Contents lists available at GrowingScience

International Journal of Data and Network Science

homepage: www.GrowingScience.com/ijds

A fine-tuning of decision tree classifier for ransomware detection based on memory data

Mosleh M. Abualhaj[a*], Mahran Al-Zyoud[a], Mohammad O. Hiari[a], Yousef Alrabanah[b], Mohammed Anbar[b], Amal Amer[b] and Ali Al-Allawee[c]

[a]*Department of Networks and Cybersecurity, Al-Ahliyya Amman University, Amman, Jordan*
[b]*Department of Data Science and Artificial Intelligence, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan*
[c]*Department Computer Science, University of Mosul, Mosul, Iraq*

C H R O N I C L E

Article history:
Received: November 7, 2023
Received in revised format: November 30, 2023
Accepted: January 7, 2023
Available online: January 7, 2024

Keywords:
Ransomware
Machine Learning
DT

A B S T R A C T

Ransomware has evolved into a pervasive and extremely disruptive cybersecurity threat, causing substantial operational and financial damage to individuals and businesses. This article explores the critical domain of Ransomware detection and employs Machine Learning (ML) classifiers, particularly Decision Tree (DT), for Ransomware detection. The article also delves into the usefulness of DT in identifying Ransomware attacks, leveraging the innate ability of DT to recognize complex patterns within datasets. Instead of merely introducing DT as a detection method, we adopt a comprehensive approach, emphasizing the importance of fine-tuning DT hyperparameters. The optimization of these parameters is essential for maximizing the DT capability to identify Ransomware threats accurately. The obfuscated-MalMem2022 dataset, which is well-known for its extensive and challenging Ransomware-related data, was utilized to evaluate the effectiveness of DT in detecting Ransomware. The implementation uses the versatile Python programming language, renowned for its efficiency and adaptability in data analysis and ML tasks. Notably, the DT classifier consistently outperforms other classifiers in Ransomware detection, including K-Nearest Neighbors, Gradient Boosting Tree, Naive Bayes, and Linear Support Vector Classifier. For instance, the DT demonstrated exceptional effectiveness in distinguishing between Ransomware and benign data, as evidenced by its remarkable accuracy of 99.97%.

1. Introduction

Cybersecurity encompasses a range of problems about safeguarding computer systems, networks, devices, and data against potential cyberattacks and associated dangers. The problems mentioned above hold considerable importance in the contemporary era of digital advancements, given the escalating integration of technology into various facets of human existence (Lei et al., 2022; Alves et al., 2018; Jain et al., 2022). Malware constitutes a prominent source of cyberattacks. Malware refers to a wide range of harmful software that hackers use with malicious intent. Malware can infiltrate various digital devices, including computers, cell phones, servers, and other devices. This pervasive presence of malware poses substantial risks and potential harm to individuals, companies, and governments on a global scale. Common types of malware include Viruses, Adware, Spyware, and Ransomware (Peng et al., 2014; Sen et al., 2018; Sonicwall, 2022).

Ransomware is malware designed to encrypt a victim's data or lock them out of their computer or digital files until a ransom is paid to the attacker. It is a form of extortion where cybercriminals demand payment from individuals, businesses, or organizations to restore access to encrypted information. Ransomware attacks have become a substantial cybersecurity threat, causing financial losses and disruptions to individuals and enterprises (Molina et al., 2022; Saurabh, 2018). In 2021, Ransomware

* Corresponding author.
E-mail address: m.abualhaj@ammanu.edu.jo (M. M. Abualhaj)

ISSN 2561-8156 (Online) - ISSN 2561-8148 (Print)
© 2024 by the authors; licensee Growing Science, Canada.
doi: 10.5267/j.ijdns.2024.1.005

climbed an unprecedented 105%, and in 2022, 623.3 million Ransomware attacks have occurred (Sonicwall, 2022). Mitigating Ransomware involves implementing preventive measures and response strategies to reduce the risk of Ransomware attacks and minimize their impact if they occur. Some of the effective Ransomware mitigation strategies are regular data backups, patch management, the use of anti-malware software, and the use of behavior-based detection (Molina et al., 2022; Saurabh, 2018).

Behavior-based detection, also known as behavioral analysis or heuristics-based detection, is a cybersecurity approach that focuses on identifying and blocking malicious activities based on their behavior patterns rather than relying solely on known signatures or patterns of malware. Unlike traditional signature-based detection that looks for exact matches with known malware, behavior-based detection is designed to catch new and previously unseen threats, including zero-day exploits and polymorphic malware. Behavior-based detection often leverages Machine Learning (ML) mechanisms to analyze large amounts of information and recognize suspicious activities (Wang & Zhu, 2022; Firdausi et al., 2010; Parizad & Hatziadoniu, 2022).

ML aims to improve algorithms and statistical models so that computers can learn new things and get better at what they do without being explicitly programmed. The main goal of ML is to make it possible for computers to find patterns and make decisions or predictions on their own instead of needing clear instructions. The topic of ML is characterized by its rapid evolution, encompassing a diverse array of applications and considerable potential. As the amount of data available and the power of computers keep improving, ML is likely to make a big difference in cybersecurity, especially when finding Ransomware (Kolhar et al., 2020; Westyarian et al., 2015; Yeo et al., 2018).

Supervised learning (SL) is a sort of ML where the method learns from labeled data, where each training example has input features and matching output labels. The goal of SL is to learn and find a relationship between the available features and the output labels so that the method can make precise forecasts on new, unseen data. Some common supervised learning algorithms are K-Nearest Neighbors (KNN), Linear SVC (Support Vector Classifier), Naive Bayes (NB), Gradient Boosting Tree (GB Tree), and Decision Tree (DT) (Yeo et al., 2018; Abualhaj et al., 2022; Choudhary & Sharma, 2020; Chen, Su, Lee, & Bair, 2020). DT algorithm will be used in this paper to detect Ransomware.

This paper contains the following sections: Section 2 introduces some of the work regarding the detection of Ransomware. Section 3 discusses the proposed model of Ransomware detection, including the DT classifier and its key hyperparameters. Section 4 shows the performance of DT in Ransomware detection. Finally, Section 5 presents the main findings of this paper.

2. Related works

Almashhadani et al. (2019) tackle the issue of Ransomware attacks, which often involve establishing connections to C&C servers before executing their destructive payloads. To detect such attacks early, the researchers adopt network-based methodologies. They meticulously analyze Ransomware network data, focusing on Locky in the study. This approach allows them to identify and counter such attacks proactively. The authors set up a specialized testbed environment and thoroughly examined PCAP files, including Locky's PCAP files from the MCFP dataset. Various network behaviors associated with Locky were uncovered through investigation, resulting in 18 potential behavioral elements extracted from HTTP, TCP, NBNS, and DNS data. These extracted characteristics prove enlightening and effective in recognizing data from compromised hosts. A multi-classifier network-based Ransomware detection system that operates at both the packet and flow levels was proposed to enhance detection capabilities. Extensive experimental evaluations demonstrate the efficiency of the derived features, achieving high detection accuracy rates of 97.92% and 97.08%, respectively, for each detection level.

Alqahtani et al. (2020) address the problems of tracing the initial call of any Application Programming Interface (API) connected to cryptography to identify Crypto-ransomware. The Crypto-Ransomware Early Detection (CRED) model, put forth by Alqahtani et al., is an early detection methodology that can more precisely identify the pre-encryption boundaries and gather the necessary data. The model uses the temporal connection between the I/O Request Packets (IRPs) and the APIs to identify the pre-encryption stage boundaries of the crypto-ransomware data. It then extracts the properties associated with the pre-encryption stage of the Crypto-ransomware lifecycle, using this boundary as a threshold. These are then used to train the LSTM algorithm to create the early detection model. When the pre-encryption boundaries are precisely defined, the model can extract the features associated with that stage and overcome the data insufficiency throughout this stage. More attack patterns are detected when data-centric and process-centric techniques are included in the detection model. This improves the CRED's capacity to recognize crypto-ransomware assaults before the encryption begins. A data benchmark from running real-world crypto-ransomware samples taken from a popular source will be used to assess the CRED model.

Manavi and Hamzeh (2021) present a technique for spotting Ransomware based on the headers of executable files. These headers contain critical data about the program's structure as a byte sequence. Altering this header information leads to a different structure for the program file. The suggested solution employs a Long Short-Term Memory (LSTM) network to analyze the byte sequence that constitutes the header, effectively distinguishing Ransomware samples from benign ones. Notably, the method achieves high accuracy in Ransomware detection without the need for time-consuming preprocessing and feature extraction, surpassing other methods in terms of overall detection rate. With an impressive accuracy of 93.25% in detecting Ransomware samples using only the raw header and without executing the program, the proposed method is well-suited for swiftly detecting suspicious samples.

Deng et al. (2022) propose a novel approach for classifying Ransomware using the entropy map found in the Ransomware binary file. This method leverages the entropy map to capture more precise characteristics within the Ransomware family, improving classification accuracy. They introduce a data augmentation technique depending on the Cycle-GAN network, integrating fine-tuning technology from transfer learning to further enhance the framework's classification results and address data imbalances among Ransomware families. Simultaneously, they incorporate an attention mechanism into VGG-16 to improve the network's feature extraction capabilities. Experimental results demonstrate that their suggested technique attains the highest performance across 14 Ransomware families, with an impressive accuracy rate of 97.16%, surpassing other standard Ransomware visualization and classification methods. Compared to other neural networks, the suggested technique exhibits the maximum level of classification performance.

As part of their research, Rakshit et al. (2019) thoroughly examined ransomware executables, aiming to uncover structural characteristics that ML systems could harness effectively. Their investigation unveiled the presence of recurrent patterns within extended sequences of ransomware, suggesting the repetition of encryption procedures. The researchers introduced an innovative component into the recurrent neural network to leverage these recurring patterns. This component incorporates attention mechanisms into the input sequences of a sequence learning module. The 'Attended Recent Inputs (ARI)-Long Short-Term Memory (LSTM)' represents a modified version of the LSTM architecture. Empirical analysis of a ransomware dataset revealed that the ARI-LSTM outperforms traditional LSTM models significantly, particularly in the domain of malware detection. For instance, the ARI-LSTM achieved an impressive 93% accuracy in ransomware detection. The introduction of the ARI cell, which integrates attention into sequence inputs, holds promise for tasks that require recognizing connections within recent inputs.

3. Proposed Ransomware Attack Detection Model

The suggested Ransomware attack detection model will be discussed in this section. First, the Obfuscated-MalMem2022 dataset, which has been used in the proposed model, will be presented. Then, the steps that will be performed on the used dataset to prepare it for training and testing the proposed model will be detailed. Finally, the DT ML classifier that will be used to detect the Ransomware attack will be introduced. Fig. 1 shows the operations performed to detect the Ransomware attack by the DT classifier.

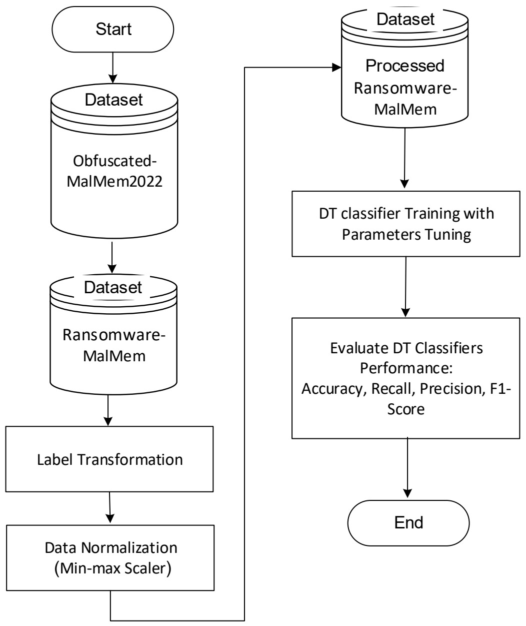

Start
Dataset
Obfuscated-MalMem2022
Dataset
Ransomware-MalMem
Label Transformation
Data Normalization (Min-max Scaler)
Dataset
Processed Ransomware-MalMem
DT classifier Training with Parameters Tuning
Evaluate DT Classifiers Performance: Accuracy, Recall, Precision, F1-Score
End

Fig. 1. Ransomware Detection Model

3.1 Obfuscated-MalMem2022 dataset

The Obfuscated-MalMem2022 dataset contains three main families of malware: Trojan Horse, Spyware, and Ransomware. The focus of this work is only on Ransomware. Therefore, all Trojan Horse and Spyware samples have been removed. The resulting dataset is called Ransomware-MalMem. The Ransomware-MalMem dataset contains 9791 samples distributed over five types of Ransomware: Conti (1988 samples), MAZE (1958 samples), Pysa (1717 samples), Ako (2000 samples), and

Shade (2128 samples) (Dener, Ok, & Orman, 2022). Fig. 2 clarifies the Ransomware types distribution. Besides, the Ransomware-MalMem dataset contains 29298 samples of benign data.

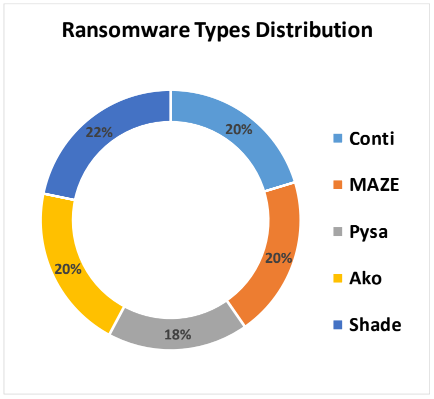


Fig. 2. Ransomware types of distribution

3.2 Ransomware-MalMem Preprocessing

In ML, data preprocessing plays a vital role as it ensures that the data is appropriately prepared for training, ultimately enhancing the performance of ML models. Data transformation and normalization are key processes among the crucial steps in data preprocessing. These steps help to make the data more suitable for training and enable the models to achieve better accuracy and generalization. A data transformation step is crucial to rendeing data more suitable for ML algorithms. This process involves converting categorical data into numerical values, often achieved through a technique known as label encoding. In ML algorithms, categorical variables are often represented as integers to facilitate processing. Each category within the categorical variable is assigned a unique integer value through label encoding (Abualhaj et al., 2022; Al-Mimi et al., 2023). In the context of the Ransomware-MalMem dataset, the categorical variable exists only in the output column, comprising six categories: Conti, MAZE, Pysa, Ako, Shade, and benign (Dener, Ok, & Orman, 2022). Label encoding translates these categories into the respective integers: 1, 2, 3, 4, 5, and 0. This ensures that the ML model can effectively utilize categorical data during training.

Data normalization is crucial in ML to prevent certain features from overpowering the model due to their high magnitudes. It involves adjusting the dataset's features to a common range. A popular data normalization method in ML is the Min-Max Scaler, which scales numerical features within a predetermined range, typically between 0 and 1. Normalization aims to bring all features to the same magnitude, ensuring that no single feature dominates the learning process due to its larger magnitude. By normalizing the data, the ML model can effectively consider all features equally, leading to better performance and more accurate predictions (Abualhaj et al., 2022; Al-Mimi et al., 2023). Table 1 and 2 show samples of the Ransomware-MalMem dataset before and after preprocessing, respectively.

Table 1
Sample of the Ransomware-MalMem dataset before preprocessing

Data Samples	Output
45, 17, 10.55555556, 0, 202.8444444, 1694, 38.5	Benign
47, 19, 11.53191489, 0, 242.2340426, 2074, 44.12765957	Benign
40, 14, 14.725, 0, 288.225, 1932, 48.3	Benign
45, 17, 10.55555556, 0, 202.8444444, 1694, 38.5	Ransomware
47, 19, 11.53191489, 0, 242.2340426, 2074, 44.12765957	Ransomware
40, 14, 14.725, 0, 288.225, 1932, 48.3	Ransomware

Table 2
Sample of the Ransomware-MalMem dataset after preprocessing

Data Samples	Output
0.138554217, 0.473684211, 0.590188649, 0, 0.006397056, 0.340008522, 0.670600093	0
0.15060241, 0.578947368, 0.657499083, 0.007985233, 0.501917341, 0.796957154	0
0.108433735, 0.315789474, 0.877631096, 0, 0.009839575, 0.441414572, 0.890638135	0
0.102409639, 0.368421053, 0.64911192, 0, 0.007112541, 0.28376651, 0.705430814, 0.003483515	1
0.090361446, 0.368421053, 0.561204769, 0, 0.006872998, 0.233915637, 0.683040191, 0.002849364	1
0.102409639, 0.368421053, 0.603151768, 0, 0.006984345, 0.279079676, 0.699097955, 0.003364552	1

3.3 DT for Ransomware detection

DTs are a well-known supervised ML algorithm widely utilized in various fields, including cybersecurity, due to their simplicity, interpretability, and effectiveness. In cybersecurity, DT classifiers find valuable applications in intrusion detection, email and spam filtering, malware detection, and more. In this paper, we focus on their role in the crucial task of Ransomware detection. There are several advantages to employing DTs in Ransomware detection. Firstly, they provide a transparent decision-making process, enabling security analysts to comprehend and visualize how the classifier arrives at its conclusions. This interpretability is invaluable in diagnosing false positives and false negatives. Secondly, DT classifiers exhibit relatively low computational overhead, making them well-suited for real-time or near-real-time Ransomware detection. Their efficiency is crucial in swiftly identifying potential threats. Moreover, DTs enable security experts to pinpoint the most indicative features of executables associated with Ransomware. This insight informs the development of more effective detection techniques and signatures, enhancing overall cybersecurity. Furthermore, DTs can effectively handle many features and samples without significantly impacting performance. This scalability is essential in managing the vast and evolving landscape of potential threats. Lastly, properly trained DTs can accomplish a low false positive rate, reducing the likelihood of legitimate software being falsely flagged as Ransomware. This accuracy is pivotal in maintaining systems and data integrity (Razali et al., 2022; Saurav et al., 2023; Vijayarangam et al., 2021).

A DT is a hierarchical structure composed of nodes. DT classifiers operate by iteratively dividing the dataset into subsets according to the values of the input features, eventually each subset is given a class label. The tree's journey begins with a single node called the "root node. Each internal node signifies a decision based on one of the input features, while each leaf node symbolizes a specific class label or outcome. The connections or edges between nodes illustrate the outcomes of the decisions made along the tree's branches (Abualhaj et al., 2022; Vijayarangam et al., 2021). Fig. 3 clarifies the DT classifier.

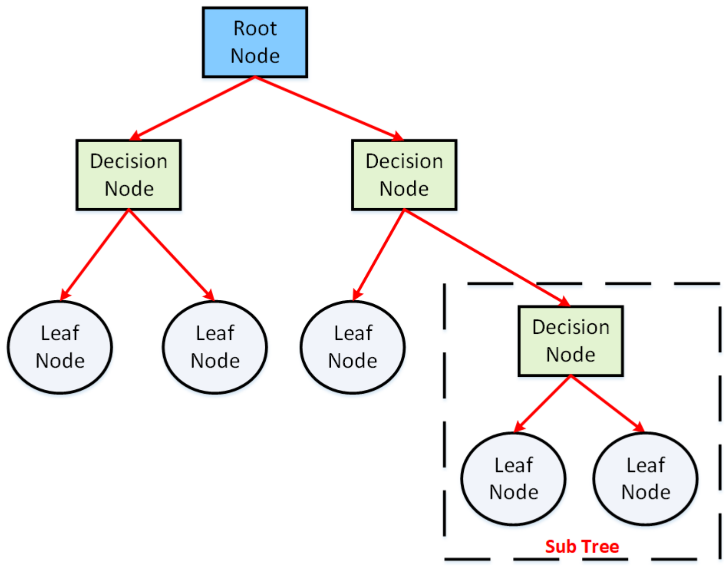


Fig. 3. DT classifier

To detect Ransomware, a DT involves several key operations executed sequentially. These operations are carried out during both the training and prediction phases. Five main steps are performed during the training phase. First, all available features of the Ransomware-MalMem dataset are evaluated to determine which one provides the best split. The selected feature is the one that maximizes information gain (or minimizes impurity) based on a chosen criterion, such as Gini impurity or entropy. Second, internal nodes are created based on the selected feature. Each internal node symbolizes a decision point based on a particular feature. Third, the Ransomware-MalMem dataset is split into two subsets based on the decision made at the internal node. Fourth, the previous operations are recursively applied to the subsets created by the split till building the entire tree (e.g., reaching a maximum tree depth). Fifth, a leaf node (Benign or Ransomware) is created once building the tree is completed. Three main steps are performed during the prediction stage. First, to classify a new data point, the DT algorithm starts at the tree's root node and evaluates the decision rules at each node to decide which child (left or right child) should be chosen. Second, the traversal continues until the Benign or Ransomware leaf node is reached. Third, Benign or Ransomware is assigned as the predicted class for the input data point based on the value associated with the reached leaf node (Vijayarangam et al., 2021; Ma et al., 2018; Zulfikar et al., 2018; MahendraVardhan & Sridhar, 2022). Fig. 4 shows the steps performed by the DT classifier.

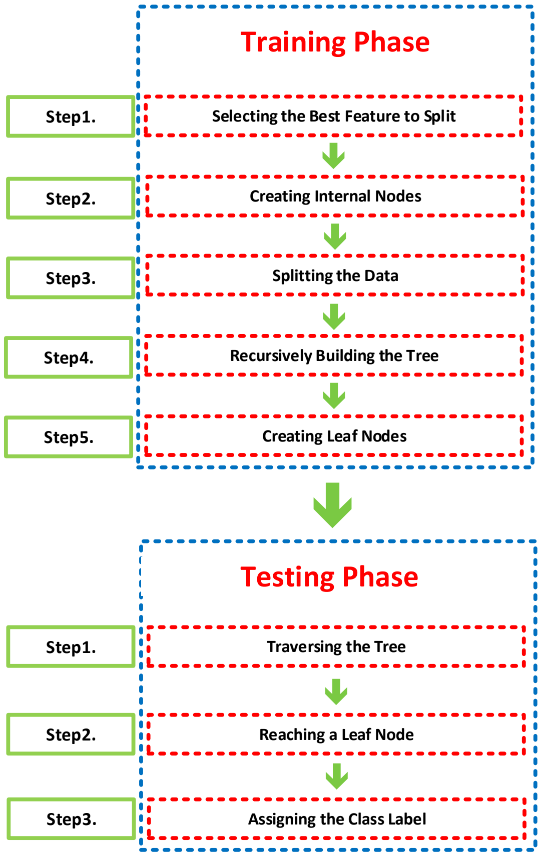


Fig. 4. Training and Testing of DT classifiers

One critical aspect of using DTs for Ransomware detection is the selection of hyperparameters. Hyperparameters are configurations that can be fine-tuned during the training of a DT classifier to optimize its performance or control its behavior. Making the right choices regarding hyperparameters is crucial for avoiding overfitting and achieving optimal results in Ransomware detection. One of the key hyperparameters to consider is the 'Criterion.' This parameter defines the quality measure used to assess the goodness of a split at each node. The most used criteria are 'gini' and 'entropy.' Another significant hyperparameter is 'Maximum Depth,' which determines the maximum depth of the DT. It can take various values, from 'None' (indicating unlimited depth) to integers such as 10, 20, or 30. 'None' implies that nodes keep growing till all leaves are pure or contain less samples than the determined 'Minimum Samples for Splitting.' Speaking of 'Minimum Samples for Splitting,' this hyperparameter sets the least number of samples needed to divide an internal node. Commonly used values for 'min_samples_split' include 2, 5, or 10. Additionally, 'Random State' is another crucial hyperparameter used to seed random number generation during tree construction. This parameter allows for reproducibility in results and can be set to values like 0, 42, or any chosen integer (MahendraVardhan & Sridhar, 2022; Joy & Selvan, 2022; Alawad et al., 2018). The values chosen in the proposed model to detect Ransomware are summarized in Table 3. These values are the most common when using a DT classifier.

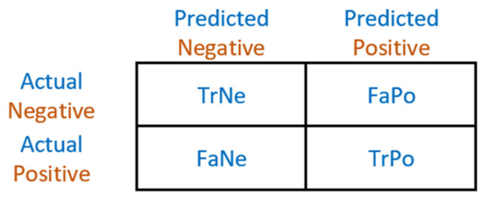
Table 3
Hyperparameters Value

Hyperparameter	Used Value
Criterion	gini
Maximum Depth	None
Minimum Samples for Splitting	2
Random State	42

4. Implementation, Results and Discussion

The proposed model was conducted on Lenovo desktop with AMD Ryzen 7 5Gen 5700G processor (8 cores, 20M cache, and up to 4.6GHz clock speed), 16GB DDR4-3200 memory, SSD M.2 512GB, NVIDIA GeForce 12GB GDDR6 graphics card, and Ubuntu 22.04.3 LTS O.S. Python was used to test and evaluate detecting Ransomware using DT. Python is one of the most popular programming languages for ML. Python has a massive ecosystem of libraries and modules that cover a wide range of functionalities. Some of the main libraries used in this work are NumPy, Pandas, and Scikit-Learn. The k-fold cross-validation technique is utilized to assess the performance of Ransomware detection using a DT classifier to reduce the risk of overfitting. K-fold divided the dataset into five subsets of about equal size. The model is then trained and assessed five times, each utilizing one-fold as the validation subset and the residual k-1 folds as the training subset (Abualhaj et al., 2022; Al-sharaiah et al., 2023).

The confusion matrix is utilized to assess the achievement of a proposed model. The confusion matrix provides a summary of how many instances were correctly classified and how many were misclassified. It consists of four components: True Positives (TrPo), True Negatives (TrNe), False Positives (FaPo), and False Negatives (FaNe). TrPo is the number of positive instances that were precisely determined as positive by the DT. For example, in Ransomware detection, this would be the number of actual Ransomware attacks precisely identified by the DT. TrNe is the number of samples that were actually negative and were precisely determined as negative by the DT. In Ransomware detection, this would be the number of none Ransomware attacks precisely identified. FaPo is the number of samples that were actually negative but were incorrectly classified as positive by the DT. In Ransomware detection, this would represent none Ransomware attacks that were falsely diagnosed as Ransomware attacks. FaNe is the number of positive instances that were imprecisely classified as negative by the DT (Abualhaj et al., 2022; Alsharaiah et al., 2023). Fig. 5 depicts the component of the confusion matrix. In a Ransomware detection, this would be cases of the Ransomware attacks that DT missed. Using the four values of the confusion matrix, we can calculate various performance metrics to assess the quality of Ransomware detection using DT, including Accuracy, Precision, Recall, and F1-scor.



	Predicted Negative	Predicted Positive
Actual Negative	TrNe	FaPo
Actual Positive	FaNe	TrPo

Fig. 5. Confusion matrix

4.1 Accuracy

Accuracy is a straightforward metric to evaluate the achievement of DT in detecting Ransomware. Accuracy is the proportion of correctly predicted instances (both Ransomware attacks and none [TrPo and TrNe]) to the total number of instances in the dataset. The Accuracy of Ransomware detection using DT is calculated using Eq. (1) (Abualhaj et al., 2022; Alsharaiah et al., 2023).

$$Accuracy = \frac{(TrPo+TrNe)}{(TrPo+TrNe+FaPo+FaNe)} \tag{1}$$

Fig. 6 clarifies the Accuracy of Ransomware detection using DT against other classifiers. As we can see, the DT classifier accomplished the highest Accuracy of 99.97%, while the closest classifier is KNN, which achieved an Accuracy of 99.95%, with an improvement of 0.02%. On the other hand, the NB classifier achieved the lowermost Accuracy of 98.41% among all the classifiers.

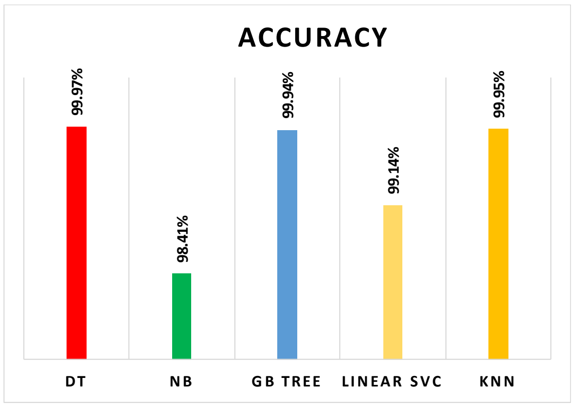


Fig. 6. Accuracy of Ransomware detection

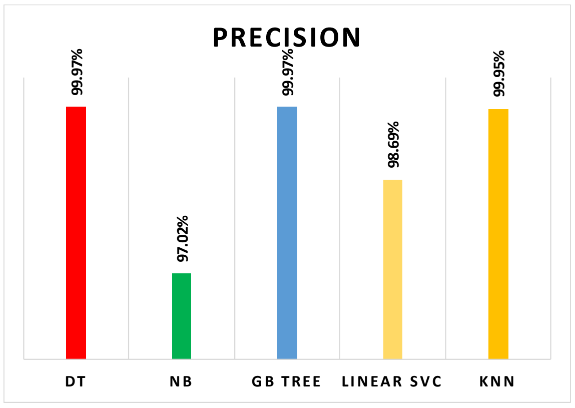


Fig. 7. Precision of Ransomware detection

4.2 Precision

Precision is a metric that can be utilized to assess the accuracy of positive predictions in detecting Ransomware when using DT. Precision quantifies the proportion of TrPo predictions among all instances where DT is predicted as a Ransomware attack. The Precision of Ransomware detection using DT is calculated using Eq. (2) (Abualhaj et al., 2022; Alsharaiah et al., 2023).

$$Precision = \frac{TrPo}{(TrPo+FaPo)} \tag{2}$$

Fig. 7 clarifies the Precision of Ransomware detection using DT against other classifiers. As we can see, the DT and GB Tree classifiers achieved the highest Precision of 99.97%, while the closest classifier is KNN, which achieved a Precision of 99.95%, with an improvement of 0.02%. On the other hand, the NB classifier achieved the lowermost Precision of 97.02% among all the classifiers.

4.3 Recall

Recall finds the capability of DT to accurately identify all Ransomware attack instances out of all actual Ransomware attack instances. Recall quantifies the proportion of TrPo predictions among all actual Ransomware attack instances. The Recall of Ransomware detection using DT is calculated using Eq. (3) (Abualhaj et al., 2022; Alsharaiah et al., 2023).

$$Recall = \frac{TrPo}{(TrPo+FaNe)} \tag{3}$$

Fig. 8 clarifies the Recall of Ransomware detection using DT against other classifiers. As we can see, the DT classifier achieved the highest Recall of 99.97%, while the closest classifier is KNN, which achieved a Recall of 99.95%, with an improvement of 0.02%. On the other hand, the Linear SVC classifier achieved the lowermost Recall of 99.57% among all the classifiers.

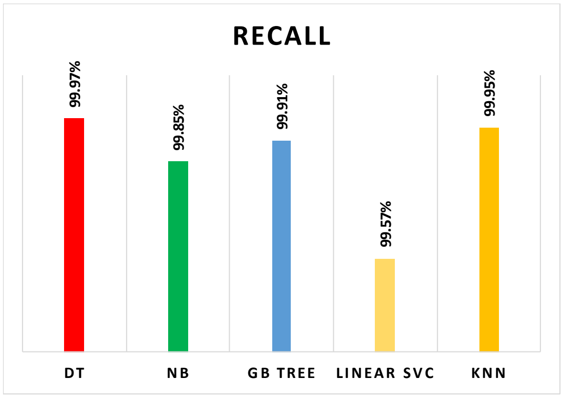


Fig. 8. Recall of Ransomware detection

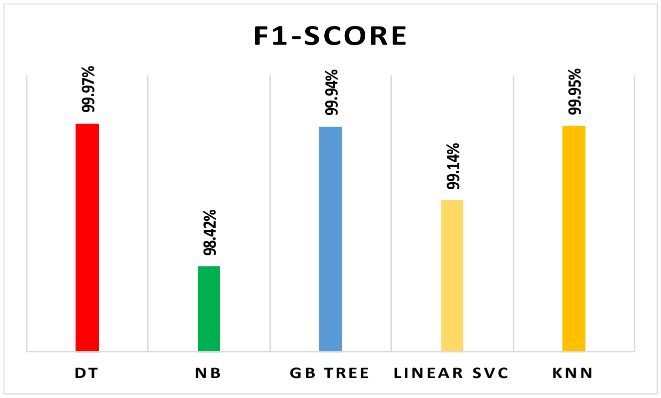


Fig. 9. F1-score of Ransomware detection

4.3 F1-score

The F1-score is the harmonic mean of precision and recall. F1-score combines both metrics into one value, providing a balanced assessment of DT achievement. The F1 Score provides a single metric for both FaPos and FaNes. The F1-score of Ransomware detection using DT is calculated using Eq. (4) (Abualhaj et al., 2022; Alsharaiah et al., 2023).

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

Fig. 9 clarifies the F1-score of Ransomware detection using DT against other classifiers. As we can see, the DT classifier accomplished the highest Accuracy of 99.97%, while the closest classifier is KNN, which achieved an Accuracy of 99.95%, with an improvement of 0.02%. On the other hand, the NB classifier achieved the lowermost Accuracy of 98.42% among all the classifiers.

5. Conclusion

In the ever-evolving cybersecurity landscape, Ransomware has emerged as a potent and widespread threat, posing numerous challenges for businesses and organizations. This article harnesses the power of the DT classifier to combat Ransomware. Our research delves into the effectiveness of DTs in identifying Ransomware attacks, capitalizing on their innate ability to decipher intricate patterns within vast datasets. We conducted rigorous testing and studies, going beyond the mere proposal of a detection method. Our study emphasizes the critical role of fine-tuning hyperparameters for DTs, underscoring their significance. We used the Obfuscated-MalMem2022 dataset for our analysis, which is renowned for its challenging Ransomware data. The implementation was carried out using the versatile Python programming language, well-regarded for its efficiency and adaptability in data analysis and ML tasks. Remarkably, our study yielded a compelling conclusion: the DT classifier consistently outperformed other classifiers in Ransomware detection, achieving a remarkable success rate of 99.97%. This outstanding accuracy underscores the efficacy of our chosen approach, solidifying DTs as a potent tool in the battle against viruses. This study bolsters the arsenal of cybersecurity tools to safeguard digital environments from increasingly sophisticated threats.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

For this research work all authors have equally contributed in related works, method design, implementation, and writing.

References

Abualhaj, M. M., Abu-Shareha, A. A., Hiari, M. O., Alrabanah, Y., Al-Zyoud, M., & Alsharaiah, M. A. (2022). A Paradigm for DoS Attack Disclosure using Machine Learning Techniques. *International Journal of Advanced Computer Science and Applications, 13*(3).

Agrawal, R., Stokes, J. W., Selvaraj, K., & Marinescu, M. (2019, May). Attention in recurrent neural networks for ransomware detection. *In ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 3222-3226)*. IEEE.

Alawad, W., Zohdy, M., & Debnath, D. (2018, September). Tuning hyperparameters of decision tree classifiers using computationally efficient schemes. *In 2018 IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE) (pp. 168-169)*. IEEE.

Almashhadani, A. O., Kaiiali, M., Sezer, S., & O'Kane, P. (2019). A multi-classifier network-based crypto ransomware detection system: A case study of locky ransomware. *IEEE access, 7*, 47053-47067.

Al-Mimi, H., Hamad, N. A., Abualhaj, M. M., Daoud, M. S., Al-dahoud, A., & Rasmi, M. (2023). An Enhanced Intrusion Detection System for Protecting HTTP Services from Attacks. *International Journal of Advances in Soft Computing & Its Applications, 15*(2).

Alqahtani, A., Gazzan, M., & Sheldon, F. T. (2020, January). A proposed crypto-ransomware early detection (CRED) model using an integrated deep learning and vector space model approach. *In 2020 10th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0275-0279)*. IEEE.

Alsharaiah, M., Abu-Shareha, A., Abualhaj, M., Baniata, L., Adwan, O., Al-saaidah, A., & Oraiqat, M. (2023). A new phishing-website detection framework using ensemble classification and clustering. *International Journal of Data and Network Science, 7*(2), 857-864.

Alves, T., Das, R., & Morris, T. (2018). Embedding encryption and machine learning intrusion prevention systems on programmable logic controllers. *IEEE Embedded Systems Letters, 10*(3), 99-102.

Chen, C. W., Su, C. H., Lee, K. W., & Bair, P. H. (2020, February). Malware family classification using active learning by learning. *In 2020 22nd International Conference on Advanced Communication Technology (ICACT) (pp. 590-595)*. IEEE.

Choudhary, S., & Sharma, A. (2020, February). Malware detection & classification using machine learning. *In 2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3) (pp. 1-4). IEEE.*

Dener, M., Ok, G., & Orman, A. (2022). Malware detection using memory analysis data in big data environment. Applied Sciences, 12(17), 8604.

Deng, X., Jiang, M., & Cen, M. (2022, December). A Ransomware Classification Method Based on Entropy Map. *In 2022 IEEE 21st International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS)* (pp. 1-8). IEEE.

Firdausi, I., Erwin, A., & Nugroho, A. S. (2010, December). Analysis of machine learning techniques used in behavior-based malware detection. *In 2010 second international conference on advances in computing, control, and telecommunication technologies (pp. 201-203). IEEE.*

Jain, P., Rajvaidya, I., Sah, K. K., & Kannan, J. (2022, February). Machine Learning Techniques for Malware Detection-a Research Review. *In 2022 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1-6). IEEE.*

Joy, J., & Selvan, M. P. (2022, June). A comprehensive study on the performance of different Multi-class Classification Algorithms and Hyperparameter Tuning Techniques using Optuna. *In 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS) (pp. 1-5). IEEE.*

Kolhar, M., Al-Turjman, F., Alameen, A., & Abualhaj, M. M. (2020). A three layered decentralized IoT biometric architecture for city lockdown during COVID-19 outbreak. *Ieee Access, 8*, 163608-163617.

Lei, J., Gao, S., Shi, J., Wei, X., Dong, M., Wang, W., & Han, Z. (2022). A Reinforcement Learning Approach for Defending Against Multiscenario Load Redistribution Attacks. *IEEE Transactions on Smart Grid, 13*(5), 3711-3722.

Ma, L., Sun, B., & Han, C. (2018, July). Training instance random sampling based evidential classification forest algorithms. *In 2018 21st International Conference on Information Fusion (FUSION) (pp. 883-888). IEEE*.

MahendraVardhan, A., & Sridhar, S. (2022, December). Determining False Positive Analysis of Software Vulnerabilities with Predefined Scan Rules using Random Forest Classifier and Decision Tree Technique. *In 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N) (pp. 622-625). IEEE*.

Manavi, F., & Hamzeh, A. (2021, March). Static detection of ransomware using LSTM network and PE header. *In 2021 26th International Computer Conference, Computer Society of Iran (CSICC) (pp. 1-5). IEEE*.

Molina, R. M. A., Torabi, S., Sarieddine, K., Bou-Harb, E., Bouguila, N., & Assi, C. (2021). On ransomware family attribution using pre-attack paranoia activities. *IEEE Transactions on Network and Service Management, 19*(1), 19-36.

Parizad, A., & Hatziadoniu, C. J. (2022). Cyber-attack detection using principal component analysis and noisy clustering algorithms: A collaborative machine learning-based framework. *IEEE Transactions on Smart Grid, 13*(6), 4848-4861.

Peng, W., Li, F., Zou, X., & Wu, J. (2013). Behavioral malware detection in delay tolerant networks. *IEEE Transactions on Parallel and Distributed systems, 25*(1), 53-63.

Razali, M. H. M., Saian, R., Moktar, B., Wah, Y. B., & Ku-Mahamud, K. R. (2022, September). Performance of ACO-based Decision Tree Algorithm with Imbalanced Class Data Sets-A Heuristic Approach. *In 2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS) (pp. 29-38). IEEE*.

Rosmansyah, Y., & Dabarsyah, B. (2015, August). Malware detection on android smartphones using API class and machine learning. *In 2015 International Conference on Electrical Engineering and Informatics (ICEEI) (pp. 294-297). IEEE.*

Saurabh. (2018). Advance malware analysis using static and dynamic methodology. *2018 International Conference on Advanced Computation and Telecommunication (ICACAT)*. https://doi.org/10.1109/icacat.2018.8933769

Saurav, Z., Mitu, M. M., Ritu, N. S., Hasan, M. A., Arefin, S., & Farid, D. M. (2023, February). A New Method for Learning Decision Tree Classifier. *In 2023 International Conference on Electrical, Computer and Communication Engineering (ECCE) (pp. 1-6). IEEE.*

Sen, S., Aydogan, E., & Aysan, A. I. (2018). Coevolution of mobile malware and anti-malware. *IEEE Transactions on Information Forensics and Security, 13*(10), 2563-2574.

SonicWall, cyber threat report, 2022.

Vijayarangam, J., Kamalakkannan, S., & Smiles, J. A. (2021, November). A Novel Comparison of Neural Network and Decision Tree as Classifiers using R. *In 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 712-715). IEEE*.

Wang, C., & Zhu, H. (2020). Representing fine-grained co-occurrences for behavior-based fraud detection in online payment services. *IEEE Transactions on Dependable and Secure Computing, 19*(1), 301-315.

Yeo, M., Koo, Y., Yoon, Y., Hwang, T., Ryu, J., Song, J., & Park, C. (2018, January). Flow-based malware detection using convolutional neural network. *In 2018 International Conference on Information Networking (ICOIN) (pp. 910-913). IEEE*.

Zulfikar, W. B., Gerhana, Y. A., & Rahmania, A. F. (2018, August). An approach to classify eligibility blood donors using decision tree and naive bayes classifier. *In 2018 6th International Conference on Cyber and IT Service Management (CITSM) (pp. 1-5). IEEE.*