A Ransomware Early Detection Model based on an Enhanced Joint Mutual Information Feature Selection Method

Tasnem Magdi Hassin Mohamed
Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia
tasnem@graduate.utm.my

Bander Ali Saleh Al-rimy
Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia
bander@utm.my (corresponding author)

Sultan Ahmed Almalki
Computer Department, Applied College, Najran University, Najran 66462, Saudi Arabia
saalmalki@nu.edu.sa

Received: 17 February 2024 | Revised: 4 March 2024 and 21 March 2024 | Accepted: 24 March 2024

ABSTRACT

Crypto ransomware attacks pose a significant threat by encrypting users' data and demanding ransom payments, causing permanent data loss if not detected and mitigated before encryption occurs. The existing studies have faced challenges in the pre-encryption phase due to elusive attack patterns, insufficient data, and the lack of comprehensive information, often confusing the current detection techniques. Selecting appropriate features that effectively indicate an impending ransomware attack is a critical challenge. This research addresses this challenge by introducing an Enhanced Joint Mutual Information (EJMI) method that effectively assigns weights and ranks features based on their relevance while conducting contextual data analysis. The EJMI method employs a dual ranking system—TF for crypto APIs and TF-IDF for non-crypto APIs—to enhance the detection process and select the most significant features for training various Machine Learning (ML) classifiers. Furthermore, grid search is utilized for optimal classifier parameterization, aiming to detect ransomware efficiently and accurately in its pre-encryption phase. The proposed EJMI method has demonstrated a 4% improvement in detection accuracy compared to previous methods, highlighting its effectiveness in identifying and preventing crypto-ransomware attacks before data encryption occurs.

Keywords-ransomware;early detection; machine learning; feature selection

I. INTRODUCTION

In the relatively short history of the Internet, the relationship between crime and technology has grown [1]. Both legitimate users and criminals have benefited from technological advancements. While technology has evolved, many traditional crime schemes such as blackmail, extortion, and theft have persisted. However, with the highly automated environment of the Internet, criminals can now automate their attacks and target millions of online users, rather than being limited to large entities like banks or corporations with substantial resources. As a result of this transformation, cybercrime has become widespread and, in the past few years, there has been a notable upswing in ransomware attacks worldwide. These attacks target victims without discrimination, rapidly encrypting their data and effectively restricting organizations and individuals from accessing their computers. As a result, the number of distinct ransomware variants has surged by an astonishing 600% [1]. Ransomware can generally be categorized into two types based on its functionality: cryptographic ransomware, which encrypts the files of the victim, and locker ransomware, which prevents access to the victim's system. Regardless of the specific approach employed, both types of ransomwares demand a ransom payment to regain access to the files or the affected system [2].

Even though the first ransomware was made in 1989 and has been around on and off for more than 30 years, it has only been one of the most well-known threats since 2005. Cybercriminals have perfected ransomware attack parts (such as stronger encryption techniques, pseudo-anonymous payment methods, worm-like capabilities, etc.) and have even started to offer Ransomware as a Service (RaaS) by learning from past mistakes and taking advantage of technological advances [3]. Crypto-ransomware stands out from other types of malwares due to two distinct characteristics, which have been emphasized in prior research [2, 4, 5]. Crypto-ransomware is characterized by its deceptive behavior resembling benign programs and the irreversibility of its attack. Crypto-ransomware demonstrates a behavior that closely resembles legitimate software by leveraging cryptography applications and APIs approved by the system, specifically targeting files associated with users [4]. Moreover, the use of cryptography makes the targeted files inaccessible even if the crypto-ransomware responsible for the attack is detected and removed. Once the targeted resource is encrypted, it becomes exceedingly difficult to restore access without possessing the decryption key [6]. This irreversibility underscores the importance of early detection in effectively mitigating crypto-ransomware attacks [5, 7-9]. Given the irreversible nature of ransomware attacks, it is crucial to detect them before encryption occurs. Current feature selection methods in ransomware detection, while effective to a certain extent, often fall short of addressing the unique behaviors and patterns of these attacks. This necessitates the development of an advanced feature selection methodology, capable of comprehensively understanding and responding to ransomware's distinct characteristics. Our research aims to fill this gap by introducing the Enhanced Joint Mutual Information (EJMI) method. This method uniquely employs a dual ranking system, specifically designed to evaluate, and select features based on their critical relevance to ransomware behavior. This approach differs from previous studies, which did not sufficiently prioritize feature selection based on the nuanced patterns and tactics of ransomware attacks, therefore missing key indicators essential for early detection. The introduction of a dual ranking system in EJMI is a novel approach that promises to enhance the accuracy and timeliness of ransomware detection.

To develop an enhanced model for selecting the most informative features for ransomware detection, a comprehensive understanding of ransomware mechanisms and previous detection models is essential. This understanding is critical for identifying the limitations of current approaches and for devising more effective strategies. The upcoming section, addresses this need. It provides a detailed examination of how ransomware operates and reviews the existing detection models. This analysis is not only crucial for highlighting the gaps in current methodologies but also serves as a foundation upon which the EJMI method is built. By dissecting existing methods, the path is paved for a more advanced and nuanced approach to ransomware detection.

II. RANSOMWARE EXISTING STUDIES

A. *Ransomware Analysis*

Analyzing ransomware helps in understanding its behavior which can help build detection models that would detect ransomware effectively. Currently, there are two primary approaches to ransomware analysis: static analysis and dynamic analysis.

Static analysis entails examining the characteristics of ransomware without executing the actual samples. This involved inspecting elements such as the Portable Executable (PE) header, opcode, and PE strings, which may have contained ransom notes. However, static analysis is time-consuming and requirs manual effort. On the other hand, the dynamic analysis involved running ransomware samples in a controlled environment and capturing real-time logs of their behaviors. This included monitoring activities such as file system operations, registry changes, memory usage, and function calls. Dynamic analysis is generally considered more effective as it enables the detection of ransomware behaviors in real time [6]. Authors in [10] introduced a framework known as UNVEIL, which aimed to detect ransomware by analyzing Input/output Request Packets (IRPs) obtained from a mini-filter driver. This approach established a simulated user environment that incorporated kernel-level indicators of threats to identify any unauthorized modification of files. Another method proposed in [11] involved monitoring the Master File Table (MFT) for encryption activity, file deletion, and duplication. Authors in [12] developed ShieldFS, an automated healing file system that incorporated critical factors such as file system activity, file entropy, and the Windows registry, including ransomware configuration keys, into its monitoring capabilities. ShieldFS was designed to proactively detect and mitigate ransomware attacks by continuously analyzing these factors and taking appropriate actions to protect the integrity of the system and the files [13]. Authors in [14] proposed a reactive approach where encryption keys are stored securely and accessed only when a ransomware attack is detected. This helps protect against crypto ransomware. Authors in [15] introduced a static analysis technique that detected ransomware by analyzing n-gram sequences derived from opcode sequences. One advantage of this approach was that it eliminated the requirement for a sandboxed environment. However, it had limitations in terms of its ability to delve deeply and effectively handle modern obfuscation techniques employed by ransomware.

Many methods, such as static analysis, dynamic analysis, and API-based techniques, have been utilized to detect malware and combat ransomware attacks. Among these strategies were the reactive and proactive approaches, which aimed to identify and counteract ransomware through different means. For instance, one approach involved monitoring API calls to extract specific features that could be used to identify ransomware attacks. These extracted features were then utilized to train ML models, enabling them to effectively detect ransomware. This combination of techniques enhanced the overall ability to identify and respond to ransomware threats. ML and Deep Learning (DL) have a significant impact on various domains, including computing and cybersecurity. These technologies have found widespread applications in these fields, revolutionizing processes, and enabling more efficient solutions [16]. Their use facilitated the detection of advanced attacks and threats in a more efficient and timely manner. ML excels in recognizing patterns within complex processes, proving itself to be valuable in domains such as medicine, security, artificial intelligence, and entertainment [17]. ML and DL are rapidly advancing technologies and have been extensively utilized in cybersecurity research. Their potential for pattern identification makes them effective tools in detecting malware and ransomware.

B. Ransomware Detection using Machine Learning

ML is highly effective in detecting ransomware due to its ability to analyze large datasets and identify intricate patterns. Ransomware attacks are complex and constantly evolving, making static rule-based detection systems challenging to implement. ML algorithms excel at recognizing patterns and anomalies, continuously adapting to new samples and variations. By training on historical data, ML models can identify general patterns and detect previously unseen ransomware variants, making them adaptable to the evolving nature of attacks. That is why ML is highly recommended for ransomware early detection. Overall, ML offers an efficient and flexible approach to ransomware detection [17]. Extensive investigations have been conducted to discover effective methods for detecting ransomware at an early stage, prior to the encryption of user data.

In [18], a pioneering approach was introduced to detect early-stage crypto-ransomware using an innovative framework. This framework incorporated an adaptable pre-encryption technique and addressed the concept of population drift, which played a crucial role in effectively countering evolving variants of ransomware. To identify the most relevant features, an improved mutual information feature selection method was utilized. These selected features were then utilized to train an online classifier that continually updated itself through a stochastic gradient descent algorithm. Through the implementation of these techniques, the framework aimed to enhance the detection capabilities of ransomware, facilitating early intervention, and minimizing the potential damage. To capture and document the actions of the ransomware sample, a process monitor was employed during execution. This monitoring tool captured and recorded the actions performed by the sample for a specified duration, typically set to 10 s [19]. The proposed approach involved monitoring and logging the activities carried out by the ransomware sample, including its interactions with the operating system via APIs. This information was utilized to train the detection model. However, it is worth noting that running the sample for a duration of only 10 s may have led to a potential oversight of the encryption phase, which is a critical component of ransomware behaviors. Authors in [20] analyzed multiple features of ransomware and goodware samples, encompassing Windows API calls, registry key operations, file system operations, file operations per file extension, directory operations, dropped files, and strings. To identify the most significant features, the sample was executed within a controlled environment known as a sandbox, and its behavior was monitored for a duration of 30 s. The study utilized the Regularized Logistic Regression classifier, EldeRan, to classify new samples as either "ransomware" or "goodware," based on the identified features. Authors in [21] introduced a technique called Dynamic Pre-encryption Boundary Delineation (DPBD) to tackle the challenges associated with fixed threshold issues in identifying the pre-encryption boundary in crypto-ransomware attacks. The DPBD technique employed a vector space model combined with Rocchio relevance feedback to construct a vector representing the pre-encryption boundary. This vector encompassed all cryptography-related APIs invoked by instances of crypto-ransomware during attacks. By analyzing the occurrence of these vector entries, the DPBD technique could accurately identify the initial occurrence of any entry as the boundary between the pre-encryption phase and the encryption phase of the attacks. Through tracking the encryption initiation point for each instance, the DPBD technique precisely determined the boundary of the pre-encryption phase based on the specific cryptography-related APIs utilized by instances of crypto-ransomware during the encryption process of user files. In [5], the authors employed three crucial components to delineate the boundary of the pre-encryption phase: Temporal Data Correlation (TDC), Pre-encryption Feature Extraction (IPFE), and the Training and Testing of the early detection model. The TDC technique involved examining the timestamps of cryptographic APIs and IRPs and grouping them accordingly. This analysis helped identify the initial occurrence where a match between an API/IRP indicated the transition from the pre-encryption phase to the encryption phase of ransomware. However, it is important to consider that an API/IRP match could arise from two distinct operations or normal system behaviors. As the system performed multitasking, an API/IRP match may have arisen as a result of the behaviors of another system.

The system proposed in [22] operated by monitoring the communication between an infected machine and the Command-and-Control (C&C) server. It utilized Programmable Forwarding Engines (PFEs), which are advanced network devices capable of accurately monitoring all incoming and outgoing traffic to and from the potential victim. The system's methodology involved analyzing the network traffic signatures associated with ransomware. By examining the patterns and characteristics present in the network traffic, the system could identify and detect the presence of ransomware activities. In [23], DL models were trained on labeled datasets containing both normal system behavior and ransomware attack instances. The models learned to identify unique patterns and signatures associated with ransomware in the logging data. These models were then deployed to continuously monitor the real-time host logs, comparing incoming data against the learned patterns.

The methodology described in [8] focused on extracting the API calls made by ransomware before encryption occurred. These extracted API calls were then subjected to analysis using ML algorithms. By comparing the extracted API calls with a repository of known ransomware signatures, the methodology aimed to identify if the ransomware belonged to a known variant. If the ransomware was not recognized as a known variant, the ML algorithm analyzed its behavior by looking at APIs that included the word "crypto." The called APIs were then labeled as pre-encryption APIs. However, crypto API calls could also be used by normal system use by benign programs or during unpacking or obfuscation. Authors in [24] introduced a novel method for early ransomware detection by analyzing the API calls made by each ransomware instance. This technique involved capturing and logging the API calls used by each instance in a separate trace file, which enabled the identification of specific activities performed by the ransomware. The paper suggested that if any cryptography-related APIs were called, it indicated the initiation of an encryption process, which could potentially be the starting point of a ransomware attack targeting the user's files and data. By monitoring and analyzing these API calls, this approach aimed to detect ransomware in its early stages, allowing for timely intervention and mitigation of the attack.

Authors in [9] proposed a method called Enhanced Mutual Information Feature Selection (EMBR), an advanced technique aimed to enhance the accuracy of early ransomware attack detection. It operates by selecting a subset of features from a larger feature set extracted during the pre-encryption phase of ransomware. These selected features are then utilized to train a ML classifier for ransomware detection. Authors in [25] discussed an innovative approach to the early detection of ransomware based on Hardware Performance Counters (HPCs). They used HPCs to gather data during the execution of both ransomware and benign software, focusing on the unique lower-level characteristics that differentiated ransomware. They extracted features from these data and built a classification model capable of distinguishing between benign and ransomware activities using ML algorithms, including Random Forest and GBM. This approach proved to be a breakthrough in augmented reactive cybersecurity measures, demonstrating high accuracy in instantly blocking ransomware upon execution. The mechanism described in [26] handles the difficulty of behavioral drift in ransomware detection. It employed weighted Generative Adversarial Networks (wGANs) for data augmentation, generating synthetic data that closely mirrored the evolving behaviors of ransomware. The enriched dataset allowed the model to train on a diverse array of ransomware attack patterns, thereby increasing its adaptability and accuracy. Additionally, the model incorporated mutual information for estimating feature significance, enabling it to adapt to changes in ransomware behavior over time. This dual strategy of data augmentation and adaptive feature assessment ensured that the detection system remained effective against new and evolving ransomware variants, thereby tackling the critical issue of behavioral drift in ransomware detection.

Authors in [27] presented an enhanced method for detecting ransomware using a Generative Adversarial Network (GAN) with an improved loss function. This GAN comprised a generator module for creating synthetic ransomware instances and a discriminator module for distinguishing between real and fake data. The main innovation lay in the Bi-Gradual Minimax (BGM) loss function, which fine-tuned the probability estimation process, thereby enhancing the accuracy of the generated attack patterns. Furthermore, the approach incorporated Long Short-Term Memory (LSTM) networks for early detection and employed Mutual Information Feature Selection (MIFS) to identify the most informative features. This method improved the model's capability in detecting ransomware during its early stages with increased accuracy, recall, and a reduced false positive rate. In [28] a component-based system for detecting crypto-ransomware attacks using straightforward, threshold-based techniques was presented. The system monitored specific performance parameters, such as CPU usage, memory usage, and disk read/write operations. Initially, it established averages for these parameters under normal, non-malicious conditions and during ransomware attacks, based on data sampling. When a parameter exceeded its established threshold, indicative of a significant deviation from normal behavior, the system flagged it as a potential ransomware activity. This method focused on the most influential parameters that demonstrated significant deviations during attacks, showcasing its practicality, scalability, and adaptability to various computer system architectures. It provided a simple yet effective approach to ransomware detection by utilizing observable changes in system performance.

In [29], the authors presented REDDS (Ransomware Early Detection and Defense System), a novel system designed to proactively identify and defend against ransomware attacks. REDDS dynamically captures API sequences during the early, pre-encryption stages of ransomware attacks and transforms them into feature vectors using the n-gram model and Term Frequency-Inverse Document Frequency (TF-IDF) algorithm. To enhance the dataset's quality, a Wasserstein GAN with Gradient Penalty (WGAN-GP) was employed. Furthermore, the system incorporated a Ransomware Defense Countermeasure Ontology (RDCO) to automatically derive defense strategies by mapping malicious APIs to security knowledge bases. This innovative combination of API sequence analysis, data augmentation with WGAN-GP, and ontology-based defense inference significantly improved the effectiveness of early ransomware detection and response strategies. In [30], the authors introduced a method for detecting ransomware based on the dynamic analysis of Windows API calls. This technique involved tracing and extracting only significant features from Windows API calls that indicated ransomware activities. The researchers employed a specialized parser engine for the abstraction of API features and constructed n-grams (subsequences of API call sequences) to develop behavioral profiles of the samples. They then fine-tuned these API calls, focusing on the most relevant ones and attenuating those that were not, thereby highlighting characteristic differences of ransomware. This refined set of API call patterns and sequences was then used to enhance the performance of ML algorithms in distinguishing ransomware from benign software, thus improving both the accuracy and effectiveness of ransomware detection.

C. Limitations and Discussion

The criticality of selecting the most informative features for an accurate detection model cannot be overstated. While previous studies have utilized APIs as features in various ways, it is essential to recognize that not all APIs contribute equally to ransomware behavior. This necessitates a more discerning approach, particularly focusing on crypto APIs due to their significant impact. However, other non-crypto APIs also play vital roles and can offer insights into ransomware operational patterns. To effectively balance this, a dual ranking system is paramount. Such a system would prioritize features based on their relevance to ransomware detection. This is where the

EJMI method comes into play. EJMI is modified for contextual data analysis, utilizing features weights in accordance with their significance in detecting ransomware. This approach marks a significant advancement over traditional methods, offering a more nuanced and effective means of early ransomware detection.

III. THE PROPOSED FRAMEWORK

A. The EJMI Concept

The EJMI method represents a significant evolution of the traditional Joint Mutual Information (JMI) approach. EJMI's theoretical foundation is based on the concept of mutual information, which quantifies the amount of information obtained about one variable through another. While JMI effectively measures mutual dependence among variables, EJMI extends this by integrating weighting factors that are sensitive to the context of the data, particularly in the domain of ransomware detection. The key innovation of EJMI over JMI is its ability to assign varying degrees of importance to different features based on their relevance to the target variable (e.g. identifying ransomware). This is achieved through the use of distinct weighting methodologies for different types of features, such as crypto and non-crypto APIs. For crypto APIs, which are less frequent but highly indicative of ransomware, EJMI employs the TF-IDF method. This method assigns higher weights to rare but significant features, ensuring that they are not overshadowed by more common features. For non-crypto APIs, EJMI uses Term Frequency (TF) weights, which reflect the occurrence rate and contextual relevance of these features. TF is simply the relevant frequency of a word in a document. It is calculated as the number of times a word appears in a document divided by the total number of words in that document. The main idea is that the more frequently a term occurs in a document, the more important it is for that document. The TF-IDF method builds upon TF by adjusting the frequency of a term based on how common or rare it is across all documents. It is calculated by multiplying the TF of a term by its Inverse Document Frequency (IDF). IDF is calculated as the logarithm of the number of documents divided by the number of documents containing the term. TF-IDF decreases the weight of terms that occur very frequently in the document set (thus considered less important) and increases the weight of terms that occur rarely.

By applying these tailored weights to the JMI scores of individual features, EJMI enhances the feature selection process, making it more nuanced and effective for ransomware detection. This approach enables EJMI to identify features that are most predictive of the target variable, leading to more accurate and reliable models in distinguishing between normal operations and ransomware activities.

B. The EJMI Mathematical Concept

EJMI incorporates C(x) and N(x) weighting factors, and can be expressed as:

$$\mathrm{EJMI}= \sum_{\mathrm{i} \in \mathrm{features}} \omega_{\mathrm{i}} \times \mathrm{JMI}(\mathrm{X_i};\mathrm{Y}) \quad (1)$$

where $\mathrm{X_i}$ represents the $\mathrm{i^{th}}$ feature in the crypto or non-crypto API set, Y is the target variable, and ω_{i} is the weighting factor for the $\mathrm{i^{th}}$ feature, which differs based on whether the feature is a crypto API or a non-crypto API

For crypto APIs, the weighting factor ω_{i} is represented by C(x) and is calculated using TF-IDF(x) and for non-crypto APIs, it is represented by N(x) and is calculated by the TF.

1) Crypto APIs

$\mathrm{C(x)}$ indicates the weighting term. It gives crypto API X a certain weight, allowing it to have more influence in the overall JMI calculation. $\mathrm{C(x)}$ is calculated based on the domain. In ransomware detection, it is a function that assigns higher weights to cryptographic API calls.

$$\mathrm{C(x) = TF - IDF(crypto\ API)} \quad (2)$$

$$\mathrm{TF - IDF(t, d, D) = TF(T, D) \times IDF(t, D)} \quad (3)$$

TF-IDF measures the frequency of the API call in a particular document and balances it against its frequency across the entire corpus. Since crypto API calls do not appear as frequently as other types of API calls, this method ensures that non-crypto API calls do not dominate the ranking process.

2) Non crypto APIs

This term captures additional context or patterns around X. It is essential to recognize unique patterns in which X operates. In a dynamic system, X will depend on the domain and data. It represents the behavioral patterns that align with API calls that represent other features than crypto API calls.

$$\mathrm{N(X) = TF(non\ crypto\ API)} \quad (4)$$

$$\mathrm{TF(t, d)} = \frac{\mathrm{(No.of\ times\ the\ term\ t\ appears\ in\ a\ document\ d)}}{\mathrm{(Total\ number\ of\ terms\ in\ the\ document\ d)}} \quad (5)$$

IV. RESULTS OF THE PROPOSED MODEL

This section presents the empirical study's results, showcasing the EJMI's effectiveness in ransomware detection. The analysis incorporates a Fully Convolutional Neural Network (FCNN), a DL classifier, alongside conventional ML classifiers, such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest (RF). These classifiers were evaluated on accuracy, precision, recall, F1 score, and ROC-AUC metrics. This evaluation serves to compare EJMI's performance against other statistical methods such as JMI [31], MRMR [32], and MIFS [33]. The dataset used in this research, which includes API call data, has been instrumental in various ransomware detection studies [7]. It contains 1524 samples, with 942 benign and 582 ransomware instances from families such as cerber, teslacrypt, cryptowall, petya, and wannacry. This dataset's consistent use in prior research highlights its significance and reliability in the cybersecurity field. The Cuckoo Sandbox was employed as the analysis tool for the dataset in this study to examine the ransomware samples and capture the distinctive features associated with ransomware

A. Random Forest

The integration of the EJMI method with the RF classifier has demonstrated significant advancements in ransomware detection accuracy. By meticulously weighting features and performing contextual data analysis, EJMI provides a nuanced approach to feature selection. This methodology, coupled with

Random Forest's inherent strengths in handling diverse data sets, ensures a balanced and effective selection of the most informative features. Consequently, this synergy enhances the classifier's capability to accurately identify ransomware patterns, marking a notable improvement over traditional detection model. The graphical presentation in Figure 1 serves as an essential reference for evaluating the impact and efficiency of the EJMI method in enhancing ransomware detection through the RF classifier.

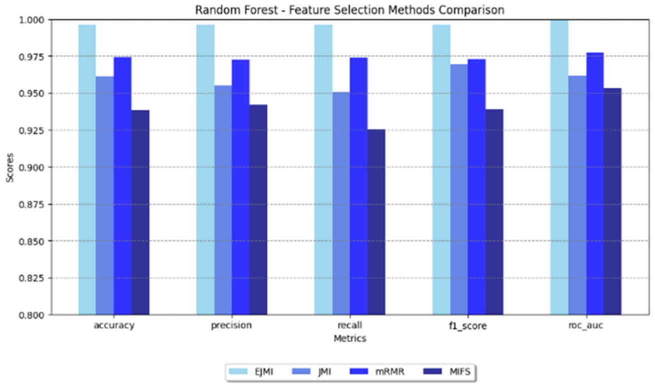

Fig. 1. Feature selection metrics using RF.

Table I shows a quantitative comparison of feature selection methods. It can be seen that EJMI outperforms the other methods, achieving nearly perfect scores across all metrics, particularly in ROC-AUC, which is almost 1..

TABLE I. QUANTITATIVE COMPARISON OF FEATURE SELECTION USING RF

Metric	EJMI	JMI	mRMR	MIFS
Accuracy	99.62%	96.14%	97.41%	94.85%
Precision	99.63%	95.52%	97.27%	94.19%
Recall	99.62%	95.06%	97.38%	94.56%
F1_score	99.62%	96.49%	97.31%	94.91%
ROC-AUC	99.98%	96.16%	97.73%	94.35%

B. *Support Vector Machine (SVM)*

In the context of ransomware detection, SVM's capability to deal with high-dimensional data makes it an excellent tool for evaluating the performance of the EJMI method. Figure 2 shows the results. The comparison with the results of JMI, MRMR, and MIFS across all evaluation metrics is listed in the Table II. EJMI achieves a near to perfect ROC-AUC score, by enhancing the JMI accuracy by almost 4%.

TABLE II. QUANTITATIVE COMPARISON OF FEATURE SELECTION USING SVM

Metric	EJMI	JMI	mRMR	MIFS
Accuracy	98.77%	94.65%	94.46%	90.26%
Precision	98.82%	95.15%	94.62%	90.31%
Recall	98.77%	94.33%	94.40%	90.08%
F1_score	98.77%	93.69%	94.30%	90.18%

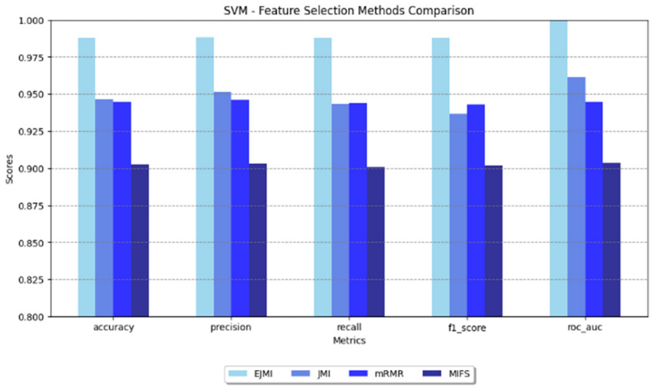

Fig. 2. Feature selection metrics using SVM.

C. *K-Nearest Neighbors (KNN)*

KNN, known for its simplicity and efficacy in classification tasks, works by identifying the nearest data points in the feature space and making decisions based on their classifications.

The results of the conjunction of KNN with EJMI can be seen in Figure 3.

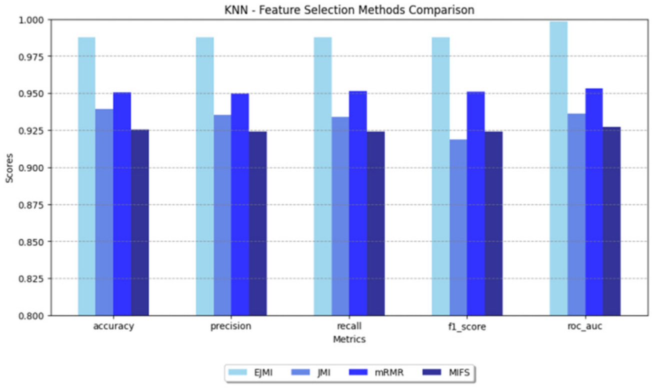

Fig. 3. Feature selection metrics using KNN.

The KNN classifier results with EJMI and other feature selection methods are listed in Table III which shows the impressive performance of EJMI.

TABLE III. QUANTITATIVE COMPARISON OF FEATURE SELECTION USING KNN

Metric	EJMI	JMI	mRMR	MIFS
Accuracy	98.77%	93.94%	95.04%	92.54%
Precision	98.80%	93.52%	94.96%	92.40%
Recall	98.77%	93.41%	95.14%	92.40%
F1_score	98.77%	91.88%	95.08%	92.42%
ROC-AUC	99.86%	93.62%	95.30%	92.71%

D. *Fully Convolutional Neural Network (FCNN)*

The FCNN, a DL classifier, stands out for its ability to process spatial hierarchies in data, making it highly suitable for tasks involving images or patterns with spatial relationships. In ransomware detection, the FCNN's architecture can be leveraged to discern complex feature interactions that are indicative of ransomware activity.

As per Figure 4, when FCNN was used with the EJMI method, showcased promising results, outperforming other feature selection methods like JMI, MRMR, and MIFS in ransomware detection. The FCNN's DL capabilities, combined with EJMI's nuanced feature selection, provided a powerful tool for identifying ransomware. This indicates that DL classifiers, with the right feature selection method, could be more effective in detecting complex patterns associated with ransomware threats.

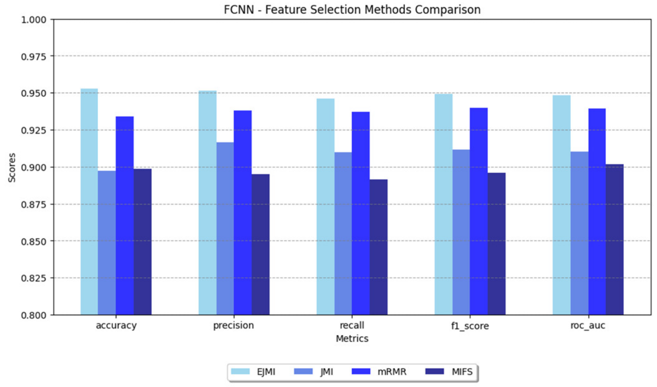

Fig. 4. Feature selection metrics using FCNN.

Table IV presents a comparative analysis of the FCNN using different feature selection methods. The EJMI method shows a high level of accuracy, precision, recall, F1 score, and ROC-AUC, indicating its strong performance in ransomware detection

TABLE IV. QUANTITATIVE COMPARISON OF FEATURE SELECTION USING FCNN

Metric	EJMI	JMI	mRMR	MIFS
Accuracy	95.18%	91.46%	95.22%	98.10%
Precision	95.05%	90.36%	94.45%	90.44%
Recall	96.20%	91.39%	94.29%	89.40%
F1_score	95.92%	91.38%	95.51%	90.86%
ROC-AUC	95.89%	91.32%	94.98%	89.71%

V. DISCUSSION

EJMI showed its strength as a feature selection technique as compared to other statistical feature selection methods. By introducing weighing concept differently for each feature, EJMI was particularly effective at choosing the characteristics that are most indicating of ransomware attacks. This proposed approach ensures that the important but less frequent indicators are not overshadowed by more common but less informative features, thus selecting the most informative features that will improve the model's detection capabilities and be able to clearly detect ransomware.

Selecting the most informative features is important for ransomware detection model and its accuracy. The EJMI method tailors this process by assigning different weights to features. Cryptographic APIs play a significant role in encryption processes and are used by ransomware for this purpose and their importance warrants distinct weighting as they directly indicate encryption activities. However, they may also appear in benign system behaviors for other processes such as packing, therefore a balanced analysis with other system APIs is necessary. EJMI's approach combines the differently weighted mutual information of both cryptographic and non-cryptographic features against the target, facilitating the selection of the most contextually informative data for clear ransomware pre-encryption identification and detection.

VI. CONCLUSION

The current study introduced the Enhanced Joint Mutual Information (EJMI) method, offering a unique approach to ransomware detection. The EJMI method performs contextual data analysis and incorporates a dual ranking system, i.e. TF for crypto APIs and TF-IDF for non-crypto APIs. This innovative approach improved the accuracy and efficiency of detecting ransomware by 4% compared to the original Joint Mutual Information (JMI) method. Additionally, the EJMI method demonstrated a significant reduction in the false alarm rate, marking a substantial advancement over traditional feature selection approaches. The successful application of the EJMI method in this research paves the way for its integration into cybersecurity strategies, offering a more robust defense against crypto-ransomware attacks by accurately identifying and ranking the most relevant features for detection. The improved performance, in terms of increased detection accuracy and reduced false alarms, highlights the effectiveness of the EJMI method and its potential for broader implementation in ransomware mitigation efforts.

REFERENCES

[1] Y. A. Ahmed, B. Kocer, and B. A. S. Al-rimy, "Automated Analysis Approach for the Detection of High Survivable Ransomware," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 5, pp. 2236–2258, May 2020.

[2] H. Oz, A. Aris, A. Levi, and A. S. Uluagac, "A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions," *ACM Computing Surveys*, vol. 54, no. 11s, Jan. 2022, Art. no. 238, https://doi.org/10.1145/3514229.

[3] Y. A. Ahmed, B. Koçer, S. Huda, B. A. Saleh Al-rimy, and M. M. Hassan, "A system call refinement-based enhanced Minimum Redundancy Maximum Relevance method for ransomware early detection," *Journal of Network and Computer Applications*, vol. 167, Oct. 2020, Art. no. 102753, https://doi.org/10.1016/j.jnca.2020.102753.

[4] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions," *Computers & Security*, vol. 74, pp. 144–166, May 2018, https://doi.org/10.1016/j.cose.2018.01.001.

[5] A. Alqahtani and F. T. Sheldon, "Temporal Data Correlation Providing Enhanced Dynamic Crypto-Ransomware Pre-Encryption Boundary Delineation," *Sensors*, vol. 23, no. 9, Jan. 2023, Art. no. 4355, https://doi.org/10.3390/s23094355.

[6] M. Almousa, S. Basavaraju, and M. Anwar, "API-Based Ransomware Detection Using Machine Learning-Based Threat Detection Models," in *18th International Conference on Privacy, Security and Trust*, Auckland, New Zealand, Dec. 2021, pp. 1–7, https://doi.org/10.1109/PST52912.2021.9647816.

[7] A. M. A. Assaggaf, B. A. Al-Rimy, N. L. Ismail, and A. Al-Nahari, "Development of Graph-Based Knowledge on Ransomware Attacks Using Twitter Data," in *The International Conference on Data Science and Emerging Technologies*, Dec. 2022, pp. 168–183, https://doi.org/10.1007/978-981-99-0741-0_12.

[8] S. H. Kok, A. Abdullah, and N. Jhanjhi, "Early detection of crypto-ransomware using pre-encryption detection algorithm," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 5, pp. 1984–1999, May 2022, https://doi.org/10.1016/j.jksuci.2020.06.012.

[9] B. A. S. Al-rimy *et al.*, "Redundancy Coefficient Gradual Up-weighting-based Mutual Information Feature Selection technique for Crypto-ransomware early detection," *Future Generation Computer Systems*, vol. 115, pp. 641–658, Feb. 2021, https://doi.org/10.1016/j.future.2020.10.002.

[10] A. Al-Marghilani, "Comprehensive Analysis of IoT Malware Evasion Techniques," *Engineering, Technology & Applied Science Research*, vol. 11, no. 4, pp. 7495–7500, Aug. 2021, https://doi.org/10.48084/etasr.4296.

[11] J. Kumar and G. Ranganathan, "Malware Attack Detection in Large Scale Networks using the Ensemble Deep Restricted Boltzmann Machine," *Engineering, Technology & Applied Science Research*, vol. 13, no. 5, pp. 11773–11778, Oct. 2023, https://doi.org/10.48084/etasr.6204.

[12] A. Continella *et al.*, "ShieldFS: A Self-healing, Ransomware-aware Filesystem," in *32nd Annual Conference on Computer Security Applications*, Los Angeles, CA, USA, Dec. 2016, pp. 336–347, https://doi.org/10.1145/2991079.2991110.

[13] N. Scaife, H. Carter, P. Traynor, and K. R. B. Butler, "CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data," in *36th International Conference on Distributed Computing Systems*, Nara, Japan, Jun. 2016, pp. 303–312, https://doi.org/10.1109/ICDCS.2016.46.

[14] E. Kolodenker, W. Koch, G. Stringhini, and M. Egele, "PayBreak: Defense Against Cryptographic Ransomware," in *ACM on Asia Conference on Computer and Communications Security*, Abu Dhabi, United Arab Emirates, Apr. 2017, pp. 599–611, https://doi.org/10.1145/3052973.3053035.

[15] K. Aldriwish, "A Deep Learning Approach for Malware and Software Piracy Threat Detection," *Engineering, Technology & Applied Science Research*, vol. 11, no. 6, pp. 7757–7762, Dec. 2021, https://doi.org/10.48084/etasr.4412.

[16] D. W. Fernando, N. Komninos, and T. Chen, "A Study on the Evolution of Ransomware Detection Using Machine Learning and Deep Learning Techniques," *IoT*, vol. 1, no. 2, pp. 551–604, Dec. 2020, https://doi.org/10.3390/iot1020030.

[17] U. Urooj, B. A. S. Al-rimy, A. Zainal, F. A. Ghaleb, and M. A. Rassam, "Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions," *Applied Sciences*, vol. 12, no. 1, Jan. 2022, Art. no. 172, https://doi.org/10.3390/app12010172.

[18] U. Urooj, M. A. B. Maarof, and B. A. S. Al-rimy, "A proposed Adaptive Pre-Encryption Crypto-Ransomware Early Detection Model," in *3rd International Cyber Resilience Conference*, Langkawi Island, Malaysia, Jan. 2021, pp. 1–6, https://doi.org/10.1109/CRC50527.2021.9392548.

[19] S. Homayoun, A. Dehghantanha, M. Ahmadzadeh, S. Hashemi, and R. Khayami, "Know Abnormal, Find Evil: Frequent Pattern Mining for Ransomware Threat Hunting and Intelligence," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 2, pp. 341–351, Apr. 2020, https://doi.org/10.1109/TETC.2017.2756908.

[20] D. Sgandurra, L. Munoz-Gonzalez, R. Mohsen, and E. C. Lupu, "Automated Dynamic Analysis of Ransomware: Benefits, Limitations and use for Detection." arXiv, Sep. 10, 2016, https://doi.org/10.48550/arXiv.1609.03020.

[21] B. A. S. Al-Rimy *et al.*, "A Pseudo Feedback-Based Annotated TF-IDF Technique for Dynamic Crypto-Ransomware Pre-Encryption Boundary Delineation and Features Extraction," *IEEE Access*, vol. 8, pp. 140586–140598, 2020, https://doi.org/10.1109/ACCESS.2020.3012674.

[22] G. Cusack, O. Michel, and E. Keller, "Machine Learning-Based Detection of Ransomware Using SDN," in *ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, Tempe, AZ, USA, Mar. 2018, pp. 1–6, https://doi.org/10.1145/3180465.3180467.

[23] K. C. Roy and Q. Chen, "DeepRan: Attention-based BiLSTM and CRF for Ransomware Early Detection and Classification," *Information Systems Frontiers*, vol. 23, no. 2, pp. 299–315, Apr. 2021, https://doi.org/10.1007/s10796-020-10017-4.

[24] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, "Crypto-ransomware early detection model using novel incremental bagging with enhanced semi-random subspace selection," *Future Generation Computer Systems*, vol. 101, pp. 476–491, Dec. 2019, https://doi.org/10.1016/j.future.2019.06.005.

[25] P. M. Anand, P. V. S. Charan, and S. K. Shukla, "HiPeR-Early Detection of a Ransomware Attack using Hardware Performance Counters," *Digital Threats: Research and Practice*, vol. 4, no. 3, 2023, Art. no. 43, https://doi.org/10.1145/3608484.

[26] U. Urooj, B. A. S. Al-Rimy, A. B. Zainal, F. Saeed, A. Abdelmaboud, and W. Nagmeldin, "Addressing Behavioral Drift in Ransomware Early Detection Through Weighted Generative Adversarial Networks," *IEEE Access*, vol. 12, pp. 3910–3925, 2024, https://doi.org/10.1109/ACCESS.2023.3348451.

[27] M. Gazzan and F. T. Sheldon, "An Enhanced Minimax Loss Function Technique in Generative Adversarial Network for Ransomware Behavior Prediction," *Future Internet*, vol. 15, no. 10, Oct. 2023, Art. no. 318, https://doi.org/10.3390/fi15100318.

[28] P. Roemsri, S. Puangpontip, and R. Hewett, "On Detecting Crypto Ransomware Attacks: Can Simple Strategies be Effective?," in *6th International Conference on Information and Computer Technologies*, Raleigh, NC, USA, Mar. 2023, pp. 138–143, https://doi.org/10.1109/ICICT58900.2023.00030.

[29] S. Zhang, T. Du, P. Shi, X. Su, and Y. Han, "Early Detection and Defense Countermeasure Inference of Ransomware based on API Sequence," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 10, pp. 632–641, Jan. 2023, https://doi.org/10.14569/IJACSA.2023.0141067.

[30] Q. Kang and Y. Gu, "Enhancing Ransomware Detection: A Windows API Min Max Relevance Refinement Approach." Nov. 16, 2023, https://doi.org/10.20944/preprints202311.1004.v1.

[31] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection using Joint Mutual Information Maximisation," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8520–8532, Dec. 2015, https://doi.org/10.1016/j.eswa.2015.07.007.

[32] A. Hashemi, M. B. Dowlatshahi, and H. Nazamabadi-pour, "Minimum redundancy maximum relevance ensemble feature selection: A bi-objective Pareto-based approach," *Journal of Soft Computing and Information Technology*, vol. 12, no. 1, pp. 20–28, 2023.

[33] B. P. Joshi, N. Joshi, S. Oli, A. Rayal, A. Kumar, and A. Singh, "MIFS Ordered Weighted Operators method for renewable-energy-source-selection," in *2nd International Conference on Industrial Electronics: Developments & Applications*, Imphal, India, Sep. 2023, pp. 248–253, https://doi.org/10.1109/ICIDeA59866.2023.10295267.