IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 4, August 2026, pp. 3865~3877
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i4.pp3865-3877

Prioritizing ransomware indicators of compromise using algorithmic scoring for enhanced threat detection

Krishna Prasad D. Subramanya[1,2], Prasanna Kumar H. Ramakrishna[2,3]
[1]Department of Computer and Communication Engineering, NMAM Institute of Technology, Nitte (Deemed to be University), Karnataka, India
[2]Department of Computer Science and Engineering, Visvesvaraya Technological University, Belagavi, India
[3]Department of Computer Science and Engineering, PES Institute of Technology and Management, Shivamogga, India

Article Info

Article history:

Received Jun 13, 2025
Revised Jun 7, 2026
Accepted Jul 9, 2026

Keywords:

Dynamic analysis
Hybrid analysis
Indicators of compromise
Ranking indicators of compromise
Ransomware
Static analysis

ABSTRACT

Ransomware is a very serious cybersecurity threat. Its attack methods are continually evolving, which means that it is not very easy to detect it quickly. A big challenge faced in the ransomware defense is how to prioritize indicators of compromise (IoC) efficiently. This paper introduces a hybrid IoC ranking scheme based on static and dynamic analysis of the behavior of commonly circulating ransomware variants. Each IoC is given a weighted score according to its significance for investigations based on actual patterns of occurrence and contextual behavior. Experimental results clearly separate high-confidence indicators from low-confidence ones. Deleting shadow copy and connecting to Tor2web receive the highest rank scores of about 99.99, while older behaviors like locking screen show lower relevance, around 73.11. The proposed algorithm has a linear time complexity of $O(n \cdot m)$ for score calculation and a bounded space complexity of $O(n \cdot m)$, allowing it to scale for large IoC sets. The findings show that this ranked IoC framework enhances early ransomware detection and helps prioritize responses based on evidence in current security systems.

Corresponding Author:

Krishna Prasad D. Subramanya
Department of Computer and Communication Engineering, NMAM Institute of Technology
Nitte (Deemed to be University)
Karnataka, India
Email: krishna270572@gmail.com

1. INTRODUCTION

The ransomware attacks threaten individuals, organizations, and critical infrastructure globally. Ransomware is a type of malware that encrypts important data or entire system and demands a ransom for its release [1]. These attacks have changed dramatically in the last few years moving from a simple locker software to complex double-extortion schemes where data is encrypted before being threatened to be publicly released to maximize financial gain [2], [3]. Recent statistics for 2024 show that 59% of organizations experienced ransomware attacks, a slight decrease since previous two years (66%). About 70% of these attacks involved data encryption, pointing out the success of ransomware tactics [4]. The average ransom demand in 2024 was $2.73 million, an increase of almost $1 million from 2023. What's more, the average cost of recovering from a ransomware attack in 2023 was $1.85 million [4], [5], pointing out the severe financial burden on the affected organizations.

Detecting and preventing ransomware attacks quickly is crucial to limit financial losses, operational interruptions, and damage to its reputation. Indicators of compromises (IoCs) are very important for detecting ransomware activities inside an organization's network [6], [7]. IoCs are artifacts or pieces of forensic

evidence such as unusual network traffic patterns, file hashes, registry changes, and suspicious domain connections, that indicate a potential security breach. However, the vast amount and range of IoCs make it difficult for cybersecurity professionals to focus on the most relevant indicators for early detection and prevention [8]. In this way, it is necessary to prioritize the IoCs according to their reliability and their behavior importance to enhance the cyber defense strategies [9]. Forensic investigations are more relevant to specific behaviors of ransomware, such as deletion of shadow copies or communication with command and control (C&C) with anonymized networks. Other artifacts, like dynamic-link library (DLL) imports, or cryptographic functions usage, that may be present in legitimate applications, may cause confusion in detection [10]. To address this problem, IoCs need to be prioritized according to their relevance and reliability of a signal of a ransomware attack [11]. IoC ranking is effective by allowing security teams to allocate resources wisely, improve accuracy of detection, and be quick to respond to threats. While there are many available, some are more important than others when it comes to ransomware detection. A well-organized ranking system can help to distinguish benign and malicious activities.

This study proposes a systematic approach for rating IoCs related to ransomware by utilizing a hybrid analysis using both static and dynamic analyses of known ransomware variants. It begins with a literature review that explains the typical ransomware attack lifecycle, and examines both static and dynamic analysis techniques for recognizing and evaluating ransomware threats, and gives context for recognizing and evaluating ransomware threats. Then, the methodology outlines the approach to the analysis of well-known ransomware variants by using a specific list of IoCs and the information about the most outstanding activities of them. A ranking algorithm for IoC is then proposed, where the indicators are prioritized according to how important they are for the diagnostics. Lastly, in the results section, the algorithm performance is analyzed, assessing its efficiency in terms of time complexity and space complexity and discussing its usability and applicability in real-time threat detection environments.

2. LITERATURE REVIEW

Cyber attacks are generally carried out in a well-defined manner. When it comes to ransomware this cycle generally includes the following: malware distribution, malware infection, C&C communication, file encryption, demand of ransom, and potential decryption. The ransomware attack cycle is a systematic process cybercriminals use to break into a system, encrypt essential data, requirement for extorting money from the victims. A typical attack cycle is shown in Figure 1.

It begins with malware being delivered primarily via phishing e-mails [12] to entice users to open malware-laden attachments or click on malicious links. After the malware is executed, it is linked to a C&C server to retrieve public encryption key [13]. It then encrypts files or locks down the system, and commonly makes it difficult to recover, including shadow copies [3], [14]. Typically, a ransom note appears, outlining the payment conditions, usually in the form of cryptocurrency for the decryption key [15]. But it is not a sure thing that ransom will provide recovery of data, making it a very sophisticated and very damaging attack.

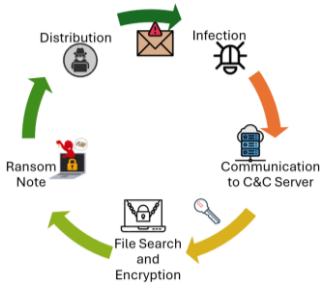

Figure 1. Ransomware attack life cycle

2.1. Ransomware analysis

Ransomware attacks are increasing in complexity and are a growing threat to individuals and businesses alike. There are several approaches that have been developed for analyzing these risks, such as static, dynamic, and hybrid approaches. In this paper, their applicability and value in identification and prevention of ransomware attacks were discussed. Static analysis is the review of a program's code without executing the program. It's a rapid and effective way to identify known malware: code signatures and patterns. It is fast, can scan files for known signatures in a quick time, and can be used to detect known malware [16]. It's not as effective with polymorphic or metamorphic malware that can change their code to evade detection, however, and is not very effective against new or unknown variants of malware [16]–[18].

Dynamic analysis is the process of executing the program on a controlled environment to observe the program behavior and its interaction with the environment. This is a powerful method, since a zero day and a polymorphic malware can be easily detected, as well as giving insights about the behavioral patterns [18]. It is very compute intensive and time consuming, however, it can be fooled by malware that can tell if it is running in a virtual environment and adjust its behavior to evade detection [17], [19].

Hybrid analysis combines static and dynamic analysis techniques to capitalize on the benefits of each, while overcoming its limitations. It provides comprehensive detection by analyzing codes and observing behaviors, and early detection by analyzing both static and runtime features [19]–[21]. It can discover ransomware early on by assessing both these attributes [22]. Although hybrid analysis has advantages, its implementation is complex, and it requires a significant number of resources [23] which can be mitigated by optimized approaches. In most cases, hybrid analysis will be more accurate in detecting than either static or dynamic analysis will be alone. In particular, static analysis achieved between 40-55% accuracy, whereas in some studies, 100% accuracy was achieved when using dynamic analysis. The hybrid method has yielded positive results, and detection rates of more than 95% are commonly reported [24]. Hybrid systems can be improved by using static analysis to eliminate innocuous programs from further analysis, and using dynamic analysis only for suspicious programs, thereby reducing the computational cost. Hybrid analysis has proven to be effective in the detection of unique behaviors and in the improvement of detection rates for ransomware families such as petya, cryptinfinite, and locky [25], [26].

Hybrid analysis is the most effective technique to detect ransomware, combining the strength and depth of dynamic analysis, and the speed and efficiency of static analysis. Its thorough detection capabilities make it an essential tool in the fight against ransomware, notwithstanding the practical hurdles it poses. Hybrid analysis can be a comprehensive and effective solution to the evolving threat of ransomware by combining the strengths of both static and dynamic approaches, resulting in more accurate and timely detection and limiting potential damage.

This work focuses on hybrid analysis, i.e., dynamic analysis combined with static analysis to extract IoCs. Ransomware samples contain a range of indicators and patterns that can be structurally and behaviorally important for accurate identification, in addition to the common indicators such as internet protocol (IP) and file hashes. In this research, an approach to rating and prioritizing these IoCs based on their relevance and frequency of occurrence in the context of their threat is proposed. By systematically analyzing these indicators, it is expected to enhance the effectiveness of ransomware detection and aid in better threat response and mitigation strategies.

2.2. Indicators of compromise for ransomware

The algorithm maintains a list of IoC names to process the extracted data efficiently. Table 1 lists the IoCs considered in this investigation. The selected IoCs are important for revealing key ransomware strategies and behaviors. Each IoC is important for a good detection and analysis because it provides unique insights into the ransomware's process of infecting, surviving, hiding, communicating, and encrypting data in a system.

Table 1. List of ransomware IoCs and its type considered for the study

Sl. No.	IoC	Description	Type
1	Delete shadow copy	Used to prevent victims from recovering files without paying the ransom. The ransomware executes commands like vssadmin delete shadows [27].	Dynamic
2	I2P anonymity network	Invisible internet project (I2P) network by ransomware indicates attempts to hide communication with C&C servers [28].	Static
3	Connect to Tor2web	Use of anonymous networks like Tor or I2P to communicate with C&C servers to avoid detection. Prevents tracking and identification of operators by law enforcement or researchers [29].	Dynamic
4	Request to high entropy domain name	Generate and connect to domain names with high entropy, often associated with domain generation algorithms (DGAs), making them difficult to track [30].	Dynamic
5	File encryption	Ransomware uses strong encryption algorithms (e.g., advanced encryption standard (AES), Rivest–Shamir–Adleman (RSA)) to encrypt files and store the decryption key with attacker [31]–[33].	Dynamic
6	Encrypts file name	Renames files using specific patterns or appends unique extensions. Adds visual indication of encryption to intimidate victims and make file recovery more difficult [29].	Dynamic
7	Locks screen	Locks the victim's screen with a ransom note, blocking access to the system until payment is made. The ransomware may disable Task Manager, override system processes, or display a full screen ransom message [33].	Dynamic
8	Deletes original files from disk	After encrypting files, ransomware often deletes the original versions to prevent recovery. Ensures the victim cannot retrieve unencrypted data using standard recovery tools [34].	Dynamic
9	Import and links to crypto libraries	The presence of imports related to cryptographic libraries, such as OpenSSL or Windows CryptoAPI, is a strong indicator of ransomware attempting to encrypt files [35].	Static
10	Packed/obfuscated	Used to evade static analysis. The entropy value is a crucial metric used to determine the presence of packed or obfuscated code [36], [37].	Static
11	Create RWX memory	Malware allocates memory regions with read-write-execute (RWX) permissions to execute malicious code in memory [38], [39].	Dynamic

3. METHOD

The major contribution of this paper is to present a unique method to prioritize and rank IoCs according to their importance and occurrence in ransomware detection. This method includes feature extraction using both static and dynamic analysis of ransomware variants, rank calculation and prioritize IoCs list. Figure 2 gives a good outline of the whole process. To be sure that the ranking algorithm receives correct and structured input data, each IoC has its frequency of occurrence and weight. The ranking system successfully identifies most important IoCs for ransomware detection by combining IoC occurrence, threat significance weights and feature identification. The main process of IoC ranking method involves examining ransomware samples by static and dynamic analysis approaches. Critical IoCs are extracted by these analyses, and they are used for ranking.

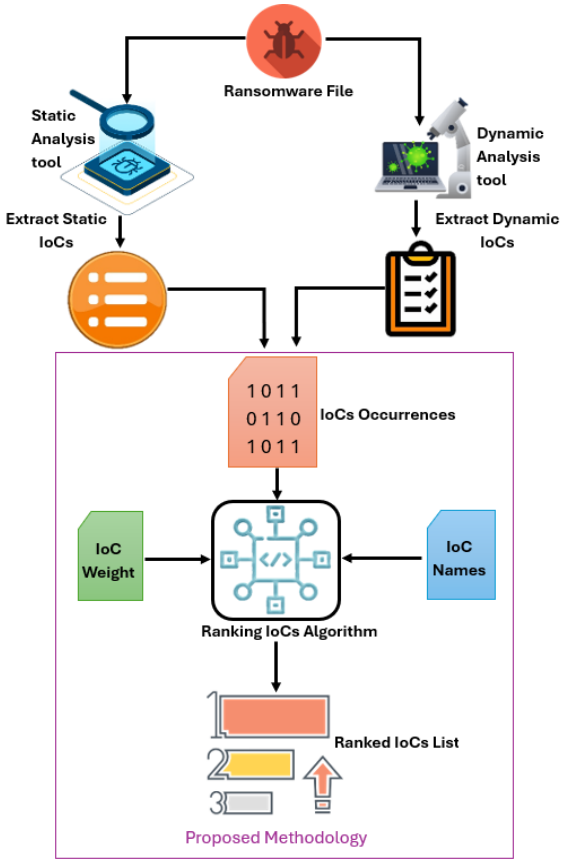

Figure 2. Process of ranking IoCs related to ransomware

3.1. Static and dynamic analysis of ransomware variants

The methodology applied hybrid analysis which includes both static and dynamic analysis of selected ransomware variants to comprehensively extract and validate major IoCs relevant to ransomware signature and behavior. For static analysis, PEStudio was used to analyze file attributes without running it and focusing on metadata, import functions, suspicious strings, obfuscation markers, and known hashes. Cuckoo Sandbox tool was used to extract dynamic behaviors of ransomware sample by executing the sample under a closed and safe environment.

The results obtained from both methods of analysis were compared to identify common and important artifacts. These were then linked to IoCs of choice for this study. They include encryption techniques, file modifications, registry modifications for persistence, creation of ransom note files, communication with anonymizing networks, or indications of packing and/or obfuscation. Table 2 summarizes these observations and gives a good view of the behavioral footprint of the ransomware samples. It also confirms the presence of IoCs that are needed for the next steps of scoring and ranking. These two layered approaches help in ensuring a wide and deep identification of IoCs, enhancing the threat intelligence base for the proposed ranking algorithm.

Table 2. Summary of IoCs of various ransomware variants analyzed during the study

Ransomware variant	File extensions	Encryption algorithm(s) used	Ransom notes	Registry changes
Crypto wall	.decrypt, .lock, .aaa, .zzz, .cryptowall	RC4, RSA-2048	HELP_DECRYPT.TXT, HELP_DECRYPT.HTML, HELP_DECRYPT.PNG	Creates an autorun registry entry
Crypto locker	.encrypted, .crypto, .locked	RSA-2048	DECRYPT_INSTRUCTION.TXT, DECRYPT_INSTRUCTION.HTML	Disables windows system restore
CTB-locker	.ctb, .ctb2, .locky	Elliptic Curve Cryptography	!Decrypt-All-Files.txt, How_To_Decrypt.txt	Creates a persistence entry
Locky	.locky, .zepto, .odin, .osiris	AES-128 and RSA-2048	Locky_Recovery_Instructions.txt, LOCKY_HELP.txt	Alters windows policies to disable recovery
TeslaCrypt	.vvv, .ttt, .micro, .mp3, .xxx	AES	HELP_RESTORE_FILES.TXT, HELP_TO_DECRYPT_YOUR_FILES.TXT	Creates persistence entry
TorrentLocker	.enc, .crypted, .locked	AES-256 and RSA-2048	ECRYPT_INSTRUCTION.TXT, README_FOR_DECRYPT.txt	Deletes windows shadow copies
WinLocker	Does not encrypt files but locks the screen	AES-256	Displays full-screen ransom messages	– Blocks task manager – Alters shell settings to prevent access
WannaCry	.WNCRY	Combination of the RSA and AES	GUI ransom note from @WanaDecryptor@.exe	– Modifies the registry to ensure persistence on reboot – Disables windows defender and security tools

3.2. Ranking indicators of compromise algorithm

The static and dynamic IoCs collected are stored in an IoC_Occurrence file. This file tracks how often each IoC is spread across different variants of ransomware. This information is crucial to the ranking algorithm. IoCs that are common across several ransomware samples likely will be more important for threat detection. After collecting the IoCs of ransomware samples, they become inputs of the ranking process. In this process, importance of the most important indicators is given priority. The algorithm takes three major inputs: IoC names, their occurrences, and their weights. The method maintains a clear list of the IoC names such as that in Table 1. To make sure the ranking algorithm receives the correct and organized input data, every IoC is associated with its frequency of occurrence and weight. Through a combination of IoC occurrence, threat significance weights, and feature identification, the ranking algorithm is designed to effectively prioritize IoC's most relevant for ransomware detection. The frequency of occurrence is an indication of how often a particular IoC is found in ransomware families. The higher the frequency, the more likely the IoC is to appear in ransomware samples, making it a good indicator of ransomware activity.

The IoC_occurrence file is set up as a binary matrix, which shows the features of different ransomware. Each row in the matrix represents a unique ransomware family. Each column represents a specific IoC. For a given ransomware family, each cell in the matrix has binary value of 1 and 0. A value of 1 indicates that a specific IoC has been observed and that this behavior is a part of the harming behavior. In contrast, a value of 0 means that the ransomware variant does not display the IoC. This binary coding makes calculation easier and makes it easier to represent the data. The matrix applied for the proposed method is presented in Table 3. This setup is important because it can be used to easily compare different strains of ransomware. By organizing the families into rows, researchers can assess the similarities and dissimilarities in behaviors of ransomware families, and which ones have unique traits. Representing IoCs as columns also helps identify common indicators in a variety of ransomware families, helping to detect and classify new threats. Based on its practicality in ransomware detection, each IoC is given a predetermined threat significance weight. The weights fall into three categories and justified in Table 4.

Table 3. Binary matrix representing IoC presence across ransomware variant

Ransomware family	IoCs										
	Delete shadow copy	I2P anonymity network	Connect to tor2web	Request to high entropy domain name	File encryption	Encrypts file name	Locks screen	Deletes original files from disk	Import and links to crypto libraries	Packed/ obfuscated	Create RWX memory
WannaCry	1	1	1	1	1	0	0	1	1	1	1
CryptoWall	1	1	1	1	1	0	0	0	1	1	1
CryptoLocker	1	0	1	1	1	0	0	0	1	1	1
CTB-locker	0	0	0	1	1	1	0	0	1	1	1
Locky	1	0	1	1	1	1	0	1	1	1	1
TeslaCrypt	0	0	1	0	1	1	0	0	1	1	0
TorrentLocker	1	0	1	1	1	1	0	1	1	1	1
WinLocker	0	0	0	0	1	0	1	0	0	0	0

Table 4. Weight allocation for the IoCs and the justification for choosing the weight allocation

Weight given to IoCs	Justification for the weight allocation	IoCs list
3	High-risk IoCs are assigned a higher weight of '3' due to their strong association with ransomware activity.	Shadow copy deletion, request to high entropy domain name, connect to tor2web, deletes original files from disk, encrypts file name.
2	IoCs with moderate risk are given medium weights with a value of '2', indicating the possibility of ransomware activity but also the possibility of legal apps using them.	I2P anonymity network, packed/obfuscated, import and links to crypto libraries
1	They are less suggestive of ransomware-specific activity, low-risk IoCs are given lower weights with a value of '1'.	File encryption, locks screen, create RWX memory

A collection of IoCs' score is determined by Algorithm 1 using their weights and the frequency with which they appear in ransomware samples. As stated in (1), the scores are calculated by multiplying the IoC occurrences by the weights allocated to them and adding up the weighted values for each IoC. The parameters and their description are as follows:

i) IOCs[]: a list or array containing all the IoCs.
ii) n = len(IOCs): this determines the number of IoCs.
iii) IoC_Occurences[][]: a 2D matrix (list of lists) where each element IOC_Occurences[j][i] represents how many times the ith IoC occurs in the jth ransomware sample.
iv) Weights[]: a vector or list in which the weight allocated to the ith IoC is represented by weights[i]. The significance of the IoC for ransomware detection is indicated by this weight.
v) Scores[]: a list containing each IoC's final scores. A certain IoC's weighted occurrence across all ransomware samples is represented by each score.

The goal is to calculate scores for each IoC as a weighted sum of its occurrences across all ransomware samples. The mathematical formulation of the score for an IoC i is as in (1).

$$Score_i = \sum_{j=0}^{m-1}(IOC\_Occurences[j][i] \times weights[i]) \quad (1)$$

Where i is the index of the IoC, j is the index of a ransomware sample, and m is the total number of ransomware samples.

Algorithm 1. Algorithm to find the ranking of iocs to detect ransomware

```
Input  : 1. Array of n ransomwares
         2. Array of m IoCs
         3. Array of weights of each IoC
         4. 2D array where each row gives the occurrence of an IoCs in the given
            ransomwares
Output : All m IoCs score
1:   Initialize: sum=0, scores[]
2:   for i=0 to m-1 do
3:      for j= 0 to n-1do
4:         sum = sum + IOC_Occurrence[i][j]*weights[i]
5:      end for
6:      scores[i]=sum
7:      sum=0   // sum is reset to 0 for the next IoC
8:   end for
9:   end
```

Algorithm 2 applies the sigmoid function as shown in (2) to a list of scores, generated by Algorithm 1, where each IoC has been assigned a raw score based on its frequency and weight. The loop runs from i = 0 to m - 1, where m represents the total number of IoCs being evaluated. The sigmoid function is applied to each raw score. This function maps real-valued numbers into a range between 0 and 1. It is especially useful in this context to normalize scores, making them easier to interpret and compare. The output of the sigmoid is then multiplied by 100, converting it into a percentage representation and stored in percentages list at index i.

$$\sigma(i) = \frac{1}{1+e^{-i}} \quad (2)$$

Algorithm 2. Algorithm to convert scores into percentage and sort the scores

```
Input  : Array of scores computed in Algorithm 1
Output : Sorted IoCs with their percentage score
1:   for i=0 to m-1 do
2:      percentage=1/(1+e^(-Scores[i]))*100
```

```
3:      Percentages[i]=percentage
4:   end for
5:   indexes=[0,..,m-1]
6:   sort(scores[0,..,m-1],indexes[0,..,m-1]) //Use any sorting algorithm to sort the scores
7:   for i=0 to m-1 do
8:      print IOCs[indexes[i]] and Percentages[indexes[i]]
9:   end for
10:  end
```

4. RESULTS AND DISCUSSION

The algorithm examines the frequency of each indicator of the IoC, and their relevance. It uses a scoring function to transform this information and are used to normalize the results through a sigmoid transformation. This gives a percentage-based ranking of IoCs based on how important they are in spotting and defining ransomware behavior.

The algorithm was applied by assessing the IoCs listed in Table 1 against the ransomware variants shown in Table 2. This process generated occurrence data of each IoC across these variants into a matrix, which is displayed in Table 3. Based on this matrix, the algorithm calculates a rank score for each IoC, with the resulting scores summarized in Table 5.

Table 5. Rank score for the ransomware IoCs received from the algorithm

Sl. No.	Ransomware IoCs	Rank score
1	Delete shadow copy	99.99996940977731
2	Connect to Tor2web	99.99996940977731
3	Packed/obfuscated	99.99546021312976
4	File encryption	99.75273768433654
5	Request to high entropy domain name	99.75273768433654
6	Deletes original files from disk	95.25741268224334
7	I2P anonymity network	88.07970779778823
8	Import and links to crypto libraries	88.07970779778823
9	Encrypts file name	88.07970779778823
10	Create RWX memory	88.07970779778823
11	Locks screen	73.1058578630005

The top-ranked indicators with 99.9999 confidence numbers are both identified as ransomware indicators: “Delete shadow copy” and “Connect to Tor2web”. Though Tor2web communication is a good sign of a criminal intent to hide C&C channels, deletion of shadow copies is a certain way of preventing data retrieval. Packed/obfuscated code (99.9954) has a high score, and is often included in benign software, too. In the middle tier of the score, IoCs are found that are critical to ransomware, but not exclusive to ransomware, lowering the trio's discriminating power, such as “File encryption” and “Request to high entropy domain name” with scores: 99.7527. Ransomware often “Deletes original files from disk”, with 95.2574 being used, but there are other methods of secure deletion used by legitimate tools.

A few moderately scored indicators with ≈88.08 each add context, but are not highly distinguishable from ransomware individually: “I2P anonymity network”, “Import and links to crypto libraries”, “Encrypts file name”, and “Create RWX memory”. Examples include secure apps that rely on cryptographic APIs, and RWX memory generation that occurs in just-in-time (JIT) compilers. Finally, in the context of today's ransomware, “Locks screen” has lowest score of 73.1058. This aligns with the reality that today's ransomware attacks are becoming more focused on data encryption and data exfiltration, and thus screen-locking attacks are becoming less common. A true and accurate linear representation of the IoC rankings is given in the bar chart of Figure 3. This chart is supplemented with color-coded zones of confidence to indicate indicators by their level of confidence, with high confidence (scores above 90) colored red, moderate confidence (scores between 80 and 90) colored orange, and low confidence (scores under 80) colored yellow. This design clearly identifies which behaviors are most indicative of ransomware, and which behaviors are likely to cause false positives (e.g., locks screen). The threshold-zoned bar chart helps the prioritization and threshold decision-making for threat detection systems.

Descriptive behavioral patterns provide an organized prioritization of IoCs and the threat detection systems are better at allocating resources and reaction tactics according to the observed behavior patterns. The ranking results offer an ordered list of IoCs based on level of criticality and actual occurrence in the wild. These results can be used by security analysts to develop detection methodologies based on IoC with high ranking, e.g., file encryption, code obfuscation, and shadow copy removal. Instead of trying to build a holistic analytical framework, focusing on these top indicators can significantly enhance early detection and mitigation of ransomware attacks, while the lower ranked IoCs can be added as supplemental evidence.

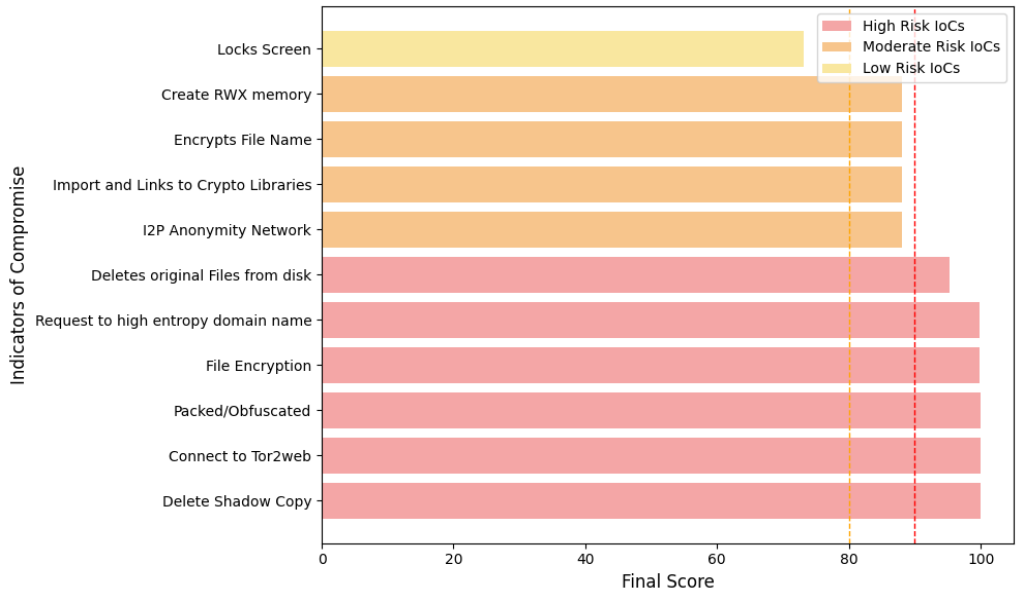

Figure 3. The confidence level of the IoCs in indicating a ransomware

4.1. Algorithm efficiency analysis

The algorithm starts by reading 4 files, namely ransomnames.txt, iocs.txt, weights.txt, and IOCoccurences.txt. There is a parsing of the lines involved in reading the file, so the time complexity of this process is $O(L1 + L2 + L3 + L4)$, where $L1, L2, L3$, and $L4$ denote the number of lines in each respective file. The space complexity for storing the parsed contents is $O(|\ RansomwareNames\ | + |\ IOCs\ | + |\ weights\ | + |\ IOC\_Occurences\ |)$, accounting for the storage of ransomware names, IoCs, weights, and their occurrences. Next, the parsed data is processed and appended to lists. Since inserting an element into a list takes $O(1)$ time per element, the overall time complexity is proportional to the total number of elements across all files.

The main computational step of matrix-vector multiplication for score calculation contains two nested loops. The outer loop iterates over n elements (number of IoCs), while the inner loop iterates over m elements (number of ransomware names). This results in a total time complexity of $O(n.m)$. The space complexity for storing the computed scores is $O(n)$, as a single score is maintained for each IoC.

The next step applies the sigmoid function to each score. Calculating the sigmoid function for a single element takes $O(1)$ time. Since there are n scores, the total time complexity is $O(n)$. The space complexity is also $O(n)$, as it requires storing intermediate results and percentages.

The final step contains sorting the calculated scores using bubble sort. In the worst case, bubble sort requires $O(n^2)$ comparisons and swaps, leading to a total time complexity of $O(n^2)$. Since bubble sort operates in place, it does not require additional memory beyond the input array, resulting in a space complexity of $O(1)$.

4.2. Overall complexity analysis

When evaluating the overall time complexity, the two dominant operations are: i) matrix-vector multiplication: $O(n.m)$ and ii) bubble sort: $O(n^2)$. Thus, the combined time complexity is $O(n.m + n^2)$. If $n \gg m$, meaning the number of IoCs is significantly larger than the number of ransomware names. The $O(n^2)$ term dominates, simplifying the total time complexity to $O(n^2)$. The performance of the proposed IoC ranking algorithm was evaluated by isolating two core components of the execution: matrix-vector multiplication (used for calculating IoC scores) and sorting (used to rank IoCs based on scores). The results, measured across increasing dataset sizes, are summarized in Table 6.

The performance of the suggested IoC ranking algorithm was measured in terms of the total execution time for different numbers of IoCs and ransomware families. The results are clear from the plot in Figure 4. The algorithm takes next to no time when the number of IoCs is small (10 to 50), and it is finished in few milliseconds. The algorithm has sustainable execution times with only a 100 increase in the number of IoCs, which it requires to complete. in about 0.15 seconds. A noticeable increase in execution time occurs as approach 500 IoCs, where execution takes around 7.51 seconds. With 1000 IoCs, the execution time climbs to 110.02 seconds. This increase shows that time complexity grows significantly with larger datasets.

Table 6. Performance evaluation of the proposed IoC ranking algorithm

Number of IoCs (n)	Families (m)	Matrix computation time (s)	Sorting time (s)	Total execution time (s)
10	5	0.002	0.0005	0.0026
50	10	0.01	0.0125	0.0235
100	20	0.05	0.1	0.152
500	50	1.25	6.25	7.51
1000	100	10	100	110.02

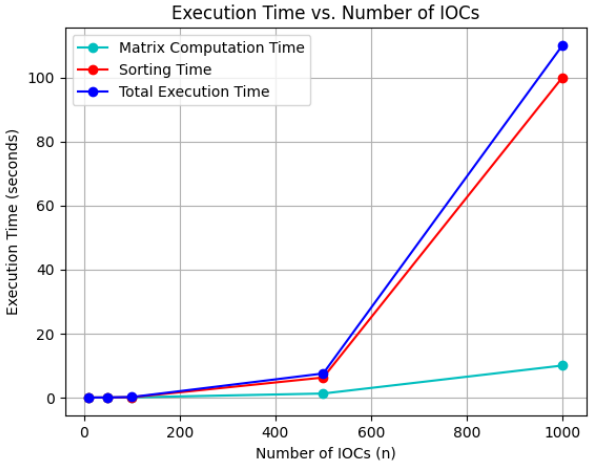

Figure 4. Breakdown of execution time of algorithm for matrix computation and sorting

Both sorting and matrix computation are extremely fast with up to 100 IoCs and take less than a tenth of a second. The whole raking process is done in milliseconds or fractions of a second and is therefore, fast. It suits best with real-time threat intelligence. As the number of IoCs and ransomware families rises, matrix-vector multiplication time scales linearly. Even at 1000 IoCs, the matrix operation remains efficient at just 10 seconds, showing that this component is computationally manageable, even in large-scale scenarios. Sorting becomes the dominant cost at larger values of 'n' as shown in Figure 4. At 1000 IoCs, sorting alone takes approximately 100 seconds, which is 90% of the total execution time. The matrix-based score computation component of the algorithm is proven to be highly efficient and scalable, reinforcing the algorithm's suitability for real-time use cases. The breakdown helps pinpoint performance bottlenecks, allowing targeted optimization without redesigning the entire algorithm.

For the overall space complexity, the primary storage components include the parsed data and computed scores. Given that $|RansomwareNames| \approx m$, $|IOCs| \approx n$, the total space complexity is $O(n.m)$. The results are summarized in Table 7. The table indicates that even if you had 1000 IoCs and 100 ransomware families, total space usage would remain low, around 100,000 units. The algorithm is thus applicable to real-life systems having moderate memory resources. In addition, space growth is easily predictable and linear with input size. This will make it scalable and efficient for real-time or batch threat analysis scenarios. Figure 5 shows the relationship of increase in memory usage with the number of indicators in the IoC ranking algorithm.

The space usage–the size of the IoC occurrence matrix and other lists–increases regularly and reliably as more IoCs are added. This trend is consistent with the linear scalability of the algorithm with respect to the number of IoCs as the number of ransomware families remains proportional. This graphical display reveals the efficiency of the algorithm and the ability to scale it, especially in the real-world cybersecurity scenario where hundreds or thousands of IoCs would be able to be processed at the same time. As indicated, the figure indicates that even with 1000 IoCs there is still room for the service to be deployed on typical computing infrastructure, and space usage is acceptable.

Table 7. Space complexity evaluation of the proposed algorithm

No. of IoCs (n)	No. of families (m)	Matrix space (n × m)	Lists space (3n)	Total space
10	5	50	30	80
50	10	500	150	650
100	20	2000	300	2300
500	50	25000	1500	26500
1000	100	100000	3000	103000

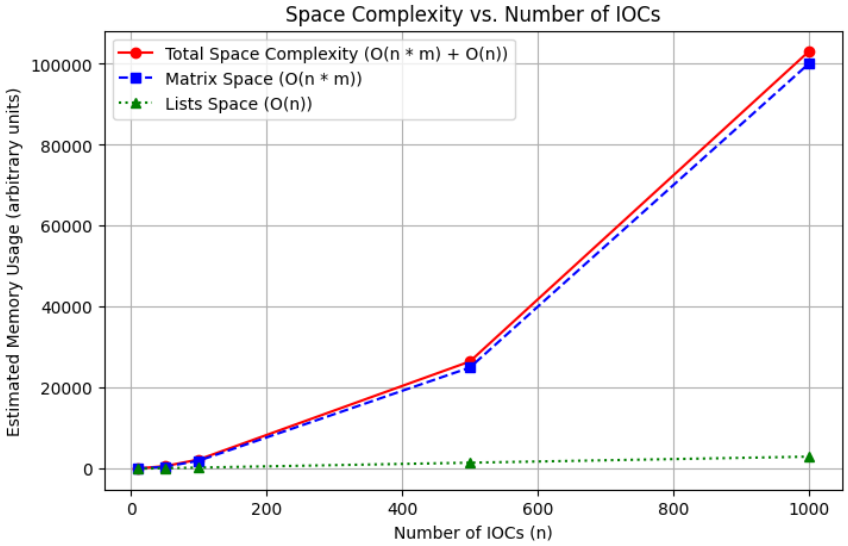

Figure 5. Space complexity growth with increasing number of IoCs

4.3. Comparative study

The proposed IoC ranking technique is compared with existing IoC prioritization techniques as presented in Table 8. The comparison shows that traditional approaches such as comprehensive assessment and rating of IoCs via CADE algorithm (CARIOCA) and FeedRank are primarily based on feed-level aggregation or heuristic rules. Although these techniques reduce the amount of non-relevant data, they have not proven effective in detecting ransomware-related activity. In contrast, graph-based or machine learning (ML) techniques, such as graph neural network (GNN)-based models, have higher detection power but lower interpretability and high computational cost. The proposed approach addresses these challenges by employing a simple weight-based approach based on frequency of occurrence, static and dynamic IoC analysis. Efficient and reliable prioritization can be done by highlighting the high impact IoCs from experimental results. Besides, it is transparent and has a high discriminative ability, and has predictable time and space complexity. Overall, the comparison demonstrates that the proposed approach strikes a realistic trade-off between efficiency, interpretability, and detection accuracy, which is appropriate for practice applications to ransomware detection.

Table 8. Comparative study of proposed method with the related existing state of the art

Aspect	CARIOCA (CADE-based) [40]	FeedRank [41]	STIX/graph-based approaches [42]	ML/GNN-based approaches [43]	Proposed IoC ranking method
Primary objective	Reduce noise in shared IoCs	Rank threat intelligence feeds	Enrich IoCs using graph context	Learn complex IoC patterns	Prioritize ransomware-specific IoCs
Analysis technique	Heuristic multi-score aggregation	Feed-level PageRank	Knowledge graph traversal	Supervised / semi-supervised learning	Hybrid static & dynamic IoC frequency + weight-based scoring
Interpretability	Medium	Medium	Low (graph complexity)	Low (black-box models)	High (transparent score contribution per IoC)
Time complexity	$O(N)$	$O(N + T \cdot E)$	$O(V + E)$	$O(\text{Epochs} \cdot (V + E))$	$O(n \cdot m)$ + sorting (optimizable to $O(n \log n)$)
Space complexity	$O(N)$	$O(N)$	$O(V + E)$	$O(V + E)$ + model parameters	$O(n \cdot m)$, bounded and predictable
Computational overhead	Low	Moderate	High	Very high	Moderate and controllable
Suitability for real-time detection	High	Moderate	Low	Low	High (after sorting optimization)
Ability to distinguish ransomware behaviors	Limited	Indirect	Context-dependent	High (if trained well)	Very High (clear score separation observed)
Experimental outcome	Feed noise reduction	Reliable feed ranking	Improved confidence	Improved accuracy	Clear prioritization of high-impact ransomware IoCs (≈99.99)

5. CONCLUSION

This research evaluated and ranked the most prevalent ransomware variants using static and dynamic behavioral indicators to suggest a structured approach to ransomware IoCs evaluation and ranking. The findings show that certain IoCs are more effective in detection, as they are highly correlated with ransomware activity. The most important functions for the prevention of recovery were shadow copy deletion and connect to Tor2web with highest rank score (99.99997). Other important indicators included packed/obfuscated (99.99546) and file encryption (99.75273), because they are often used to get the payload to run or to evade detection. Some indicators, less prevalent in today's ransomware variants such as locks screen (73.11), indicate old-school techniques. The weighted ranking system developed in this study can serve as a foundation for automated threat detection systems that can prioritize responses according to the behavioral evidence and degree of trustworthiness in the IoC.

FUNDING INFORMATION

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. No funding was raised or provided for the conduct of this study.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Krishna Prasad D.	✓	✓		✓	✓	✓			✓					
Subramanya														
Prasanna Kumar H.	✓	✓		✓		✓	✓		✓	✓	✓	✓	✓	
Ramakrishna														

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The author declares that there are no known conflicts of interest associated with this publication. There are no financial or personal relationships that could inappropriately influence or bias the content of this work.

INFORMED CONSENT

This section not applicable, as this study did not involve human participants or personally identifiable information.

ETHICAL APPROVAL

This section not applicable, as this study did not involve human or animal subjects and used only non-sensitive computational or public data.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] B. Yamany, M. S. Elsayed, A. D. Jurcut, N. Abdelbaki, and M. A. Azer, "A holistic approach to ransomware classification: leveraging static and dynamic analysis with visualization," *Information*, vol. 15, no. 1, Jan. 2024, doi: 10.3390/info15010046.
[2] G. Hull, H. John, and B. Arief, "Ransomware deployment methods and analysis: views from a predictive model and human responses," *Crime Science*, vol. 8, no. 1, Dec. 2019, doi: 10.1186/s40163-019-0097-9.

[3] J. A. H. Silva, L. I. B. López, Á. L. V. Caraguay, and M. H. -Álvarez, "A survey on situational awareness of ransomware attacks—detection and prevention parameters," *Remote Sensing*, vol. 11, no. 10, May 2019, doi: 10.3390/rs11101168.

[4] S. Adam, "The state of ransomware 2024," *sophos.com*. 2024. [Online]. Available: https://www.sophos.com/en-us/blog/the-state-of-ransomware-2024

[5] M. Elgan, "Roundup: the top ransomware stories of 2024," *ibm.com*. 2024. [Online]. Available: https://www.ibm.com/think/insights/roundup-the-top-ransomware-stories-of-2024

[6] M. Lang, L. Connolly, P. Taylor, and P. J. Corner, "The evolving menace of ransomware: a comparative analysis of pre-pandemic and mid-pandemic attacks," *Digital Threats: Research and Practice*, vol. 4, no. 4, pp. 1–22, Dec. 2023, doi: 10.1145/3558006.

[7] F. A. Aboaoja, A. Zainal, F. A. Ghaleb, B. A. S. Al-Rimy, T. A. E. Eisa, and A. A. H. Elnour, "Malware detection issues, challenges, and future directions: a survey," *Applied Sciences*, vol. 12, no. 17, Aug. 2022, doi: 10.3390/app12178482.

[8] S. Razaulla *et al.*, "The age of ransomware: a survey on the evolution, taxonomy, and research directions," *IEEE Access*, vol. 11, pp. 40698–40723, 2023, doi: 10.1109/ACCESS.2023.3268535.

[9] H. J. Hadi, M. A. Riaz, Z. Abbas, and K. U. Nisa, "Cyber threat intelligence model: an evaluation of taxonomies and sharing platforms," in *Big Data Analytics and Intelligent Systems for Cyber Threat Intelligence*, New York: River Publishers, 2023, pp. 3–33, doi: 10.1201/9781003373384-2.

[10] I. Chaudhary and S. Adhikari, "Ransomware detection using machine learning techniques," *Researcher CAB: A Journal for Research and Development*, vol. 3, no. 1, pp. 96–114, Aug. 2024, doi: 10.3126/rcab.v3i1.68424.

[11] K. Paine, O. Whitehouse, J. Sellwood, and A. Shaw, "Indicators of compromise (IoCs) and their role in attack defence," *RFC Editor*, Aug. 2023, doi: 10.17487/RFC9424.

[12] T. Dargahi, A. Dehghantanha, P. N. Bahrami, M. Conti, G. Bianchi, and L. Benedetto, "A cyber-kill-chain based taxonomy of crypto-ransomware features," *Journal of Computer Virology and Hacking Techniques*, vol. 15, no. 4, pp. 277–305, Dec. 2019, doi: 10.1007/s11416-019-00338-7.

[13] M. Keshavarzi and H. R. Ghaffary, "I2CE3: a dedicated and separated attack chain for ransomware offenses as the most infamous cyber extortion," *Computer Science Review*, vol. 36, May 2020, doi: 10.1016/j.cosrev.2020.100233.

[14] I. Kara and M. Aydos, "The rise of ransomware: forensic analysis for windows based ransomware attacks," *Expert Systems with Applications*, vol. 190, Mar. 2022, doi: 10.1016/j.eswa.2021.116198.

[15] D. P. F. Möller, "Ransomware attacks and scenarios: cost factors and loss of reputation," in *Guide to Cybersecurity in Digital Transformation*, Cham, Switzerland: Springer, 2023, pp. 273–303, doi: 10.1007/978-3-031-26845-8_6.

[16] D. Kim, D. Mirsky, A. M. -Kupaei, and R. Barua, "A hybrid static tool to increase the usability and scalability of dynamic detection of malware," in *2018 13th International Conference on Malicious and Unwanted Software (MALWARE)*, Oct. 2018, pp. 115–123, doi: 10.1109/MALWARE.2018.8659373.

[17] M. M. Hasan and M. M. Rahman, "RansHunt: a support vector machines based ransomware analysis framework with integrated feature set," in *2017 20th International Conference of Computer and Information Technology*, Dec. 2017, pp. 1–7, doi: 10.1109/ICCITECHN.2017.8281835.

[18] D. Vidyarthi, C. Kumar, and S. Rakshit, "Identifying ransomware - specific properties using static analysis of executables," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 8, no. 4, pp. 358–366, Apr. 2019, doi: 10.17148/IJARCCE.2019.8461.

[19] A. AlSabeh, H. Safa, E. B. -Harb, and J. Crichigno, "Exploiting ransomware paranoia for execution prevention," in *ICC 2020 - 2020 IEEE International Conference on Communications*, Jun. 2020, pp. 1–6, doi: 10.1109/ICC40277.2020.9149005.

[20] R. Almohaini, I. Almomani, and A. AlKhayer, "Hybrid-based analysis impact on ransomware detection for android systems," *Applied Sciences*, vol. 11, no. 22, Nov. 2021, doi: 10.3390/app112210976.

[21] F. Mercaldo, "A framework for supporting ransomware detection and prevention based on hybrid analysis," *Journal of Computer Virology and Hacking Techniques*, vol. 17, no. 3, pp. 221–227, Sep. 2021, doi: 10.1007/s11416-021-00388-w.

[22] M. A. Ayub, A. Siraj, B. Filar, and M. Gupta, "RWArmor: a static-informed dynamic analysis approach for early detection of cryptographic windows ransomware," *International Journal of Information Security*, vol. 23, no. 1, pp. 533–556, Feb. 2024, doi: 10.1007/s10207-023-00758-z.

[23] A.-R. Belea, "Methods for detecting malware using static, dynamic and hybrid analysis," in *Proceedings of the International Conference on Cybersecurity and Cybercrime*, May 2023, pp. 258–265, doi: 10.19107/CYBERCON.2023.34.

[24] S. Poudyal and D. Dasgupta, "AI-powered ransomware detection framework," in *2020 IEEE Symposium Series on Computational Intelligence*, Dec. 2020, pp. 1154–1161, doi: 10.1109/SSCI47803.2020.9308387.

[25] A. Y. Prasetya, K. I. Aini, and C. Lim, "Comparative analysis of attack behavior patterns in petya, cryptinfinite, and locky ransomware using hybrid analysis," in *2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity*, Aug. 2023, pp. 29–34, doi: 10.1109/ICoCICs58778.2023.10276477.

[26] B. Fiore, K. Ha, L. Huynh, J. Falcon, R. Vendiola, and Y. Li, "Security analysis of ransomware: a deep dive into wannacry and locky," in *2023 IEEE 13th Annual Computing and Communication Workshop and Conference*, Mar. 2023, pp. 285–294, doi: 10.1109/CCWC57344.2023.10099114.

[27] M. Wecksten, J. Frick, A. Sjostrom, and E. Jarpe, "A novel method for recovery from crypto ransomware infections," in *2016 2nd IEEE International Conference on Computer and Communications*, Oct. 2016, pp. 1354–1358, doi: 10.1109/CompComm.2016.7924925.

[28] N. A. Hassan, *Ransomware revealed: a beginner's guide to protecting and recovering from ransomware attacks*. Berkeley, California: Apress, 2019, doi: 10.1007/978-1-4842-4255-1.

[29] M. Verma, P. Kumarguru, S. B. Deb, and A. Gupta, "Analysing indicator of compromises for ransomware: leveraging IoCs with machine learning techniques," in *2018 IEEE International Conference on Intelligence and Security Informatics*, Nov. 2018, pp. 154–159, doi: 10.1109/ISI.2018.8587409.

[30] J. Bang, J. N. Kim, and S. Lee, "Entropy sharing in ransomware: bypassing entropy-based detection of cryptographic operations," *Sensors*, vol. 24, no. 5, Feb. 2024, doi: 10.3390/s24051446.

[31] R. Brewer, "Ransomware attacks: detection, prevention and cure," *Network Security*, vol. 2016, no. 9, pp. 5–9, Sep. 2016, doi: 10.1016/S1353-4858(16)30086-1.

[32] P. O'Kane, S. Sezer, and D. Carlin, "Evolution of ransomware," *IET Networks*, vol. 7, no. 5, pp. 321–327, Sep. 2018, doi: 10.1049/iet-net.2017.0207.

[33] S. Kok, A. Abdullah, N. Z. Jhanjhi, and M. Supramaniam, "Prevention of crypto-ransomware using a pre-encryption detection algorithm," *Computers*, vol. 8, no. 4, Nov. 2019, doi: 10.3390/computers8040079.

[34] Y. Yilmaz, O. Cetin, B. Arief, and J. H. -Castro, "Investigating the impact of ransomware splash screens," *Journal of Information Security and Applications*, vol. 61, Sep. 2021, doi: 10.1016/j.jisa.2021.102934.

[35] Y. Jin, M. Tomoishi, S. Matsuura, and Y. Kitaguchi, "A secure container-based backup mechanism to survive destructive ransomware attacks," in *2018 International Conference on Computing, Networking and Communications*, Mar. 2018, pp. 1–6, doi: 10.1109/ICCNC.2018.8390376.
[36] E. Kolodenker, W. Koch, G. Stringhini, and M. Egele, "PayBreak: defense against cryptographic ransomware," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, Apr. 2017, pp. 599–611, doi: 10.1145/3052973.3053035.
[37] L. Yang, A. Ciptadi, I. Laziuk, A. Ahmadzadeh, and G. Wang, "BODMAS: an open dataset for learning based temporal analysis of PE malware," in *2021 IEEE Security and Privacy Workshops*, May 2021, pp. 78–84, doi: 10.1109/SPW53761.2021.00020.
[38] T.-N. To *et al.*, "On the effectiveness of adversarial samples against ensemble learning-based windows PE malware detectors," *International Journal of Information Security*, vol. 24, no. 1, Feb. 2025, doi: 10.1007/s10207-024-00969-y.
[39] S. Mehnaz, A. Mudgerikar, and E. Bertino, "RWGuard: a real-time detection system against cryptographic ransomware," in *Research in Attacks, Intrusions, and Defenses (RAID 2018)*, Cham, Switzerland: Springer, 2018, pp. 114–136, doi: 10.1007/978-3-030-00470-5_6.
[40] P. Delvecchio, S. Galantucci, A. Iannacone, and G. Pirlo, "CARIOCA: prioritizing the use of IoC by threats assessment shared on the MISP platform," *International Journal of Information Security*, vol. 24, no. 2, Apr. 2025, doi: 10.1007/s10207-025-01006-2.
[41] S.-S. Chen, R.-H. Hwang, A. Ali, Y.-D. Lin, Y.-C. Wei, and T.-W. Pai, "Improving quality of indicators of compromise using STIX graphs," *Computers & Security*, vol. 144, Sep. 2024, doi: 10.1016/j.cose.2024.103972.
[42] R. Meier, C. Scherrer, D. Gugelmann, V. Lenders, and L. Vanbever, "FeedRank: a tamper-resistant method for the ranking of cyber threat intelligence feeds," in *2018 10th International Conference on Cyber Conflict (CyCon)*, May 2018, pp. 321–344, doi: 10.23919/CYCON.2018.8405024.
[43] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," *Computers & Security*, vol. 81, pp. 123–147, Mar. 2019, doi: 10.1016/j.cose.2018.11.001.

BIOGRAPHIES OF AUTHORS

Krishna Prasad D. Subramanya received the B.Eng. degree in Information Science and Engineering and the M.Tech. degrees in Computer Science and Engineering from Visvesvaraya Technological University, India, in 2012 and 2014, respectively. Currently, he is working as an assistant professor at the Department of Computer and Communication Engineering, Nitte (Deemed to be University), Karnataka, India. His research interests include computer networks, cryptography and network security, cyber security, and malware analysis. He can be contacted at email: krishna270572@gmail.com.

Prasanna Kumar H. Ramakrishna received M.Tech. degree in Computer Science from National Institute of Technology, Surathkal and Ph.D. from Visvesvaraya Technological University, India, in 2004 and 2018 respectively. Currently, he is working as professor and head of Department of Computer Science and Engineering, PES Institute of Technology and Management, Shivamogga, Karnataka, India. He is having 26+ years of teaching experiences. His area of interest includes cryptography and network security, theory of computation, data structures, and algorithms. He has published over 30 research papers in international journals and conferences. He can be contacted at email: hrpbhat@gmail.com.