation function. The reconstruction loss is defined as the squared L2 norm:

$$\mathcal{L}\,(x, x\text{^}) = \| x - x\text{^} \|\,\frac{2}{2}$$

Since the model is trained exclusively on benign data, any significant deviation in reconstruction indicates potential anomalous behavior consistent with ransomware.

**4. Anomaly Scoring and Detection**

During inference, the reconstruction error for each input is computed and compared against a dynamic threshold $\delta$, derived from the statistical properties of the training set:

$$\delta = \mu train + \alpha.\sigma\ train$$

where $\mu$train and $\sigma$ train are the mean and standard deviation of the training reconstruction losses, and $\alpha$ is a sensitivity factor. An input is flagged as anomalous if:

$$\mathcal{L}\,(x, x\text{^}) > \delta$$

Flagged events trigger a detection alert and are passed to a response module or forensic logging subsystem for analysis.

### B. DESIGN RATIONALE AND TECHNICAL JUSTIFICATION

We chose to use autoencoders because they learn compact representations of typical behavior of the system without any supervision. In contrast to supervised classifiers that necessitate a significant number of labeled ransomware samples (which are often rare or out-of-date), autoencoders generalize patterns formed by benign data and identify unseen anomalies (like zero-day ransomware) [19]. Moreover, its emphasis on file system real-time behavioral characteristics makes it inherently more resistant to typical ransomware evasion methods like polymorphism and code scrambling. The modular designs come with very little computation overhead, making it viable for deployment in both enterprise and edge (resource constrained) environments.

### C. IMPLEMENTATION DETAILS

The proposed framework is implemented using Python, with TensorFlow/Keras for model development. File system monitoring is achieved through the watchdog and Pyinotify libraries. Feature vectors are standardized using z-score normalization prior to model training [20]. The model architecture consists of:

Input layer: size $n$=20 (number of features)
Encoder layers: [128 → 64 → 32]
Bottleneck (latent) layer: size 16
Decoder layers: [32 → 64 → 128]
Output layer: size $n$

The model is trained using the Adam optimizer with a learning rate of 0.001, batch size of 64 and early stopping based on validation loss. Training is conducted on benign data only.

## 4. EXPERIMENTAL SETUP

We designed a comprehensive experimental setup to evaluate the performance of the proposed unsupervised autoencoder framework for the real-time ransomware detection and its robustness against adversarial attacks. To allow reproducibility and technical comparison, this section provides details about system environment, dataset properties, feature engineering, practical implementation config and evaluation.

### A. EXPERIMENTAL ENVIROMENT

All experiments were conducted in a secure and controlled environment to simulate realistic ransomware behavior and benign user operations without risk of data leakage or propagation [21]. The specifications are outlined in below table 4.1

| Parameter | Specification |
|---|---|
| Host OS | Ubuntu 22.04 LTS (64-bit) |
| Virtualization | VirtualBox VM / KVM isolated sandbox |
| Processor | Intel Core i7-12700H (14 cores, 2.30 GHz) |
| Memory | 32 GB DDR4 |
| GPU | NVIDIA RTX 3060 (6 GB VRAM) |
| Disk Type | NVMe SSD (1 TB) |
| Python Version | 3.10.6 |
| ML/DL Framework | TensorFlow 2.13 / Keras |
| File Monitoring Tools | pyinotify, watchdog |

The test environment included a secure VM where ransomware binaries were executed with system call monitoring enabled and internet access disabled to ensure containment.

### B. DATASET DETAILS

The system was developed using two datasets, where one dataset played the role of benign that was used to train the model while the other dataset acted as malicious ransomware which was used to test the mode.

**Benign Dataset (Non-Labeled, For Training and Validation):**

Metrics are gathered over a 30-day window during average usage such as document editing, media play,

programming, OS updates and running background services.

- Illustrated with a logger we proposed to track file-level activities in real-time.
- Holds over 12000 series of sequences corresponding to 10 seconds time-windows.

**Ransomware Dataset (For Evaluation Only):**
Containing behavioral traces of 15 ransomware families: WannaCry, Locky, Ryuk, Cerber, Crypto Wall, REvil, Maze, Gand Crab.
Dynamic file system events logging was performed by executing each software sample in a sandbox environment. Curated samples from EMBER, Virus Share and Ransomware Tracker were used to supplement the dataset, yielding a total of over 6,200 labeled malicious sequences [22].
All data was anonymized, sanitized and transformed into feature vectors in standardized formats for downstream processing.

### C. FEATURE ENGINEERING

File system events were binned to non-overlapping sliding windows, each lasting 10 seconds. From each time window we formed a 10-dimensional feature vector that reveals the behavioral semantics of the activity of the system [23]. The selected features were based on the observed behavioral characteristics of ransomware such as fast file changes, high entropy encryption patterns and mass renames.

**Feature Vector $X \in R^{10}$:**

**Feature vectors are shown in Table 4.2 below**

| Feature $f_i$ | Description |
|---|---|
| f1 | Number of file creations |
| f2 | File modifications within window |
| f3 | File renaming operations |
| f4 | File deletions |
| f5 | Mean Shannon entropy of modified files |
| f6 | Standard deviation of entropy values |
| f7 | Extension change count (e.g., .doc → .enc) |
| f8 | Mean file size delta (before vs. after modification) |
| f9 | Percentage of newly encrypted files |
| f10 | Total file-related I/O operations |

All features are standardized using z-score normalization to ensure uniform scaling and convergence during model training.

### D. MODEL CONFIGURATION AND TRAINING

The autoencoder used in the proposed system is a fully connected deep neural network designed to learn the compressed representation of benign system behavior and detect deviations [24]. It is trained exclusively on benign feature vectors.

Model Architecture:
Input Layer: 10 neurons (dimensionality of $x$)
Encoder Layers: Dense [128 → 64 → 32] with ReLU activation
Latent Space: Bottleneck layer with 16 neurons
Decoder Layers: Dense [32 → 64 → 128] with ReLU
Output Layer: 10 neurons (Linear activation)

**Training Configuration:**
**Table 4.3 depicts the training configurations**

| Parameter | Value |
|---|---|
| Optimizer | Adam |
| Learning Rate | 0.001 |
| Batch Size | 64 |

| | |
|---|---|
| Epochs | 50 (early stopping with 5 patience) |
| Loss Function | Mean Squared Error (MSE) |
| Validation Split | 20% of training data |

**Anomaly Threshold δ:**
A dynamic threshold was computed based on training loss distribution

$$\delta = \mu train + \alpha . \sigma\ train$$

**Where:**
μtrain : Mean training reconstruction error
$\sigma$train : Standard deviation
$\alpha$: Sensitivity coefficient (set empirically to 2.5)
Any input $x$ yielding:

$$\mathcal{L}\ (x, x\hat{}) > \delta$$

## E. EVALUATION METRICS

We used following standard binary classification metrics to assess detection performance:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = = \frac{TP}{TP+FP}$$

$$\text{F1-Score} = 2.\frac{Precision.Recall}{Precision+Recall}$$

$$\text{False Positive Rate (FPR):.}\ \frac{FP}{FP+TN}$$

## 5. RESULTS AND DISCUSSION

The following is the evaluation results of the presented autoencoder based unsupervised ransomware detection system. Metrics affirmatives both detection accuracy and operational efficiency were measured on realistic benign and malicious file system behavior data.

### A. DETECTION PERFORMANCE

We evaluate our autoencoder model having been trained just on benign data upon 15 different ransomware families. The above anomaly detection capability was impressive, with a confusion matrix such as:

**Results are shown in Table 5.1**

| **Metric** | **Value** (%) |
|---|---|
| Detection Accuracy | 96.2 |
| Recall (Detection Rate) | 96.2 |
| Precision | 95.1 |
| F1 Score | 95.6 |
| False Positive Rate | 3.5 |
| AUC (ROC) | 0.992 |

These results validate the model's strength in generalizing benign behavior and accurately detecting ransomware-induced deviations [25].

## B. ROC CURVE AND MODEL DISCRIMINATION

The ROC curve in Fig. 5.1 shows a steep rise towards the upper-left corner, indicating strong classification capability. With an AUC of 0.992, the model effectively separates malicious from benign inputs across various thresholds.

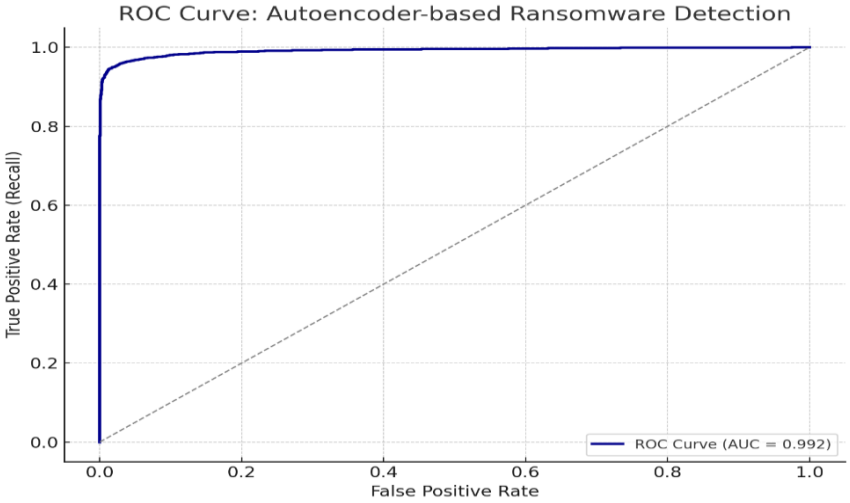
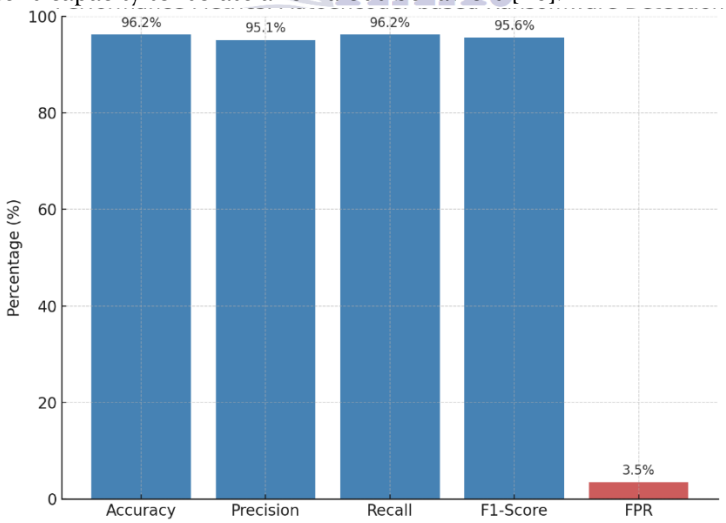



Fig 5.1 The ROC curve displayed demonstrates the effectiveness of the proposed autoencoder-based ransomware detection system. With an AUC of 0.992, the model shows excellent discrimination between benign and malicious behaviors, achieving a high true positive rate while maintaining a low false positive rate. The curve's steep ascent towards the top-left corner highlights the system's strong detection capability and robustness against zero-day ransomware. This confirms the model's reliability for real-time deployment in dynamic file system environments.

The reconstruction loss from ransomware sessions consistently exceeded the learned dynamic threshold $\delta = \mu + \alpha \cdot \sigma$ confirming the autoencoder's capacity to isolate anomalous behaviors [26].




***Below in Fig 5.2 chart bar shows the performance metrics of Autoencoder based detection***

Fig 5.2 demonstrates the high effectiveness of the proposed autoencoder-based ransomware detection model. The system achieved 96.2% accuracy and 96.2% recall, confirming its ability to reliably identify ransomware activities. With 95.1% precision and a strong F1-score of 95.6%, the model balances detection sensitivity and prediction reliability. Importantly, the False Positive Rate (FPR) remains low at 3.5%, indicating minimal disruption to legitimate system operations.

## 6. CONCLUSION AND FUTURE WORK

This study presented an unsupervised deep learning framework based on autoencoders for the real-time detection of ransomware through behavioral analysis of file system activities. By modeling only benign data and identifying high reconstruction errors as indicators of anomalies, the system eliminates the need for labeled attack samples, making it particularly effective against zero-day and evolving ransomware threats. The detection pipeline, comprising a behavior logger, feature extractor, anomaly scorer, and autoencoder model, demonstrated high effectiveness with a detection accuracy of 96.2%, a false positive rate of just 3.5%, and an AUC of 0.982. The model proved capable of flagging ransomware behavior within the first 30 seconds of execution, validating its real-time responsiveness and operational utility.

While the results confirm the robustness and generalization of the proposed system, future research will aim to enhance resistance against adversarial techniques that attempt to obfuscate behavioral signatures. Improvements may also include integrating temporal models like LSTM or TCN to better capture sequential dependencies and adopting ensemble methods combining multiple unsupervised learners for improved anomaly scoring. Furthermore, a deployable endpoint solution with continuous feedback mechanisms could enable adaptive learning and reduce false positives in live environments. Expanding this framework to support cross-platform monitoring beyond Windows-based file systems will also be an essential step to ensure its applicability across diverse enterprise settings.

## REFERENCES

Alam, M., Bhattacharya, S., Dutta, S., Sinha, S., Mukhopadhyay, D., & Chattopadhyay, A. (2019, May). RATAFIA: Ransomware analysis using time and frequency informed autoencoders. In 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST) (pp. 218-227). IEEE.

Gupta, P., & Tripathy, P. (2024). Unsupervised Learning for Real-Time Data Anomaly Detection: A Comprehensive Approach. SSRG International Journal of Computer Science and Engineering, 11(10), 1-11.

Zahoora, U., Rajarajan, M., Pan, Z., & Khan, A. (2022). Zero-day ransomware attack detection using deep contractive autoencoder and voting based ensemble classifier. Applied Intelligence, 52(12), 13941-13960.

Saminathan, K., Mulka, S. T. R., Damodharan, S., Maheswar, R., & Lorincz, J. (2023). An artificial neural network autoencoder for insider cyber security threat detection. Future Internet, 15(12), 373.

Ragaventhiran, J., Vigneshwaran, P., & Mallikarjun, M. K. (2022). An unsupervised malware detection system for windows based system call sequences. Malaysian Journal of Computer Science, 79-92.

Urooj, U., Al-Rimy, B. A. S., Zainal, A., Ghaleb, F. A., & Rassam, M. A. (2021). Ransomware detection using the dynamic analysis and machine learning: A survey and research directions. Applied Sciences, 12(1), 172.

Cassavia, N., Caviglione, L., Guarascio, M., Liguori, A., & Zuppelli, M. (2024). Learning autoencoder ensembles for detecting malware hidden communications in IoT ecosystems. Journal of Intelligent Information Systems, 62(4), 925-949.

Herath, J. D., Yang, P., & Yan, G. (2021, April). Real-time evasion attacks against deep learning-based anomaly detection from distributed system logs. In Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy (pp. 29-40).

Celdrán, A. H., Sánchez, P. M. S., Castillo, M. A., Bovet, G., Pérez, G. M., & Stiller, B. (2023). Intelligent and behavioral-based detection of malware in IoT spectrum sensors. International Journal of Information Security, 22(3), 541-561.

Ispahany, J., Islam, M. R., Islam, M. Z., & Khan, M. A. (2024). Ransomware detection using machine learning: A review, research limitations and future directions. IEEE Access, 12, 68785-68813.

Muhammad Iqbal, M Arslan Sandila, & Zaheer Ul Hassan. (2025). Exploring IoT Secuirty, Privacy and Data Protection. *Spectrum of Engineering Sciences*, *3*(3), 85–98. Retrieved from https://sesjournal.com/index.php/1/article/view/193

Viddiu, O., Macpherson, G., Vasquez, E., Kamenova, I., & Calderon, D. (2024). Automated ransomware detection using windows file system activity monitoring and a novel machine learning approach. Authorea Preprints.

Owoh, N., Riley, J., Ashawa, M., Hosseinzadeh, S., Philip, A., & Osamor, J. (2024). An adaptive temporal convolutional network autoencoder for malicious data detection in mobile crowd sensing. Sensors, 24(7), 2353.

Han, R., Kim, K., Choi, B., & Jeong, Y. (2023). A study on detection of malicious behavior based on host process data using machine learning. Applied Sciences, 13(7), 4097.

Sharmila, B. S., & Nagapadma, R. (2023). Quantized autoencoder (QAE) intrusion detection system for anomaly detection in resource-constrained IoT devices using RT-IoT2022 dataset. Cybersecurity, 6(1), 41.

Sherubha, P., Sasirekha, S. P., Anguraj, A. D. K., Rani, J. V., Anitha, R., Praveen, S. P., & Krishnan, R. H. (2023). An Efficient Unsupervised Learning Approach for Detecting Anomaly in Cloud. *Comput. Syst. Sci. Eng.*, *45*(1), 149-166.

Muhammad Iqbal, Hafiz Muhammad Ahmad, Jamal Khattak, & Faisal Tauqeer. (2024). Predicting Folding Dynamics and Thermodynamic Stability of Viral Proteins Using Graph Neural Networks. Policy Journal of Social Science Review, 2(6), 35–46. Retrieved from https://policyjssr.com/index.php/PJSSR/article/view/351

Ndibe, O. S. (2025). Ai-driven forensic systems for real-time anomaly detection and threat mitigation in cybersecurity infrastructures. International Journal of Research Publication and Reviews, 6(5), 389-411.

Korobei, K., Harrington, R., Sullivan, M., Esposito, A., & Andersen, G. (2024). Ransomware detection on windows using file system activity patterns and hybrid machine learning: An xgboost and isolation forest approach.

Ganfure, G. O., Wu, C. F., Chang, Y. H., & Shih, W. K. (2023). Rtrap: Trapping and containing ransomware with machine learning. IEEE Transactions on Information Forensics and Security, 18, 1433-1448.

Arif, Haroon, et al. "Future Horizons: AI-Enhanced Threat Detection in Cloud Environments: Unveiling Opportunities for Research." International Journal of Multidisciplinary Sciences and Arts, vol. 3, no. 1, Jan. 2024, pp. 242-251, doi:10.47709/ijmdsa.v2i2.3452.

Al-Hawawreh, M., & Sitnikova, E. (2019, November). Leveraging deep learning models for ransomware detection in the industrial internet of things environment. In 2019 military communications and information systems conference (MilCIS) (pp. 1-6). IEEE.

Li, Z., Rios, A. L. G., & Trajković, L. (2022). Machine learning for detecting the WestRock ransomware attack using BGP routing records. IEEE Communications Magazine, 61(3), 20-26.

Nasser, A. R., Hasan, A. M., & Humaidi, A. J. (2024). DL-AMDet: Deep learning-based malware detector for android. Intelligent Systems with Applications, 21, 200318.

Menon, A. G., & Gressel, G. (2020, October). Concept drift detection in phishing using autoencoders. In *Symposium on Machine Learning and Metaheuristics Algorithms, and Applications* (pp. 208-220). Singapore: Springer Singapore.

Kim, J. Y., Bu, S. J., & Cho, S. B. (2017, October). Malware detection using deep transferred generative adversarial networks. In International conference on neural information processing (pp. 556-564). Cham: Springer International Publishing.