Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Intelligent Behavioral Analysis for Detection of Ransomware Threats

Dr. Anumol Mathai
Associate Professor
Department of Computer Science and Engineering
G. Madegowda Institute of Technology
Bharathinagara, Karnataka, India

Ankitha HP | Pragathi DS | Sanika JM | Sanjay D
Department of Computer Science and Engineering
G. Madegowda Institute of Technology
Bharathinagara, Karnataka, India

ABSTRACT

Ransomware is a type of malware that can encrypt files, cause disruption to computers and cause monetary and data loss. Signature-based security solutions may not be able to detect newer, modified and/or obfuscated ransomware. This paper presents an Intelligent Ransomware Detection System (IRDS). The system uses machine learning techniques to classify ransomware types. It uses a binary classifier based on memory forensics, which used the Obfuscated-MalMem2022 dataset. A multi-class approach was also used to classify 5 different ransomware families. The system contains a real-time monitor for detecting anomalies in file system/process activity. An executable scanner was also included. The data was preprocessed and reduced using Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA). Random Forest was chosen as it was shown to perform better than other classifiers such as Decision Tree, Logistic Regression, Naive Bayes and Support Vector Machine. The memory forensic results showed Random Forest achieved a high accuracy of 99.97%. The results from the RFE/PCA experiment showed an accuracy of 99.98% with 6 dimensions. The results of the 5 family classification were 46.75% weighted accuracy. This shows there is more work to be done to improve this part of the system. A desktop application has been made to allow scanning, monitoring, alerting, scan history and producing a PDF report. The results show it is possible to create a prototype detection system. However, more validation should be done with a variety of different real-world data.

Keywords *- Ransomware detection, behavioral analysis, machine learning, Random Forest, RFE, PCA, real-time monitoring, malware classification.*

1. INTRODUCTION

Ransomware is malicious software used to lock a users computer or data. It commonly works by encrypting files and then demanding payment towards the recovery of these. It can effect many different types of users such as individuals, educational institutions, businesses, healthcare and financial institutions. If a ransomware attack is successful it could lead to data loss, service disruption, recovery costs and loss of reputation.

Most traditional anti-virus software will use signature based or pattern based detection techniques. While these can be very useful for detecting known threats they may struggle to detect ransomware that has been altered using packing, obfuscation or polymorphic techniques. Another technique that can be used is behaviour-based detection. This looks at what a program is doing such as creating, modifying and deleting files. It also looks at what processes are running and other changes within the system.

The objective of this project is to create an Intelligent Behavioural Analysis system using machine learning to detect ransomware. The system will allow early detection of suspicious activity, PCA and RFE will be used to select the most relevant features, classification of supported ransomware families, monitoring of behaviours within a system and alert generation when suspicious activity occurs.

2. LITERATURE SURVEY

The work done by Masum et al., used machine-learning techniques like Decision Tree, Random Forests, Naive Bayes, Logistic Regression and Neural Networks to classify ransomware. In their study they found Random Forests to be an effective classifier on the specific dataset. This shows how useful it can be to select the correct features and compare different models for ransomware classification [1].

Kunku, Zaman and Roy investigated ransomware detection and identification with Random Forest and XGBoost. This research looked at ransomware characteristics and machine learning techniques to identify malicious samples. Classification using datasets can form a good benchmark but detecting unknown variants and continual monitoring are still an issue [2].

Khoza, Komba and Mswikaneng investigated how ransomware can be defended against using a behaviour based approach with machine learning. Empirical research and monitoring is a key step when applying the proposed architecture to a working detection system [3].

Jalandra, Kulhia, and Kaur conducted an investigation into using a hybrid approach for android ransomware using several machine learning algorithms such as Random Forest, LightGBM, Support Vector Machine, Logistic Regression and Naive Bayes. This shows that multiple classifiers can be used but platform specific features such as URLs may not be indicative of all ransomware on desktops [4].

3. PROBLEM STATEMENT AND OBJECTIVES

Many ransomware detection approaches depend on known signatures or analyze a limited set of features. This can make it difficult to identify new variants and detect suspicious behavior before significant damage occurs. Binary detection alone also does not provide information about the likely ransomware family. There is therefore a need for a system that combines behavioral indicators, feature selection, classification, continuous monitoring, and timely alerts.

The main objectives of the proposed system are:

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

1. To develop an early-stage ransomware detection approach using behavioral analysis.
2. To apply Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA) to reduce irrelevant or redundant features.
3. To build a multi-class classification module for identifying supported ransomware families.
4. To develop a real-time monitoring module that observes system activities and generates alerts for suspicious behavior.

4. PROPOSED METHODOLOGY

The proposed system consists of inputting system activities, gathering behaviour data, pre-processing, selecting features, creating the models, classifying, real time monitoring and raising alerts. The desktop application will contain the models which have been trained and display the results from a scan and monitoring the files in the system.



Fig. 1. Workflow of the proposed behavioral ransomware detection system.

4.1 Dataset and Preprocessing

Binary memory forensics. For this experiment we used the Obfuscated-MalMem2022 data set. This has a total of 58,596 data points and 57 variables. We removed 534 duplicates to leave us with 58,062 data points. The target variable was removed from our input features to prevent target leaking. Our data was split into training and test sets using an 80/20 stratified split. Data preprocessing steps were applied to clean our data, manage missing values and prepare our numerical features for training our model.

4.2 Feature Selection

Recursive Feature Elimination is used for selecting useful input features. Principal Component Analysis is used to reduce these features to a reduced set of components. An example of this is shown below with the RFE and PCA reducing the features to 6 components. These transformations are fitted on the training data and applied to the test data.

4.3 Machine-Learning Model Training

A comparison of Decision Tree, Random Forest, Logistic Regression, Naive Bayes and Support Vector Machine classifiers was carried out. It was decided that Random Forest would be the primary binary classifier due to its performance during testing. The trained classifiers were then used to predict if the given data was benign or malware, plus the supported ransomware family within its own module.

4.4 Multi-Class Ransomware Classification

This module is used to determine which family (Ako, Conti, Maze, Pysa, or Shade) the ransomware belongs to. The output of this module can be seen as an extra classification step beyond just determining if a file is malicious or not. The output does depend on what features you input to the classifier. You should think of it as a prediction made by the model.

4.5 Real-Time Behavioral Monitoring and Alerts

This monitoring prototype watches for file creation, alteration and deletion in a specified folder. It will take into account any processes and system resources available. Events are extracted into behavioural features within a sliding window and fed into the monitoring model. If the risk threshold exceeds the specified value then an alert is shown in the application. The monitoring score indicates the risk level but does not indicate calibrated probabilities of infection.

4.6 Desktop Application

The interface is created using CustomTkinter for the desktop. It includes modules for detecting, scanning for executable files, monitoring, viewing predictions, storing history of scans, viewing data within a dashboard and creating reports in PDF format. This application is used as a prototype to show the process and results.

4.7 Mathematical Formulation

Following are the mathematical expressions used for preprocessing, transforming features, classification of data and model evaluation.

4.7.1 Min–Max Normalization

$$x' = (x - x_min) \,/\, (x_max - x_min) \quad (1)$$

Min-Max normalization: This scales values of a numerical feature between 0 and 1. Where x is the original feature value, x min is the minimum value and x max is the maximum value. When dealing with a constant feature we need to handle it separately as dividing by zero would occur.

4.7.2 Principal Component Analysis (PCA)

$$z = W^{T}(x - \mu) \quad (2)$$

PCA projects the centred input vector to reduce the dimensionality of the data. Where x is the input feature vector, μ is the mean vector calculated using the training data, W are the principal component directions we keep, and z is the projected feature vector.

4.7.3 Random Forest Classification

$$\hat{y} = \text{mode}\ \{\, h_1(x), h_2(x), ..., h_T(x) \,\} \quad (3)$$

Random Forest combines all the predictions of each decision tree in T. Where h_t(x) is the predicted class by the t-th tree, mode finds the class that has the most votes. The final prediction $\hat{y}$ is the predicted class.

4.7.4 Accuracy

$$\text{Accuracy} = (TP + TN) \,/\, (TP + TN + FP + FN) \quad (4)$$

Accuracy is the ratio of correct predictions.

4.7.5 Precision

$$\text{Precision} = TP \,/\, (TP + FP) \quad (5)$$

Precision is the ratio of positive predictions that are actually positive.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

4.7.6 Recall

$$\text{Recall} = TP / (TP + FN) \quad (6)$$

Recall is the ratio of actual positives that are correctly identified by the classifier.

4.7.7 F1-Score

$$\text{F1-score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (7)$$

F1 score: is the harmonic average of precision and recall. TP: true positive. TN: true negative. FP: false positive. FN: false negative.

To ensure that our evaluation is reliable we have to fit the Min-Max Scaler and PCA on our training data. After that we can transform our test data with these transformations.

5. RESULTS AND DISCUSSION

5.1 Application Interface and Dashboard

The desktop gives us a central hub to view our scanning activity and includes summary cards showing how many total scans have been made, how many were malicious, benign, high risk, critical, alerts, early warning alerts, average early detection score, alerts raised and family candidates. As we haven't made any scans yet these values will all be 0. It also includes several quick links to look at files, the live monitor, history, create a report, view graphs, settings.

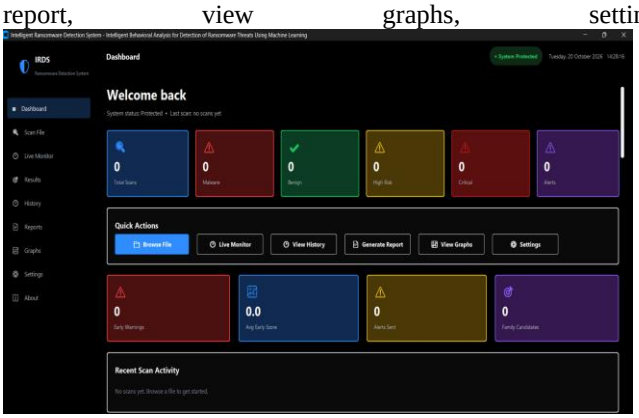


Fig. 2. IRDS application dashboard and quick-access functions.

5.2 Comparative Machine-Learning Model Performance

The model performance screen shows how well each model (Decision Tree, Random Forest, Logistic Regression, Naïve Bayes, Support Vector Machine) performed on the test set. From the charts we can see that the accuracy scores are 99.95%, 99.97%, 99.54%, 99.62% and 99.94%. Precision scores are 99.95%, 99.97%, 99.53%, 99.50% and 99.95%. Recall scores are 99.95%, 99.98%, 99.55%, 99.74%, 99.93%. Finally F1-scores are 99.95%, 99.97%, 99.54%, 99.62% and 99.94%. It can be seen that the Random Forest model had the highest accuracy, F1 score and recall score and so that has been chosen as our model, it can be seen in the application. This does not mean that this model will perform exactly the same on unseen real world ransomware data, this just means that these were the results with the held out dataset..

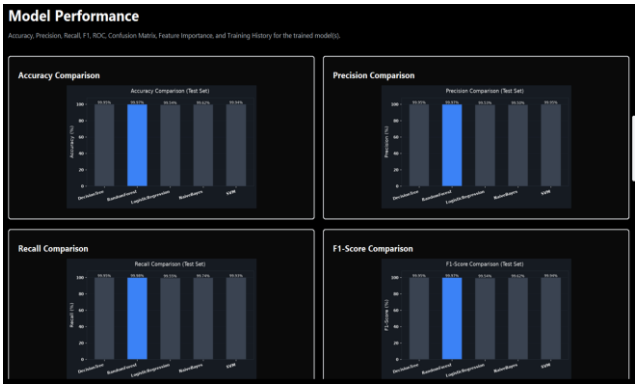


Fig. 3. Test-set comparison of accuracy, precision, recall, and F1-score.

Model	Accuracy	Precision	Recall	F1-score
Decision Tree	99.95%	99.95%	99.95%	99.95%
Random Forest	99.97%	99.97%	99.98%	99.97%
Logistic Regression	99.54%	99.53%	99.55%	99.54%
Naïve Bayes	99.62%	99.50%	99.74%	99.62%
SVM	99.94%	99.95%	99.93%	99.94%

5.3 Live Behavioral Monitoring and Safe Test Simulation

This is the monitoring screen. It monitors a user selected folder and shows any file system activity. This captures the events that occur and gives a count of how many times a file was changed every 10 seconds along with the early detection score, family match, last event and warnings. There are behavioral indicators like how many files were changed, created, deleted per second and the burst score. In this case the monitoring screen has captured events of files being created and deleted as part of the safe test simulation. However, it is clear to see the message at the bottom states that no ransomware was actually ran and that the test had not altered any files. This shows the monitoring screen and how the events are captured.

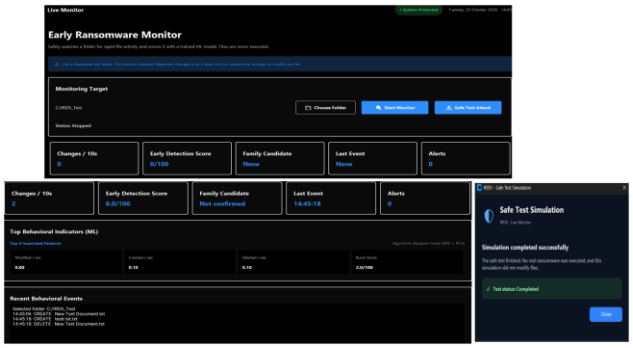


Fig. 4. Live monitor interface showing file activity and safe-test simulation.

5.4 File-Scan Demonstration and Interpretation

The file scanning screen allows users to select an executable file from their system. In this case a file named harmless.exe was used. Here you can see the results, it has been identified as a 'BENIGN' and 'LOW RISK' file. The results show a confidence level of 27.00%, model accuracy of 99.97%, using the Random Forest algorithm. The prediction took 612.85ms. It should be noted that the model accuracy shows how well the model performed when tested on the test set. The Early detection score shows how risky the file is, using a scale from 0-100.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

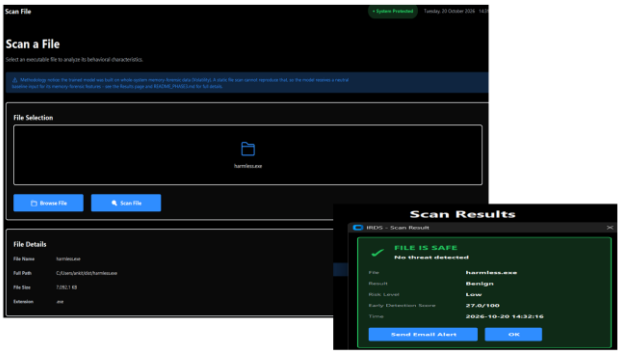


Fig. 5. Executable-file selection screen used for the demonstration.

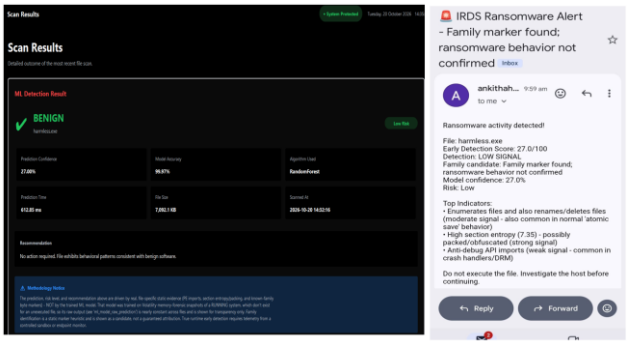


Fig. 6. Example scan result and email notification for the selected benign file.

A family indicator was found, however there was no confirmation of ransomware activity. A low risk indicator was found. File listing, renaming and deleting activities were observed. High section entropy and anti-debug API imports were also identified. This is heuristic evidence of ransomware activity. However, these indicators alone do not confirm ransomware execution. The email has more serious sounding language (Ransomware Alert / Ransomware activity detected!). It would be better to reflect the actual situation to avoid any confusion. For instance: 'Low Risk Scan Notice: Suspicious Family Indicator Found. No confirmation of ransomware activity.'

There is an issue with our methodology here. The scanning page has been created using the trained Random Forest model that was trained on the global memory forensic data from Volatility. A singular static executable will not have the same properties as the memory. The application provides a neutral base for the memory forensic model to compare against for the scanning of files. The detection from the static file and risk alert should not be considered as execution testing. This is demonstrated by having real time behaviour detection in the folder monitoring. This example shows that the UI and reports work with the chosen executable but doesn't indicate that the chosen executable was run dynamically or ransomware detected in real time.

5.5 Summary of Demonstration Results

The screenshots show an overview of the Desktop Dashboard, comparing metrics between models, real time monitoring of files, simulated testing safely, selecting executable files, showing results of a scan and emailing. The highest metric that can be seen in the model performance is the Random Forest model has 99.97% accuracy, 99.98% recall and 99.97% F1 score. The example provided (harmless.exe) is considered benign with a low-risk rating of 27/100. The safe-test monitoring process logs files but does not execute ransomware. This should be noted separately due to being different components and evidence. This example has evaluated the dataset classifier, provided an example of monitoring in controlled conditions and provided heuristic/static indicators from files.

The binary memory forensics experiment was performed using 80:20 stratified sampling of the cleaned Obfuscated-MalMem2022 dataset. The Random Forest classifier produced the highest accuracy and was used as the primary binary model.

Classifier	Accuracy	Precision	Recall	F1-score	ROC-AUC
Decision Tree	99.95%	99.95%	99.95%	99.95%	99.95%
Random Forest	99.97%	99.97%	99.98%	99.97%	100.00%
Logistic Regression	99.54%	99.53%	99.55%	99.54%	99.99%
Naive Bayes	99.62%	99.50%	99.74%	99.62%	99.98%
SVM (RBF)	99.94%	99.95%	99.93%	99.94%	100.00%

Table 1. Reported binary memory-forensics classification results

Using RFE and PCA together in feature selection achieved an accuracy and F1 score of 99.98%. Using mutual information, RFE alone or PCA alone gave an accuracy of 100.00%, 100.00% and 99.97% respectively. RFE + PCA gives a simple 6D feature representation. The results in this experiment and run-time should be verified against final log files.

Feature-selection method	Accuracy	F1-score	Output dimensions
RFE + PCA	99.98%	99.98%	6
Mutual Information	100.00%	100.00%	25
PCA only	99.97%	99.97%	25
RFE only	100.00%	100.00%	25

Table 2. Reported feature-selection comparison

The 5 class classifier achieved a weighted accuracy of 46.75%, indicating that further improvements are required to achieve acceptable performance for classification at family level. The behaviour monitoring prototype was run in real-time, however the prototype/static risk score does not show that the detection is valid against real ransomware attacks. This was due to the use of prototypes or synthetic inputs.

6. FUTURE SCOPE

1 Use a wider variety of data to increase the accuracy of ransomware family classification.

2 Add more features from the systems and processes to improve real-time behaviour monitoring.

3. Test the detection framework using independent real-world data and with new ransomware variants not seen before.

4. Tune thresholds and evaluate the model further to get more accurate alerts and reduce false positives.

5. Implement automated actions such as quarantining suspicious processes and limiting access to impacted files. After proper safety testing.

7. CONCLUSION

This paper details the development of an intelligent ransomware detection system. This uses classification based on machine learning, feature selection, family detection, real time

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

monitoring and alerting. The binary classifier used with Random Forest showed high performance with an accuracy of 99.97%. The RFE-PCA experiment gave a high accuracy of 99.98% using 6 dimensions. However, family detection did not perform as well. The monitoring capabilities of the prototype need to be tested in real scenarios. Overall this demonstrates how machine learning can be applied with behavioural analysis to detect ransomware. Further testing is required before deploying in real scenarios.

ACKNOWLEDGMENT

We would like to express our sincere thanks to Dr. Anumol Mathai, Assistant Professor, Department of Computer Science and Engineering, G Madegowda Institute of Technology for her guidance and support during this work. We would also like to thank the Department and Institute for providing us with the necessary facilities to complete this work.

REFERENCES

[1] M. Masum, M. J. H. Faruk, H. Shahriar, K. Qian, D. Lo, and M. I. Adnan, "Ransomware Classification and Detection With Machine Learning Algorithms," in *Proc. 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, Jan. 2022, pp. 316–322, doi: 10.1109/CCWC54503.2022.9720869.

[2] K. Kunku, A. N. K. Zaman, and K. Roy, "Ransomware Detection and Classification using Machine Learning," in *Proc. 2023 IEEE Symposium Series on Computational Intelligence (SSCI)*,2023,pp.862866,doi:10.1109/SSCI52147.2023.10371924.

[3] C. Khoza, G. M. Komba, and S. P. Maswikaneng, "An Enhanced Ransomware Defense Strategy through Behavior-Based Detection using Machine Learning Algorithms," in *Proc. 2024 IEEE 8th Conference on Energy Internet and Energy System Integration(EI2)*,Nov.2024,doi:10.1109/EI264398.2024.10991551.

[4] M. Jalandra, M. Kuliha, and J. Kaur, "Hybrid Machine Learning Model for Android Ransomware Detection Using URL-Based Features," *International Journal on Science and Technology (IJSAT)*, vol. 16, no. 3, Jul.–Sep. 2025, doi: 10.71097/IJSAT.v16.i3.7324.

[5] C.-M. Hsu, C.-C. Yang, H.-H. Cheng, P. E. Setiasabda, and J.-S. Leu, "Enhancing File Entropy Analysis to Improve Machine Learning Detection Rate of Ransomware," *IEEE Access*, vol. 9, pp. 138345–138351, 2021, doi: 10.1109/ACCESS.2021.3114148.

[6] H. I. Daku, P. Zavarsky, and Y. Malik, "Behavioral-Based Classification and Identification of Ransomware Variants Using Machine Learning," Master of Information Systems Security Management Project Report, Concordia University of Edmonton, Canada, Apr. 2018.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)