Global Research journal of Natural Science and Technology (Grjnst)
Volume: 03 - Issue 4 (2025)
p.ISSN: 2790-7643 e.ISSN: 2790-7651

Explainable AI-Based Ransomware Early Detection Using File-System Behavioral Features and Random Forest

Dr. Ajab Khan

ORIC, Abbottabad University of Science and Technology, Abbottabad

Ajabkhan@aust.edu.pk

Fahad Amin, Member, IEEE

North American University,

Department of Computer Science-Cyber security, Houston, TX, USA

famin1@na.edu

Usman Imtiaz Member, IEEE (Corresponding Author)

Washington University of Science and Technology (WUST),

Department of Computer Science-Cyber security, Virginia, USA

usmanimtiaz1992@gmail.com

Abstract:

Ransomware is considered one of the major cybersecurity threats that has the capability to either encrypt or blocks access to organizational data. The Hacker demands compensation for recovery and identifications or prevent signature-based detection is useful, but it may fail against new variants that target by changing file names, using packed binaries, or exploit trusted system tools. The Research presented in this paper proposes to develop a method shortly referred as RF-RBD, an Artificial Intelligence-based ransomware detection model that make use of a Random Forest classifier. The model has the capability to identify ransomware like behavior by using filesystem features such as write rate, rename rate, and delete rate, entropy change, extension-change ratio, ransom-note count, shadow copy command count, and network-share access. In this study a synthetic defensive dataset is used for safe experimentation to avoid in system down time risk. The model is evaluated using parameters such as accuracy, precision, recall, F1 score, ROC-AUC, confusion

matrix, and feature importance. The Experiments performed and the associated results show a strong performance on the simulated dataset, wherein a high recall is observed along with interpretable feature rankings. The paper also explores the limitations of the approach and associated ethical concerns, adversarial risks, and suggests future improvements for real system deployment.

Keywords: *Index Terms— Artificial Intelligence, Cyber Security, Ransomware Detection, Random Forest, Machine Learning, Explainable AI, File-System Behavior, Endpoint Security*

I. Introduction

Ransomware is considered one of the most detrimental cybersecurity threats. This directly affects system availability and brings system availability into sabotage. This is archived by encrypting user files, locking business critical databases, interrupt operational services, and thus causing a major financial and reputational loss [1, 2]. The attacking method is also advancing and now it has the capability to combine encryption with data theft along with extortion, which further increases pressure on victims even when backups are readily available [1, 3].

Classical anti malware tools mainly depend on known signatures, rules, and indicators of arrangement. Although it has been observed that these methods may not work against the new variants that make use of packed binaries, living off the land techniques, or automated attacks. Further, it has been observed that ransomware can spread quickly across endpoints and network shares. If it is identified late this may lead to large scale data loss [3, 4].

The use of Artificial Intelligence can definitely improve ransomware detection by observing and identifying behavioral patterns rather than concentrating on matching known signatures. A behavior-based detector has the capability to monitor file-system activity such as rapid writing, batch renaming, file deletion, entropy increase, and ransom-note creation. These behaviors can help to identify suspicious activity even when the malware signature is unknown [4, 5].

The research work presented in this paper proposes a mode RF-RBD that makes use of the Random Forest classifier for ransomware behavior detection. The core objective is defensive and educational strategy, not offensive strategy. The model classifies and simulates user activity as benign or ransomware like. This also includes risk scoring and feature importance

analysis to explain why such alert is generated. The previous studies show that Random Forest-based approaches have also been used ransomware detection research which shows significance for classification tasks accordingly [6].

The main contributions of this paper include an AI-based early detection framework, a safe synthetic behavioral dataset along with a Random Forest evaluation model. Further, visual analysis of the results obtained is performed and associated performance charts shows the level observed.

II. Related Work

Ransomware is usually coupled with the influenced by tactic of data encryption. MITRE ATT&CK portrays this behavior as Data Encrypted for Effect, whereas the adversaries encrypt local, remote, cloud, or virtualized resources to disrupt availability and call for payment [1]. This behavior confirms the need for recognition methods that monitor file modification and encryption related activity.

CISA's Stop Ransomware guidance emphasizes preparation, prevention, detection, response, and recovery as key parts of ransomware defense [2]. Therefore, AI-based detection should be seen as one element of a larger security strategy, not as a substitute for patching, backups, access control, monitoring, and user awareness.

According to NIST Cybersecurity Framework 2.0, there are five functions to manage cyber risks: Govern, Identify, Protect, Detect, Respond, and Recover [3]. The system developed mainly addresses the Detect function by identifying anomalies, and the Respond function by raising risk-aware alerts for analysts' consumption. NIST AI Risk Management Framework is also pertinent, as the AI detectors should be validated, reliable, explainable, secure, and actively managed post-deployment [4].

It is known that the analysis of dynamic behavior and machine-learning approaches may allow ransomware detection. Herrera-Silva and Hernndez-lvarez developed dynamic ransomware features and reported good performances with a number of models (e.g., Gradient Boosting, Random Forest, neural network) [5]. Arnyi, Miseta and Szcs proved that server-side lightweight file-operation logs may support quasi real-time ransomware detection [6]. Chen et al. Addressed automatic ransomware behavior analysis through host log and pattern extraction

[7]. Yang and Sahita suggested also that behavior-based ransomware classifier should be resilient and interpretable [8].

Static malware datasets, for example EMBER [9], allow training on characteristics extracted from Portable Executable files. Static analysis is limited by techniques such as packing, obfuscation and adversarial evading techniques. Behavioral features, especially for ransomware detection, may be very helpful, as filesystem changes due to encryption and widespread modification are real run-time effects that are impossible to completely conceal from such analysis. This work focuses on the use of file-system behaviors.

III. Research Problem and Objectives

The research problem addressed in this paper is how an explainable AI model can detect ransomware-like behavior at an early stage by using file-system activity features, while remaining safe, interpretable, and suitable for defensive cybersecurity education.

The objectives of this study are:

• To design a defensive AI-based model for detecting ransomware-like behavior.

• To identify important filesystem features that may indicate encryption or suspicious file activity.

• To train and test a Random Forest classifier using simulated behavioral logs.

• To evaluate the performance of the model through standard classification metrics such as accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix.

• To present the complete framework through code, algorithm design, charts, experimental results and discussion.

• To discuss the limitations of the proposed approach, ethical considerations, and possible directions for future work.

IV. Proposed RF-RBD Model

The proposed model is named RF-RBD, which stands for Random Forest Ransomware Behavior Detector. It is designed as a host-based or file-server monitoring component that observes file-system activity within short time windows. The model does not decrypt files,

attack systems, or execute malware. Instead, it focuses only on classifying behavioral records as either benign or ransomware-like.

The architecture of the proposed model is presented in Fig. 1. First, activity logs are collected from endpoints or file servers. Next, relevant behavioral features are extracted from these logs. The data is then validated and preprocessed before being passed to the Random Forest classifier. The classifier predicts the probability of ransomware-like behavior, and a risk-scoring layer converts this prediction into an alert level for security analysts.

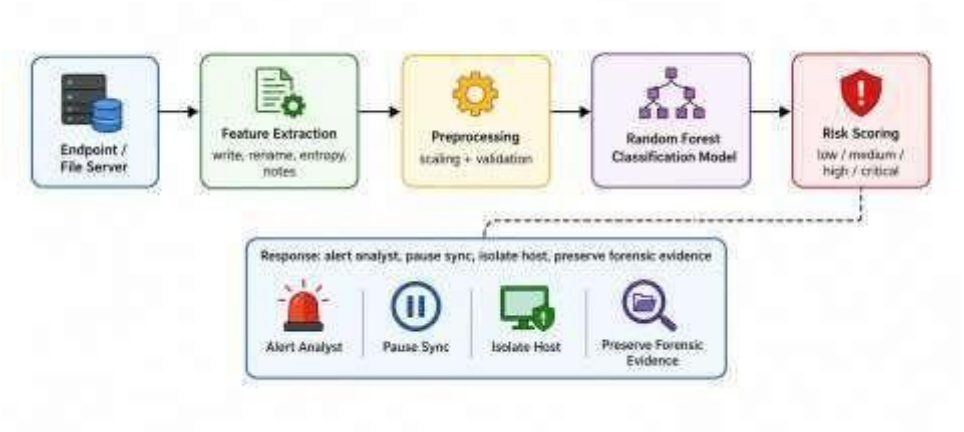

Fig. 1. Proposed RF-RBD model architecture.

A. Feature Set

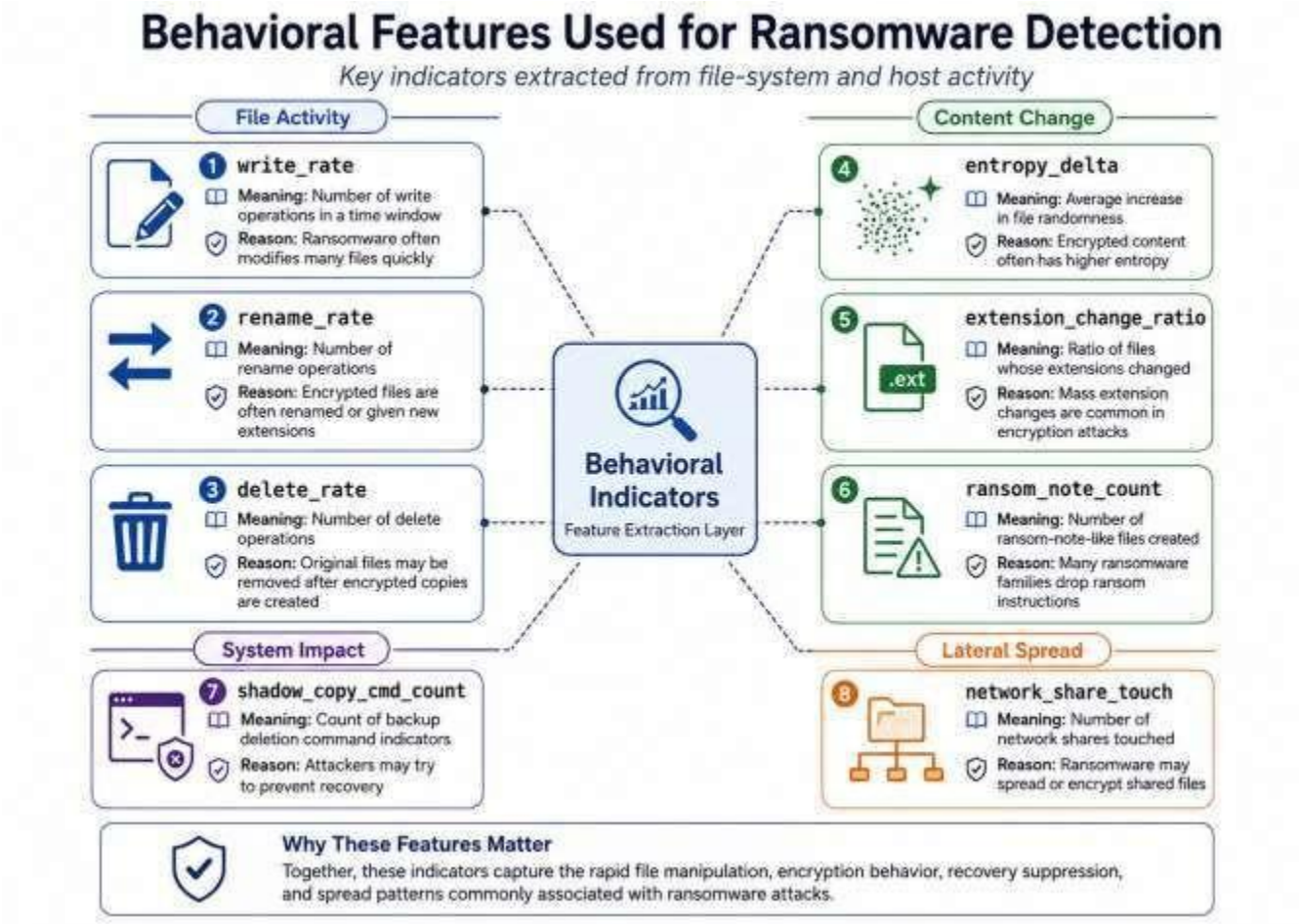

Table I. Behavioral features used by the proposed model.

B. Risk Scoring

The model returns a probability score between 0 and 1. A low probability means the behavior is likely benign, while a high probability means it is ransomware-like. The risk-scoring layer converts the probability into analyst-friendly categories.

Probability Range	**Risk Level**	**Suggested Defensive Response**
0.00–0.39	Low	Log the event and continue monitoring
0.40–0.69	Medium	Increase monitoring and correlate with other logs

0.70–0.89	High	Alert analyst and consider pausing file sync
0.90–1.00	Critical	Isolate host, preserve evidence, and start incident response

Table II. Risk levels generated from model probability scores.

V. Methodology

The research question of this paper is how to use file-system activity features to enable an explainable AI to detect ransomware-like behavior in an early stage safely, interpretably and suitable for a defensive cybersecurity education tool. The goals of this paper are to demonstrate the model design, the algorithm, and the evaluation procedure, real ransomware datasets are not used, as the goal is to build a safe to use academic prototype. Instead, a synthetically generated dataset simulating the benign and ransomware-like file-system activity is used so that no potentially harmful code, malware samples or live attack details need be shared.

The benign class comprises normal office tasks, document editing, back-up tasks, and the average file-server usage. Certain benign entries have high write-speeds representing back-up operations, as well as certain administrator operations to make the classification task harder. The ransomware-like class comprises such behaviors as an increased file-write rate, an increased rate of renaming/deleting files, an increased entropy, some indicators of a ransom note as well as access to network shares. Certain ransomware-like records have low volume behavior in order to pose more difficult challenges to the classification algorithm.

1000 simulated sessions were generated, out of which 700 are benign and 300 have ransomware-like behavior. The data was split into 70% training and 30% testing using stratified sampling so that the class distribution is maintained. The classifier used is a Random Forest classifier. A Random Forest of 220 trees was trained with class-balancing weights, with control over tree-depth, and a minimum number of leaves, to prevent over-fitting. The final performance is measured on the unseen test set.

The following Python code implements the proposed defensive prototype. It creates a synthetic behavioral dataset, trains a Random Forest classifier, evaluates performance, and produces charts. The code is intentionally safe because it does not create ransomware, encrypt files, spread malware, or perform offensive actions.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score
from sklearn.metrics import f1_score, roc_auc_score, confusion_matrix

np.random.seed(44)

# ------------------
# 1. Create safe synthetic data
# ------------------
n_normal = 700
n_ransom = 300

normal = pd.DataFrame({
    "write_rate": np.random.normal(20, 12, n_normal).clip(1, 90),
    "rename_rate": np.random.normal(4, 4, n_normal).clip(0, 25),
    "delete_rate": np.random.normal(3, 3, n_normal).clip(0, 20),
    "entropy_delta": np.random.normal(0.45, 0.35, n_normal).clip(0, 2.0),
    "extension_change_ratio": np.random.beta(1.5, 10, n_normal).clip(0, 0.6),
    "ransom_note_count": np.random.poisson(0.10, n_normal).clip(0, 2),
    "shadow_copy_cmd_count": np.random.poisson(0.04, n_normal).clip(0, 2),
    "network_share_touch": np.random.poisson(1.5, n_normal).clip(0, 8),
    "label": np.zeros(n_normal, dtype=int)
})

ransom = pd.DataFrame({
    "write_rate": np.random.normal(75, 35, n_ransom).clip(10, 170),
    "rename_rate": np.random.normal(22, 14, n_ransom).clip(0, 75),
    "delete_rate": np.random.normal(14, 10, n_ransom).clip(0, 60),
    "entropy_delta": np.random.normal(1.8, 0.75, n_ransom).clip(0.2, 4.0),
    "extension_change_ratio": np.random.beta(3, 3, n_ransom).clip(0.05, 1.0),
    "ransom_note_count": np.random.poisson(1.5, n_ransom).clip(0, 7),
```

```
    "shadow_copy_cmd_count": np.random.poisson(0.55, n_ransom).clip(0, 4),
    "network_share_touch": np.random.normal(6.5, 4, n_ransom).clip(0, 18),
    "label": np.ones(n_ransom, dtype=int)
})

df = pd.concat([normal, ransom], ignore_index=True).sample(frac=1, random_state=8)

features = [
    "write_rate", "rename_rate", "delete_rate", "entropy_delta",
    "extension_change_ratio", "ransom_note_count",
    "shadow_copy_cmd_count", "network_share_touch"
]

X = df[features]
y = df["label"]

# ------------------
# 2. Train/test split
# ------------------
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.30, random_state=7, stratify=y
)

# ------------------
# 3. Train model
# ------------------
model = RandomForestClassifier(
    n_estimators=120,
    max_depth=5,
    min_samples_leaf=8,
    class_weight="balanced",
```

```
    random_state=7
)
model.fit(X_train, y_train)

# - - - - - - - - - - - - - - - - - -
# 4. Evaluate model
# - - - - - - - - - - - - - - - - - -
y_pred = model.predict(X_test)
y_score = model.predict_proba(X_test)[:, 1]

print("Accuracy:", round(accuracy_score(y_test, y_pred), 4))
print("Precision:", round(precision_score(y_test, y_pred), 4))
print("Recall:", round(recall_score(y_test, y_pred), 4))
print("F1-score:", round(f1_score(y_test, y_pred), 4))
print("ROC-AUC:", round(roc_auc_score(y_test, y_score), 4))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))

# - - - - - - - - - - - - - - - - - -
# 5. Explainability
# - - - - - - - - - - - - - - - - - -
importance = pd.Series(model.feature_importances_, index=features)
importance.sort_values().plot(kind="barh")
plt.title("Random Forest Feature Importance")
plt.xlabel("Importance")
plt.tight_layout()
plt.show()

# - - - - - - - - - - - - - - - - - -
# 6. Risk scoring for new session
# - - - - - - - - - - - - - - - - - -
def risk_level(probability):
```

```
    if probability >= 0.90:
        return "Critical"
    elif probability >= 0.70:
        return "High"
    elif probability >= 0.40:
        return "Medium"
    return "Low"

new_session = pd.DataFrame([{
    "write_rate": 110,
    "rename_rate": 45,
    "delete_rate": 20,
    "entropy_delta": 2.7,
    "extension_change_ratio": 0.72,
    "ransom_note_count": 4,
    "shadow_copy_cmd_count": 1,
    "network_share_touch": 10
}])

probability = model.predict_proba(new_session)[0, 1]
print("Ransomware probability:", round(probability, 3))
print("Risk level:", risk_level(probability))
```

VII. Experimental Results

The model was assessed on 300 unseen test records. The test set included both benign and ransomware-like sessions. It is worth mentioning that the results are based on the synthetic dataset as discussed in the methodology section and it should be treated as prototype results, not real-world production guarantees.

Metric	Score
Accuracy	92.33%

Precision	87.64%
Recall	86.67%
F1-score	87.15%
ROC-AUC	97.50%

Table III. Performance results of the proposed RF-RBD model.

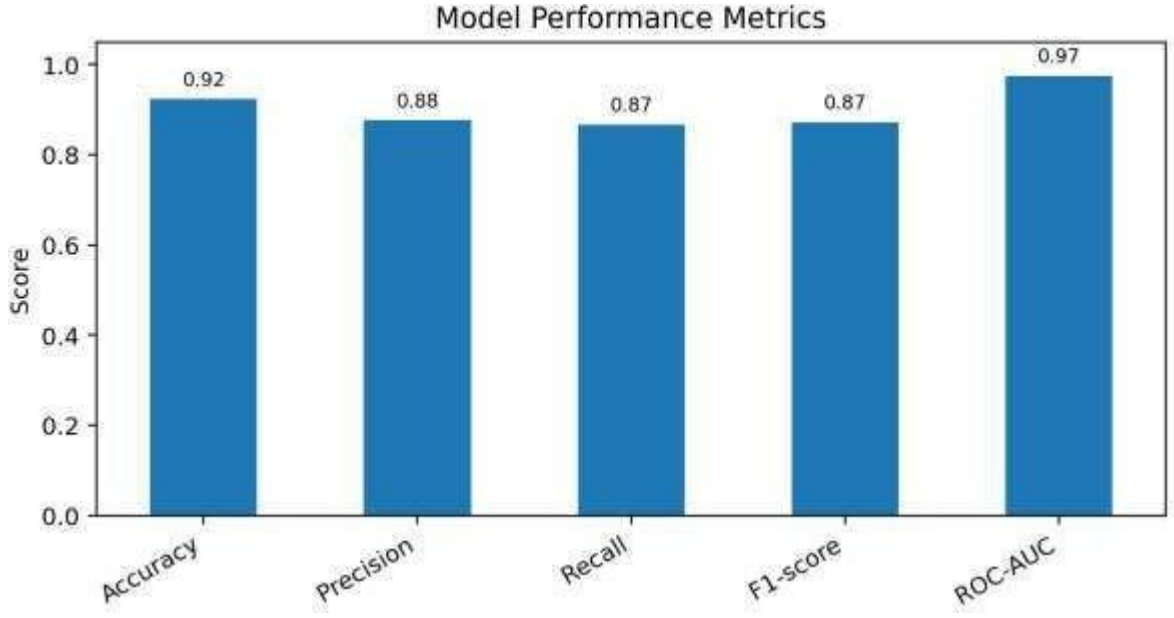

Fig. 2. Performance metrics of the proposed Random Forest model.

	Predicted Benign	**Predicted Ransomware-like**
Actual Benign	199	11
Actual Ransomware-like	12	78

Table IV. Confusion matrix values on the unseen test set.

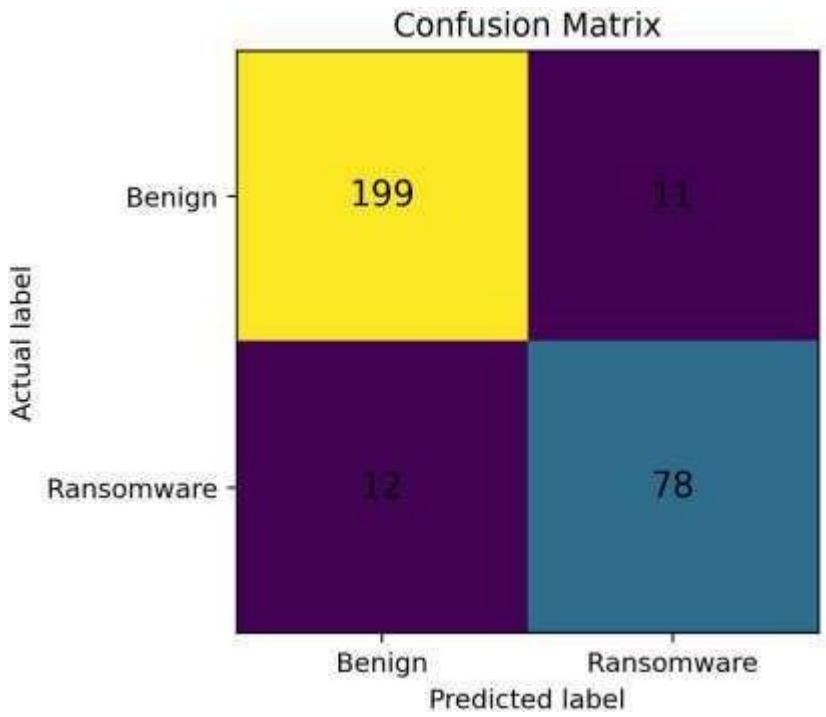

Fig. 3. Confusion matrix visualization.

A. Interpretation of Results

The experimental results show that the proposed model achieved 92.33% accuracy, 87.64% precision, 86.67% recall, 87.15% F1-score, and 97.50% ROC-AUC on the simulated test set. The confusion matrix recorded 78 true positives, 199 true negatives, 11 false positives, and 12 false negatives. In cybersecurity, recall is especially important because a false negative means that a malicious session was not detected. Precision is also important because too many false positives can cause alert fatigue for security analysts.

These results indicate that the proposed model performs effectively because ransomware-like behavior differs from normal activity across several file-system features. Entropy increase, extension-change ratio, rename rate, and ransom-note count appear to be strong indicators of suspicious activity. However, real-world environments are more complex than synthetic experiments. Backup jobs, compression tools, software updates, and data migration scripts may sometimes produce behavior similar to ransomware. Therefore, in practical deployment, model

predictions should be combined with endpoint context, user identity, process lineage, threat intelligence, and analyst review.

VIII. Explainability and Feature Importance

Explainability is important in cyber security because analysts need to understand why a model produced an alert. The Random Forest models provide feature importance values that estimate how much each feature contributes to classification. Although, feature importance is not a complete explanation, it is useful for showing which behavioral signals are most influential.

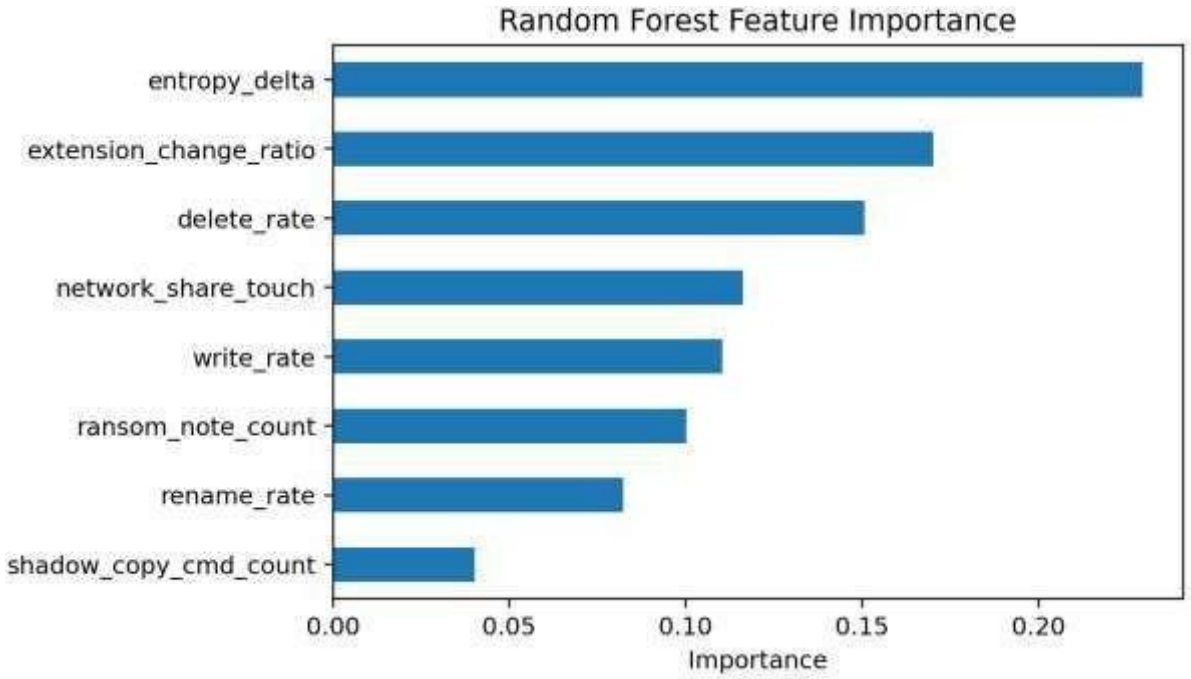

Fig. 4. Feature importance values produced by the Random Forest model.

The feature-importance chart shows that entropy change and extension-change ratio are highly useful. This is reasonable because ransomware encryption commonly changes the statistical properties of files and may rename encrypted files. Rename rate, write rate, and ransom-note count also contribute to classification. Shadow-copy command indicators are less frequent, but they are high-risk when present because ransomware operators often attempt to remove recovery options.

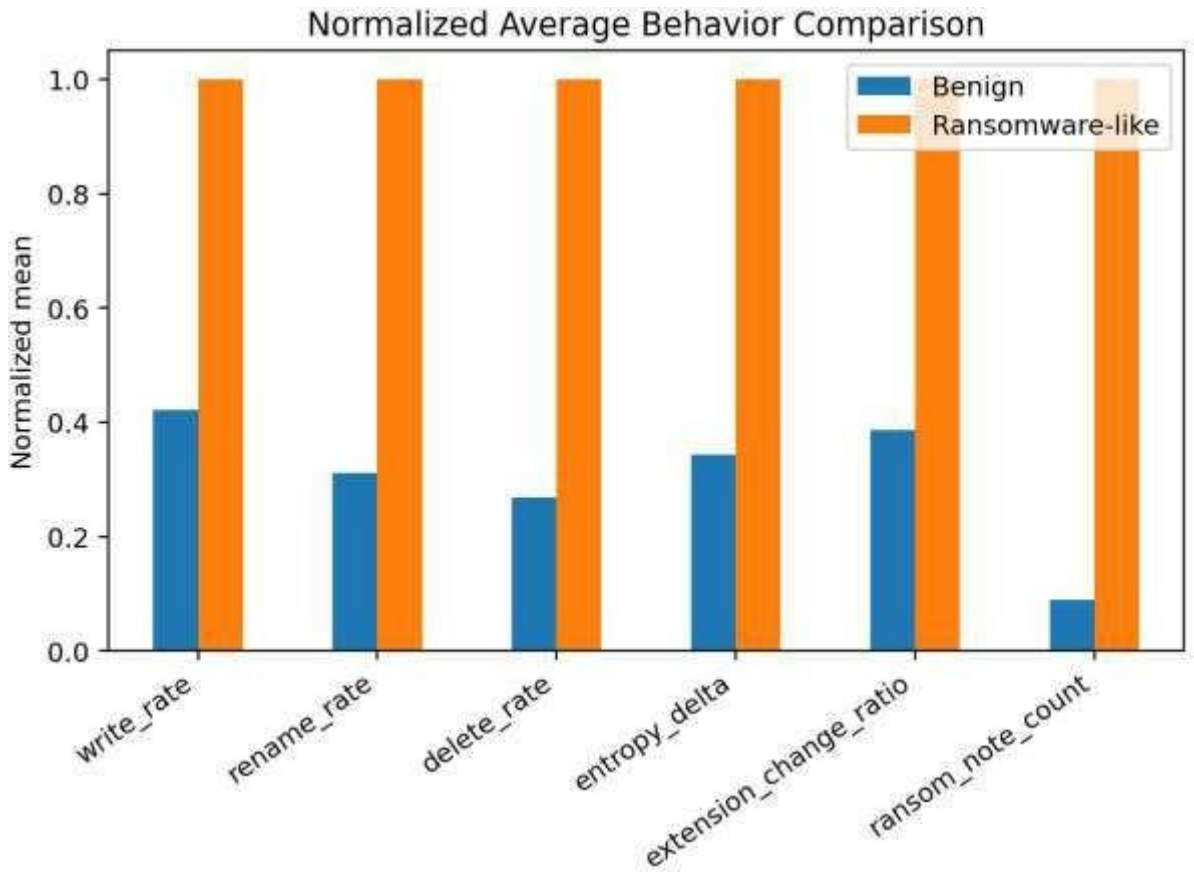

Fig. 5. Normalized average behavior comparison between benign and ransomware-like sessions.

IX. Discussion

The RF-RBD model illustrated above explains one way the AI can assist with early ransomware detection via behavioral monitoring. The system does not rely on identifying known malicious signatures as its primary mechanism. Instead, it identifies unusual file system activity as it pertains to encrypting, which helps detect novel or altered versions of ransomware that still display encryption behavior.

Another important consideration is explainability. While deep learning models can yield high accuracy, they tend to be difficult to explain. Random Forest balances performance with an element of explainability: security analysts may review the importance as well as the individual

risk level of a file operation to understand the basis of the alerted behavior, which should make an alerts system easier to use and act upon.

Early preventative action is another area that this model may assist with. An organization may wish to automatically cease synchronization, isolate an infected endpoint, revoke a user session, or gather evidence as soon as an alert is returned with a critical or high score. Nevertheless, these activities need to be balanced carefully with preventing impact to users' operations on valid machines, and thus blocking, or other defensive actions, should only be enacted automatically where confidence is high and consistent with organization policy.

Finally, it illustrates the need for a production ready data set. While the system here relies on a demonstration and explanation using synthetic data, in a real scenario the system will need to use real logs and use methods that account for privacy considerations and accurately labeled data and need to continually be monitored once deployed. Ransomware actors will likely alter their tactics to attempt to evade such detection, such as encrypting files very slowly or using common system utilities. This means that the system needs to be re-trained and validated regularly and continuously both before deployment and while in use.

X. Security, Ethics, and Safe Use

The research is purely defensive and informative, it does not feature the use of ransomware code or any encryption payloads, nor evasion techniques or damage instructions. In the provided code, the machine learning component is used solely for the purpose of classification, with artificial data being utilized. This is a crucial observation as it should be the duty of researchers to minimize damage, maximize security and promote the practice of responsible defensive efforts in the field of cybersecurity.

Data protection and governance issues arise from the monitoring of user and system actions. Institutions should establish very clear guidelines detailing what data is collected, what personnel may have access to this information, how long it should be stored for, and employees should be thoroughly informed about the monitoring practice. NIST's AI Risk Management Framework applies here, in the context that all AI-based systems need to be assessed to determine they have been properly tested for reliability, validity, security, accountability and

transparency. A ransomware detector should not be seen as a flawless system, but rather as a helpful resource for human-driven security.

One significant risk associated with this research is adversarial machine learning, whereby an attacker tries to elude the detector through modified behavior patterns, poisoned training data or normal administrative actions. The guidelines from NIST on adversarial machine learning state that attacks on an AI-based system can take place during training, deployment or inference. As such, training data for AI-based systems should be protected, and updates to the system verified. Model drift should also be observed and tested for evasive scenarios.

XI. Limitations

There are four limitations associated with this study. First, the data used is synthesized. Although the synthesized features are supposed to reflect typical ransomware-like behavior, real-world data can be much more complex and unpredictable. As a result, the performance measures here are indicative of the prototype level rather than production readiness. Second, this model uses session-level features of aggregated values, rather than time-series, and process ancestry, sensitive file path, user identity, device identity, network information would likely be required to provide additional features. Some ransomware strains may exhibit stealth and behave much slower than this model can account for. Third, false positives are unavoidable with this type of behavioral detection. For instance, backup routines, sync processes, archives, data movement, and software installations may share similar features with ransomware activity. This type of system would have to be supplemented with whitelisting of administrative processes and cross correlation of alerts with the trustworthiness of a given process's digital certificate, known signatures, or behavior on the endpoint. Finally, model drift may occur, as both user and system behavior evolve over time, such as the increasing use of remote, cloud based systems, installation of new applications, and retooling of business processes, this model would need to be regularly monitored and re-trained to ensure it continues to perform.

XII. Future Work

The proposed model should be validated against real or benchmarked ransomware data in the future work. The dynamic analysis data and server-side file operation log would further

strengthen the model to assess its performance in real-world. The comparison between Random Forest and other machine learning and deep learning models like XGBoost, LightGBM, SVM, and LSTM would be other directions for future work.

Another future work would be applying explainable AI like SHAP value to interpret the classification results. The feature importance can describe the global contribution of a feature whereas SHAP value could even provide record-level explanation. With SHAP value, analysts can tell which features contribute most to a specific session for predicting ransomware like traffic, making it easier for him/her to analyze alerts.

The model can also be integrated into real-time detection pipelines. With logs from end hosts or file servers streaming into SIEM system, the AI model would be called to score each time window and trigger response playbook when high risk alerts are generated. These response playbooks include pausing synchronization, isolating the affected host, and launching forensic evidence acquisition.

Finally, testing the proposed model against adversarial and realistic environments is worth conducting for future work. In that work, we can emulate the scenarios with slow encryption, small amount of file modification, benign backup noises, and data drifting to see if the model is resilient to the change of ransomware behavior over time.

.

XIII. Conclusion

In this paper we proposed RF-RBD, an explainable AI-based framework that uses file-system behavioral features and Random Forest classification to predict the presence of early ransomware activities. We used safe synthetic logs to show how certain behavior features (write rate, rename rate, delete rate, entropy change, extension-change ratio, ransom-note count, presence of shadow-copy commands, access to network shares) are indicative of ransomware behavior.

The tests performed yielded high results on the simulation environment for accuracy, recall, F1-score and ROC-AUC. Feature importance analysis further enhanced explainability of the results, ranking features by their contribution to model prediction. The discussed framework could be used for academic purposes or prototypes, but it's crucial to know that it cannot be

directly applied for production scenarios. Further development would require real data, governance framework for privacy, extensive security testing, drift detection, validation of results by security experts and so on.

In conclusion, this research demonstrated the potential of Artificial Intelligence in improving ransomware defense if applied carefully and used in combination with a holistic security strategy. AI systems for detection should supplement but not replace core security principles such as secure backups, patch management, identity and access management, principle of least privilege, user education, endpoint security, incident response capabilities and so on.

References

[1] Cybersecurity and Infrastructure Security Agency, "#StopRansomware Guide," CISA, 2023.

[2] MITRE ATT&CK, "Data Encrypted for Impact, Technique T1486," The MITRE Corporation, 2025.

[3] C. Pascoe, S. Quinn, and K. Scarfone, "The NIST Cybersecurity Framework (CSF) 2.0," National Institute of Standards and Technology, NIST CSWP 29, 2024.

[4] E. Tabassi, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," National Institute of Standards and Technology, NIST AI 100-1, 2023.

[5] J. A. Herrera-Silva and M. Hernández-Álvarez, "Dynamic Feature Dataset for Ransomware Detection Using Machine Learning Algorithms," Sensors, vol. 23, no. 3, 2023.

[6] M. A. Putrevu, H. Chunduri, V. S. C. Putrevu, and S. K. Shukla, "A Comprehensive Analysis of Machine Learning Based File Trap Selection Methods to Detect Crypto Ransomware," arXiv preprint arXiv:2409.11428, 2024.

[7] Q. Chen, S. R. Islam, H. Haswell, and R. A. Bridges, "Automated Ransomware Behavior Analysis: Pattern Extraction and Early Detection," in Proc. 2nd International Conference on Science of Cyber Security, 2019.

[8] C.-Y. Yang and R. Sahita, "Towards a Resilient Machine Learning Classifier: A Case Study of Ransomware Detection," arXiv, 2020.

[9] H. S. Anderson and P. Roth, "EMBER: An Open Dataset for Training Static PE Malware

Machine Learning Models," arXiv, 2018.

Computing Surveys, vol. 54, no. 5, pp. 1-36, 2021.