Topic: Detecting Ransomware Using Machine Learning:

Random Forest and Other Algorithms

Project2 Submitted

By: Adrian Magno

BOWIE STATE UNIVERSITY

CTEC402.001

Fall 2024

[**Introduction** 3](#_Toc178786070)

[**Methodology** 3](#_Toc178786071)

[Part 1: Detecting Ransomware with Random Forest 3](#_Toc178786072)

[Part 2: Using Other Machine Learning Algorithms 8](#_Toc178786073)

[Conclusion 9](#_Toc178786074)

#### **Introduction**

Ransomware has emerged as one of the most severe cyber threats, with attacks targeting individuals and organizations alike (Kharraz et al., 2016). These attacks often involve encrypting valuable data and demanding ransom for its release. Machine learning techniques have been increasingly employed to detect and mitigate such attacks by identifying patterns and anomalies in system behavior and network traffic (Sgandurra et al., 2016). This paper investigates the application of the Random Forest algorithm and other machine learning classifiers—namely Decision Tree, K-Nearest Neighbors (KNN), and Naive Bayes—to the detection of ransomware.

#### **Methodology**

This paper is divided into two parts. Part 1 focuses on detecting ransomware using the Random Forest algorithm, a popular ensemble learning method. Part 2 extends the study by applying three additional algorithms—Decision Tree, K-Nearest Neighbors (KNN), and Naive Bayes—to the same dataset, allowing us to compare their performance.

##### Part 1: Detecting Ransomware with Random Forest

Random Forest is an ensemble learning technique that builds multiple decision trees and aggregates their predictions to produce more accurate results. By averaging the predictions of individual trees, Random Forest reduces the risk of overfitting, providing a more generalized model (Breiman, 2001). This makes Random Forest particularly useful for complex classification tasks, such as distinguishing between benign data and ransomware. The dataset used in this study consists of two parts: one for benign data and the other for ransomware data. After loading and combining the datasets, we dropped irrelevant columns such as FileName and md5Hash, focusing on the features most indicative of ransomware behavior. The target variable, Benign, is used to label the data (1 for benign data and 0 for ransomware). The data was split into training and testing sets, with 80% of the data used for training and 20% reserved for testing. A Random Forest classifier was trained using different configurations of n\_estimators (10, 25, 50, 100). Accuracy and loss metrics were calculated for both the training and testing sets to evaluate the performance of the model. The results demonstrate that the Random Forest classifier achieves high accuracy in detecting ransomware, with improvements seen as the number of estimators increases.

**Results**

The Random Forest classifier was evaluated using four different configurations of n\_estimators, ranging from 10 to 100. Table 1 presents the accuracy and loss metrics for both the training and testing sets. As the number of estimators increases, we observe a corresponding improvement in both training and test accuracy. This indicates that Random Forest becomes more effective with a higher number of decision trees, which helps reduce the variance in predictions.

A screenshot of a computer

Description automatically generated

|  |
| --- |
| **Table 1: Performance of Random Forest with Various Configurations** |
| | **Random Forest Config** | **Train Accuracy** | **Test Accuracy** | **Train Loss** | **Test Loss** | | --- | --- | --- | --- | --- | | n\_estimators=10 | 0.9678 | 0.9556 | 0.0322 | 0.0444 | | n\_estimators=25 | 0.9705 | 0.9579 | 0.0295 | 0.0421 | | n\_estimators=50 | 0.9718 | 0.9600 | 0.0282 | 0.0400 | | n\_estimators=100 | 0.9731 | 0.9615 | 0.0269 | 0.0385 | |

In Figures 1 we visualize the accuracy and loss metrics for the Random Forest classifier. Figure 1 illustrates the training and validation accuracy, showing a clear improvement as more estimators are added. It also presents the loss metrics, where lower loss values indicate better model performance.

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

To further analyze the model’s performance, we generated confusion matrices for each configuration. The confusion matrix provides a detailed view of the model’s ability to classify benign data and ransomware correctly. Figures 2 through 5 show the confusion matrices for different configurations of Random Forest, with increasing accuracy for higher estimator values.

A blue squares with white text

Description automatically generatedA blue squares with white text

Description automatically generatedA blue squares with white text

Description automatically generatedA blue squares with white text

Description automatically generated

##### Part 2: Using Other Machine Learning Algorithms

In addition to Random Forest, we explored the performance of three other machine learning algorithms: Decision Tree, K-Nearest Neighbors (KNN), and Naive Bayes. Each of these models offers unique advantages and provides useful insights into the ransomware detection problem.

The Decision Tree algorithm builds a model by repeatedly splitting the dataset into branches based on feature values, ultimately creating a hierarchical structure of decisions. KNN is a non-parametric algorithm that classifies a data point by considering the majority label of its nearest neighbors, while Naive Bayes is a probabilistic classifier that applies Bayes’ theorem with strong independence assumptions between the features (Quinlan, 1986; Altman, 1992; McCallum & Nigam, 1998).

The same dataset used in Part 1 was employed here. We trained and evaluated the three algorithms using accuracy and loss metrics, as shown in Table 2.

A screenshot of a computer

Description automatically generated

**Table 2: Performance of Decision Tree, KNN, and Naive Bayes**

| **Algorithm** | **Train Accuracy** | **Test Accuracy** | **Train Loss** | **Test Loss** |
| --- | --- | --- | --- | --- |
| Decision Tree | 0.9965 | 0.9710 | 0.0035 | 0.0290 |
| KNN | 0.9732 | 0.9525 | 0.0268 | 0.0475 |
| Naive Bayes | 0.9601 | 0.9402 | 0.0399 | 0.0598 |
|  |  |  |  |  |

Figures 7 visualize the accuracy and loss metrics for Decision Tree, KNN, and Naive Bayes. As seen in Figure 7, Decision Tree outperforms both KNN and Naive Bayes in terms of accuracy, with the lowest loss values observed.

A screenshot of a graph

Description automatically generated

#### Conclusion

This paper demonstrates how machine learning algorithms, particularly Random Forest, Decision Tree, KNN, and Naive Bayes, can be employed to detect ransomware effectively. In Part 1, the Random Forest classifier achieved high accuracy and low loss, with better performance observed as the number of estimators increased. In Part 2, we found that Decision Tree outperformed KNN and Naive Bayes in terms of accuracy and loss. Overall, the results show that ensemble methods like Random Forest and tree-based models such as Decision Tree are well-suited for ransomware detection, although simpler algorithms like KNN and Naive Bayes can also provide valuable insights.

References

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician, 46*(3), 175–185. https://doi.org/10.2307/2685209

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Kharraz, A., Arshad, S., Mulliner, C., Robertson, W., & Kirda, E. (2016). UNVEIL: A large-scale, automated approach to detecting ransomware. In *Proceedings of the 25th USENIX Security Symposium* (pp. 757–772). USENIX Association. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/kharraz

McCallum, A., & Nigam, K. (1998). A comparison of event models for Naive Bayes text classification. In *Proceedings of the AAAI-98 workshop on learning for text categorization* (pp. 41–48). AAAI Press. https://www.aaai.org/Papers/Workshops/1998/WS-98-05/WS98-05-008.pdf

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106. https://doi.org/10.1023/A:1022643204877

Sgandurra, D., Muñoz-González, L., Mohsen, R., & Lupu, E. C. (2016). Automated dynamic analysis of ransomware: Benefits, limitations, and use for detection. *arXiv preprint arXiv:1609.03020*. <https://arxiv.org/abs/1609.03020>