

Mitigating ransomware attacks through cyber threat intelligence and machine learning

Mamady Kante, Vivek Sharma, Keshav Gupta

Department of Computer Science and Engineering, Sharda University, Greater Noida, India

Article Info

Article history:

Received Nov 3, 2023

Revised Dec 5, 2023

Accepted Dec 10, 2023

Keywords:

Cyber threat intelligence

Machine learning

Malware

Ransomware

Static analysis

ABSTRACT

In the face of escalating cyber threats, particularly the rampant and sophisticated nature of ransomware attacks, organizations are compelled to adopt a proactive and multi-faceted strategy for mitigation. The fusion of machine learning (ML) algorithms enables the system to dynamically adapt and evolve in response to evolving attack vectors and tactics employed by cybercriminals. This paper presents a comprehensive approach that synergistically integrates ML and cyber threat intelligence (CTI) to fortify defenses against ransomware assaults. The proposed methodology incorporates three distinct machine learning techniques, namely random forest (RF), extreme gradient boosting (XGBoost), and adaptive boosting (AdaBoost). Empirical evidence derived from the study affirms the efficacy of this approach in effectively discriminating between malicious and ransomware, achieving a notable identification rate of 98.55%. The incorporation of CTI enhances the strategic posture by providing actionable insights into the threat landscape. The proposed focuses on identifying and neutralizing ransomware, aligning with contemporary cybersecurity imperatives, offering a proactive defense against ransomware attacks, ultimately safeguarding critical assets, and preserving the integrity of digital ecosystems.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Mamady Kante

Department of Computer and Engineering, Sharda University

201310 Greater Noida, Uttar Pradesh, India

Email: mhdkante@gmail.com

1. INTRODUCTION

The rate at which technology is evolving, demonstrated by internet, has facilitated the enhancement of human existence in terms of convenience and comfort. Nevertheless, this evolution has engendered a reliance on the internet, positioning it as the nucleus of our daily lives. The imperative for constant connectivity, ubiquitously and instantaneously, has intensified the intricacies of the information system, leading to multiple vulnerabilities. Presently, the escalating number and interconnectivity of computers, coupled with the escalating complexity of systems and their facile extensibility, contribute to the escalating incidence of malware infections daily. A prominent and perilous menace to organizational integrity is ransomware [1], [2]. This insidious form of malicious software effectively takes control of a computer system, encrypts files on the hard drive, or forces the computer to shut down, demanding a ransom in return for restoring normal functionality and obstructing user access to the system. This form of cybercrime has grown exponentially in recent years, targeting businesses, healthcare institutions, government agencies, and individuals [3]. The motivation behind ransomware attacks is often financial gain, and the consequences can be catastrophic, ranging from financial losses to reputational damage. The operation of ransomware is described in Figure 1.

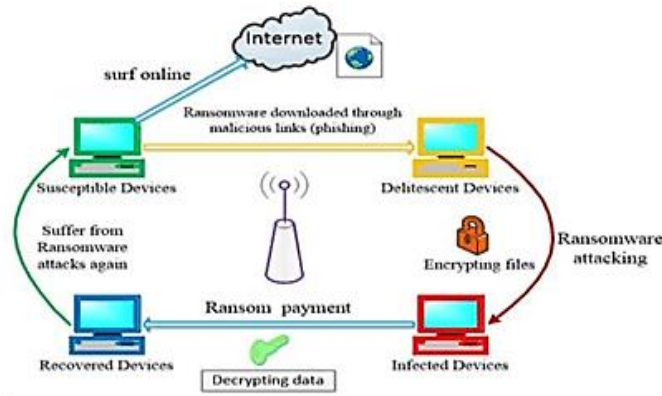


Figure 1. Ransomware attacks operation [4]

The tenuous state of healthcare delivery amid the COVID-19 pandemic was intricately linked to a surge in ransomware attacks during 2020 [5]. Consequently, medical institutions experienced severe disruptions in healthcare services, accompanied by enduring ramifications. Throughout 2020, a staggering 550,000 ransomware incidents were recorded daily, yielding cyber attackers an estimated 1.5 trillion dollars. In 2021, a notable escalation occurred, with 66% of monitored organizations falling prey to ransomware assaults, a substantial increase from the 37% reported in 2020 [6].

For this reason, many researchers have devoted their time to exploring the impact of machine learning (ML) in Ransomware mitigation. Notably, [7], [8] demonstrated how well ML models can identify ransomware activities based on patterns in network data. In a parallel vein, [9] investigated artificial intelligent (AI)-powered anomaly detection techniques to expose ransomware activity in its earliest stages. The collective contributions of [10]–[12] contribute to the expanding collection of knowledge about the use of ML for ransomware defense. All of these academic investigations highlight how ML and AI can be revolutionary tools for strengthening organizational structures' resistance to ransomware.

Subsequently, various authors have explored diverse approaches to address the pervasive threat of ransomware over the internet. Manoj and Rani [13], employed fuzzy neural and neural networks, achieving 98% accuracy for fuzzy neural and 95% for neural networks. Despite its success, the fuzzy neural network, being rule-based, faces limitations in detecting new threats. In an attempt to automate incident response, [14] suggested using actionable cyber threat intelligence (CTI). However, this approach, while effective, is time-consuming in analyzing reports and log files.

The paper titled “utilizing artificial intelligence for the detection, examination, and alleviation of malicious software” reported a 98% accuracy in malware detection using AI. Nevertheless, updating training data proves challenging due to the time-consuming nature of human expert analysis. Using ML techniques, [15] improved the detection rate of ransomware through enhanced file entropy analysis, reaching an 85.17% accuracy. Nevertheless, a challenge is the constantly changing nature of ransomware, leading to high false positive rates. AlAhmadi and Martinovic [16], Ren *et al.* [17] created a dataset with dynamic features for ransomware detection, utilizing gradient-boosted regression trees with 98% accuracy. However, reliance on sandbox-extracted data affects reliability. Bae *et al.* [18] designed an offensive system using CTI enhanced with counterattack and counterintelligence, offering insights into the motivation behind attacks but displaying limitations against zero-day attacks.

Chakkaravarthy *et al.* [19] introduced the Social Leopard algorithm-based detection of intrusion system, effectively limiting ransomware activity with improved detection metrics compared to traditional methods. However, its resource-intensive nature makes it unsuitable for real-time detection. Moreira *et al.* [20] proposed an improved file entropy analysis, enhancing the probability of identifying ransomware by recognizing files rather than running programs. The study in [21]–[25] focused on API calls for ransomware detection, achieving high accuracy using support vector machine (SVMs). Asrafi *et al.* [26] presented a dynamic feature dataset for ML-based ransomware detection, achieving high accuracy with resilience in historical data. Khammas *et al.* [27] explored various ML algorithms and platforms for ransomware identification, emphasizing the importance of ML in predicting and analyzing ransomware. Khammas *et al.* [28] introduced a stacking ML model combining boosting and stacking techniques for malware classification. While the multi-layer perceptron (MLP)-Adaboost classifier demonstrated superior performance, future investigations may explore alternative feature selection methods.

After reviewing different approaches toward overcoming ransomware challenges on the internet by various authors, clear and precise methods are not clearly stated by the authors on how to overcome the menace

completely, the ransomware landscape proved dynamic, with attackers adopting sophisticated techniques to bypass traditional security measures. The use of advanced evasion tactics and the continuous evolution of malware strains make it challenging for organizations to stay ahead of potential threats. As a result, there is a critical need for proactive and adaptive strategies to prevent, detect, and respond to ransomware incidents effectively. Thus, the concept of mitigating ransomware attacks through ML and CTI came into the limelight. The study's remaining section is organised as follows: the paper's approach is identified in section 2, discussed and analysed in section 3, and the proposed method's scope and conclusion are drawn in section 4.

2. METHOD

This study introduces a ransomware identification method using ML and integrated CTI feed data. Figure 2 outlines the procedural framework, where API sequences are extracted from specimens and used to generate n-gram sequences [29]. Binary input vectors are created based on the presence or absence of n-grams, with weights assigned according to class frequency. A comprehensive explanation of class frequency (CF) follows. Three ML models are implemented with the weighted vectors to create a classification model. This model categorizes unknown binary samples as benign files or ransomware. The evaluation utilized a personal computer with a Core i5 CPU and 12GB RAM, testing on Windows 10 (64-bit) across two distinct operating systems [30].

The suggested approach, depicted in Figure 2(a), involves a process starting with a dataset of executable files containing ransomware and safe versions. The dataset, consisting of 62,486 executable and dynamic-link library (DLL) files, includes 20,000 instances of ransomware across diverse families and benign counterparts. Training and testing data are separated from the dataset, followed by a three-step preprocessing stage: feature extraction, Ngram sequence analysis, and Class_Frequency-NonClas_Frequency assessment. The subsequent steps include feature selection, model creation, and data classification. Criteria limit each file to 1 MB, and files exceeding this size are excluded. Ransomware files are sourced from CTI feeds virus total, ShieldFS, virus share, and Kaggle, while benign files are obtained from the Windows platform.

2.1. Preprocessing and features extraction

The preprocessing of files involves a formalized three-step approach with implicit stages, utilizing an attribute extractor module to extract attributes from executable files. N-gram feature extraction is employed to analyze files, using substrings of varying lengths referred to as n-grams. Optimal accuracy is achieved with an n value of 4, though increasing n diminishes accuracy. The study focuses on N-gram features in short malware signatures and prevalent ransomware files, limiting the feature set to 100 features through the collaborative filtering and neural collaborative filtering (CF-NCF) algorithm. Given the dataset's numerous n-gram features, an attribute selection process is crucial to recognise condensed set of significant features. Gain ratio (GR) technique is applied to carefully choose a subset of 15 features, reducing dimensionality and enhancing the classifier's predictive model [29], [31], [32].

2.2. CF-NCF

Figure 2(b) demonstrates the process of constructing input vectors for a ML model. N-grams, derived from the analyzed recovered API log, are assigned binary values (1 for presence, 0 for absence) in a vector. These values are further weighted by their corresponding CF-NCF values. The ML model is then built using the resulting weighted n-gram vector. Evaluation of classification models involves the CF-NCF measure, employing term frequency inverse document frequency (TF-IDF) in (1) to (3) for simplified computation.

$$TF(a, b) = 0.5 + \frac{0.5 * f(a, b)}{\max(a, b) : a \in b} \quad (1)$$

$$idf(a, B) = \log \frac{|B|}{|\{b \in B : a \in b\}|} \quad (2)$$

$$TF - IDF(a, b, f) = tf(a, b) * idf(a, B) \quad (3)$$

when;

F (a, b) = frequency of recurrence of the term "a" in file "b".

|b ∈ B: a ∈ b| = "T" containing documents in corpus "B" are regarded as cardinal.

|B| = the total number of log files present in the dataset.

Drawing from the TF-IDF, the study, introduces class and non-class frequency, prioritizing features specific to each class over traditional methods. This approach computes weights for elements within a class, enhancing predictive accuracy for classification. Calculating CF-NCF is done using (2):

$$CF(s, C) = f(s, C) \tag{4}$$

$$NCF(s, N) = \log\left(\frac{1}{0.001+f(s,N)}\right) \tag{5}$$

$$CF - CNF = CF * NCF \tag{6}$$

where:

- S = Ngram
- f(s, C) = frequency with which Ngram appears in Ngram sequence
- C = Ngram sequence
- N = collection of Ngram sequences

Value to mitigate the possibility of division by zero = 0.01.

The experimental approach in the study leverages class and non-class frequency variables to enhance the accuracy of ransomware detection. The unique N-gram sequence "C" is associated with ransomware, while "N" is linked to benign classes. The goal is to categorize unknown binary samples, distinguishing between malicious and benign files. This classification employs vectors with weights derived from a training dataset, facilitating the categorization of exec files. Three distinct ML models, trained on the dataset, were evaluated for classification precision using a test dataset.

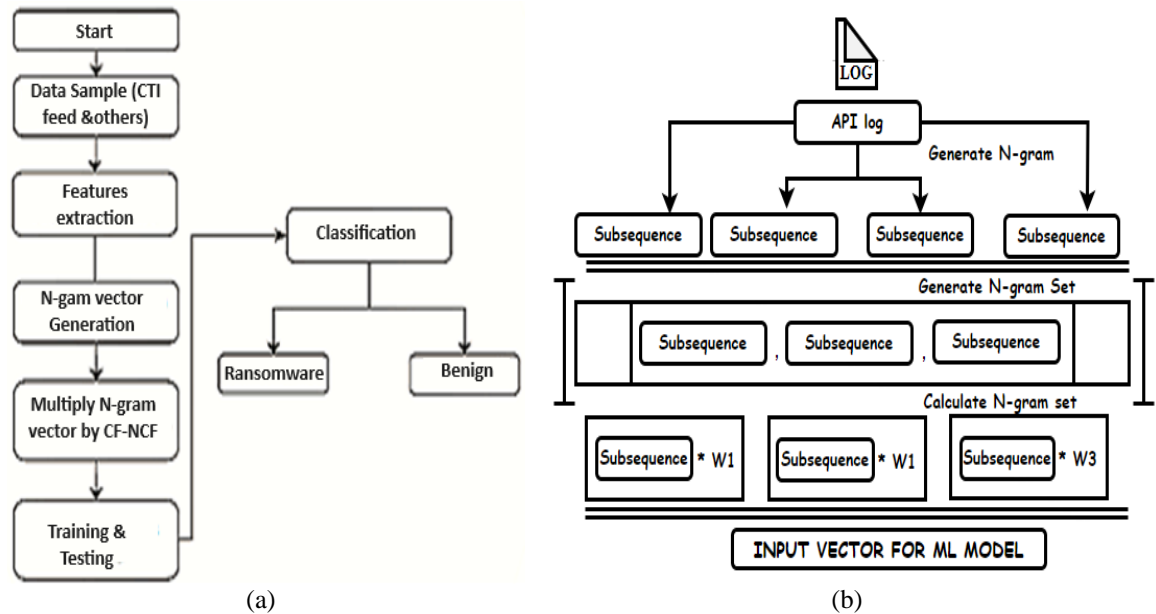


Figure 2. Outlines the procedural framework (a) proposed method workflow and (b) input generation for ML model

2.3. ML algorithm

In ML, algorithms are broadly categorized into prediction, regression, and classification. The study focuses on binary categorization methods adaptable for multiclass classifications [32], [33]. The selected algorithms for experimentation include XGBoost, an extension of gradient boosting with regularization for enhanced computational efficiency; random forest (RF), a bagging-based ensemble method reducing overfitting and increasing robustness; and AdaBoost, an adaptive boosting algorithm that sequentially trains weak learners to create a strong and adaptive ensemble [34]. The research explores the theoretical foundations of each algorithm, shedding light on the mechanisms behind their success in diverse machine-learning applications.

3. RESULTS AND DISCUSSION

The suggested approach helps to distinguish between malicious and safe software. Several tests were carried out to evaluate multiple ML performance measures to demonstrate the efficiency of the class and non-class frequency. Categorization accuracy, false_positive, precision, true_negative, recall, F1-score, and true_positive, are only a few of these indicators. The aforementioned metrics were computed for every machine-learning method that was used using the recommended formulas.

$$\text{Recall} = \frac{\text{TruePositive}}{(\text{TruePositive}+\text{FalseNegative})} \quad (7)$$

$$\text{Precision} = \frac{\text{TruePositive}}{(\text{TruePositive}+\text{FalsePositive})} \quad (8)$$

$$\text{Accuracy} = \frac{(\text{TruePositive}+\text{TrueNegative})}{(\text{TruePositive}+\text{TrueNegative}+\text{FalsePositive}+\text{FalseNegative})} \quad (9)$$

$$F1_{\text{score}} = 2 \times \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (10)$$

$$\text{TruePositive} = \frac{\text{TruePositive}}{\text{TruePositive}+\text{FalseNegative}} \quad (11)$$

$$\text{FalsePositive} = \frac{\text{FalsePositive}}{\text{FalsePositive}+\text{TrueNegative}} \quad (12)$$

$$\text{TrueNegative} = \frac{\text{TrueNegative}}{\text{TrueNegative}+\text{FalsePositive}} \quad (13)$$

$$\text{FalseNegative} = \frac{\text{TrueNegative}}{\text{TrueNegative}+\text{TruePositive}} \quad (14)$$

The dataset underwent partitioning to address the potential imbalance, creating 80% for training and 20% for testing. Fair distribution was achieved through random sampling, ensuring equitable representation of benign and ransomware instances. Implementing non-class frequency techniques on training and testing data results in distinct metrics for ML performance evaluation pre and post-incorporation of class and non-class frequency as displayed in Table 1. Table 2 present a comparison results between our proposed method and others approaches.

Table 1. Results of simulation

State	Metric	ML-algorithm			
		XGBoost	RF	AdaBoost	
Before CF-NC	AC	0.9583	0.9712	0.8883	
	True-P(+)	0.9833	0.9943	0.8866	
	False-P(+)	0.0666	0.05	0.11	
	True-Ne(-)	0.9333	0.95	0.89	
	False-Ne(-)	0.0166	0.0067	0.1133	
	Precision	0.9365	0.959	0.8896	
	Recall	0.9833	0.9943	0.8866	
	F-score	0.9593	0.9728	0.8881	
	After CF-NCF	A	0.9711	0.9856	0.8985
		True-P(+)	0.9933	0.96	0.8866
False-P(+)		0.05	0.0133	0.09	
True-Ne(-)		0.95	0.9866	0.91	
False-Ne(-)		0.0066	0.04	0.1133	
Precision		0.952	0.9863	0.9078	
Recall		0.9933	0.99	0.8866	
F-score		0.9722	0.9729	0.8971	

The categorization procedure and implementation of the CF-NCF equation were conducted using Python 3 and Google Colab. The outcomes show how well the suggested methodology works to differentiate between safe and malicious code. Notably, applying class and non-class frequency to training and testing data significantly enhances classification accuracy and related metrics. Specifically, the XGBoost, RF, and AdaBoost classifiers experience notable accuracy improvements from 95.83%, 97.12%, and 88.83% to

97.16%, 98.56%, and 89.83%, respectively Figure 3. Figures 4 and 5 illustrate the percentage of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) with pre and post-implementation of CF-NCF. Precision Figure 6, recall Figure 7, and F1_score Figure 8 are also presented with pre- and post-application of CF-NCF.

Figures 3 to 8 illustrates the empirical outcomes of various metrics for evaluating the ML model, such as F1-score, accuracy, recall, and precision. These metrics are presented both before and after the incorporation of class and non-class frequency equations into our comprehensive Threat Intelligence dataset, aiming to augment the overall performance.

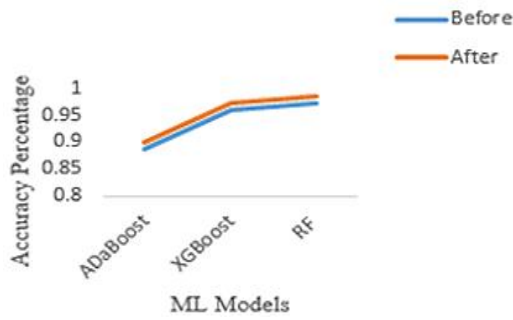


Figure 3. Accuracy before and after CF-NCF

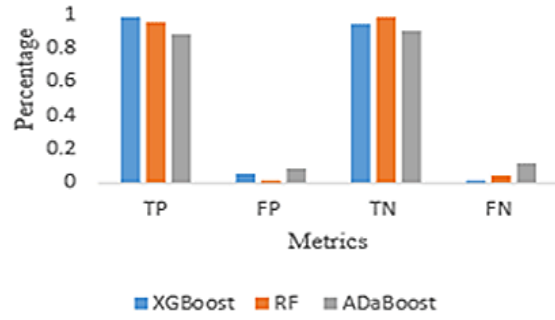


Figure 4. Percentage of TP, TN, FN, FP (Pre- CF-NCF)



Figure 5. Percentage of TP, TN, FN, FP (post- CF-NCF)

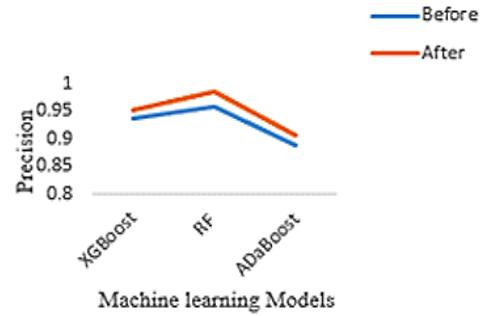


Figure 6. Precision before and after CF-NCF

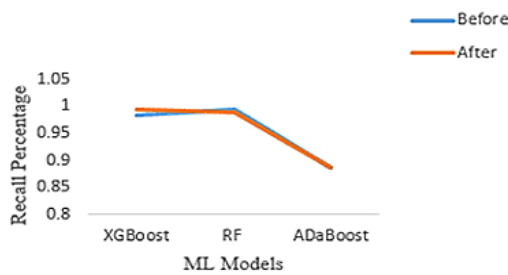


Figure 7. Recall before and after CF-NCF

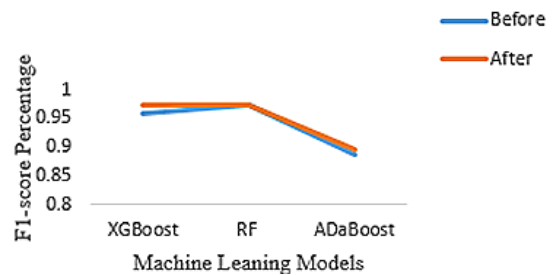


Figure 8. F1-score before and after CF-NCF

Table 2. Comparative analysis between proposed method and others approaches

Method	Method of Analysis	Feature type	ML algorithm	Feature selection method	Accuracy %
Proposed method	Static	N-gram	XGBoost, RF, AdaBoost.	Gain Ratio	98.56 %
Takeuchi <i>et al.</i> [21]	Dynamic	API call	SVM	2-gram	97%
Kim [32]	Dynamic	System call	SVM, SGBD	TF-IDF	96%
Pektaş and Acarman [35]	Dynamic	API-call	Voting experts algorithm	N gram	98%

4. CONCLUSION

Ransomware remains a formidable challenge for individuals and organizations, necessitating innovative approaches for risk mitigation. To address this, the paper proposed, emphasizing static analysis to surmount the limitations of dynamic analysis. This method involves extracting byte-level data characteristics directly, enhancing detection capabilities through the application of CF-NCF alongside n-gram characteristics. In the classification phase, the proposed approach leverages three distinct ML methods namely; RF, XGBoost, and AdaBoost. Notably, the result that comes from the simulation shows an impressive 98.56% identification rate for RFs which shows a slight improvement compared to other approaches. To fortify research efforts for future challenges, it is imperative to devise a novel methodology for designing an autonomously updated dataset, ensuring the longevity and relevance of our research endeavors in the ever-evolving landscape of cybersecurity.




REFERENCES

- [1] B. V. Solms and R. V. Solms, "Cybersecurity and information security – what goes where?," *Information & Computer Security*, vol. 26, no. 1, pp. 2–9, Mar. 2018, doi: 10.1108/ICS-04-2017-0025.
- [2] R. S. Abujassar, M. Sayed, and H. Yaseen, "A new algorithm to enhance security against cyber threats for internet of things application," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 4, pp. 4452–4466, Aug. 2023, doi: 10.11591/ijece.v13i4.pp4452-4466.
- [3] Y. Ayachi, Y. Mellah, M. Saber, N. Rahmoun, I. Kerrakchou, and T. Bouchentouf, "A survey and analysis of intrusion detection models based on information security and object technology-cloud intrusion dataset," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 4, pp. 1607–1614, Dec. 2022, doi: 10.11591/ijai.v11.i4.pp1607-1614.
- [4] W. Liu, "Modeling ransomware spreading by a dynamic node-level method," *IEEE Access*, vol. 7, pp. 142224–142232, 2019, doi: 10.1109/ACCESS.2019.2941021.
- [5] B. Horowitz, "2020 offered a 'perfect storm' for cybercriminals with ransomware attacks costing the industry \$21B," *Fierce Healthcare*, 2021. <https://www.fiercehealthcare.com/tech/ransomware-attacks-cost-healthcare-industry-21b-2020-here-s-how-many-attacks-hit-providers> (accessed Nov. 25, 2023).
- [6] S. Adam, "The state of ransomware 2022," *Sophos News*, 2022. <https://news.sophos.com/en-us/2022/04/27/the-state-of-ransomware-2022/> (accessed Nov. 25, 2023).
- [7] 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, Dec. 2021, doi: 10.3390/app12010172.
- [8] A. Wani and S. Revathi, "Ransomware protection in IoT using software defined networking," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 3166–3175, Jun. 2020, doi: 10.11591/ijece.v10i3.pp3166-3175.
- [9] M. Hassan, F. Abrar, and M. Hasan, *An explainable AI-driven machine learning framework for cybersecurity anomaly detection*. 1st Edition, Routledge, 2023.
- [10] S. Mehnaz, A. Mudgerikar, and E. Bertino, "RWGuard: a real-time detection system Against cryptographic ransomware," in *RAID 2018: Research in Attacks, Intrusions, and Defenses*, 2018, pp. 114–136. doi: 10.1007/978-3-030-00470-5_6.
- [11] 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, doi: 10.3390/iot1020030.
- [12] 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, pp. 1–37, Jan. 2022, doi: 10.1145/3514229.
- [13] M. Manoj and V. G. Rani, "Ransomware classification using fuzzy neural network algorithm," *International journal of health sciences*, pp. 11268–11278, May 2022, doi: 10.53730/ijhs.v6nS2.8026.
- [14] C. Leite, J. den Hartog, D. R. Santos, and E. Costante, "Actionable cyber threat intelligence for automated incident response," in *NordSec 2022: Secure IT Systems*, 2022, pp. 368–385. doi: 10.1007/978-3-031-22295-5_20.
- [15] C.-M. Hsu, C.-C. Yang, H.-H. Cheng, P. E. Setiasabda, and J.-S. Leu, "Enhancing file entropy analysis to improve machine learning detection rate of ransomware," *IEEE Access*, vol. 9, pp. 138345–138351, 2021, doi: 10.1109/ACCESS.2021.3114148.
- [16] B. A. AlAhmadi and I. Martinovic, "MalClassifier: malware family classification using network flow sequence behaviour," in *2018 APWG Symposium on Electronic Crime Research (eCrime)*, May 2018, pp. 1–13. doi: 10.1109/ECRIME.2018.8376209.
- [17] Y. Ren, Y. Xiao, Y. Zhou, Z. Zhang, and Z. Tian, "CSKG4APT: A cybersecurity knowledge graph for advanced persistent threat organization attribution," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–15, 2022, doi: 10.1109/TKDE.2022.3175719.
- [18] S. Il Bae, G. Bin Lee, and E. G. Im, "Ransomware detection using machine learning algorithms," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 18, Sep. 2020, doi: 10.1002/cpe.5422.
- [19] S. S. Chakkaravarthy, D. Sangeetha, M. V. Cruz, V. Vaidehi, and B. Raman, "Design of intrusion detection honeypot using social leopard algorithm to detect IoT ransomware attacks," *IEEE Access*, vol. 8, pp. 169944–169956, 2020, doi: 10.1109/ACCESS.2020.3023764.
- [20] C. C. Moreira, D. C. Moreira, and C. de S. de Sales Jr., "Improving ransomware detection based on portable executable header using xception convolutional neural network," *Computers & Security*, vol. 130, Jul. 2023, doi: 10.1016/j.cose.2023.103265.
- [21] Y. Takeuchi, K. Sakai, and S. Fukumoto, "Detecting ransomware using support vector machines," in *Proceedings of the 47th International Conference on Parallel Processing Companion*, Aug. 2018, pp. 1–6. doi: 10.1145/3229710.3229726.
- [22] M. Scalas, D. Maiorca, F. Mercaldo, C. A. Visaggio, F. Martinelli, and G. Giacinto, "On the effectiveness of system API-related information for Android ransomware detection," *Computers & Security*, vol. 86, pp. 168–182, Sep. 2019, doi: 10.1016/j.cose.2019.06.004.
- [23] K. Cabaj and W. Mazurczyk, "Using software-defined networking for ransomware mitigation: The case of CryptoWall," *IEEE Network*, vol. 30, no. 6, pp. 14–20, Nov. 2016, doi: 10.1109/MNET.2016.1600110NM.
- [24] J. A. Herrera-Silva and M. Hernández-Álvarez, "Dynamic feature dataset for ransomware detection using machine learning algorithms," *Sensors*, vol. 23, no. 3, Jan. 2023, doi: 10.3390/s23031053.
- [25] L. Bekkers, S. van 't Hoff-de Goede, E. Misana-ter Huurme, Y. van Houten, R. Spithoven, and E. R. Leukfeldt, "Protecting your business against ransomware attacks? Explaining the motivations of entrepreneurs to take future protective measures against cybercrimes using an extended protection motivation theory model," *Computers & Security*, vol. 127, Apr. 2023, doi: 10.1016/j.cose.2023.103099.




- [26] N. Asrafi, D. C.-T. Lo, R. M. Parizi, Y. Shi, and Y.-W. Chen, "Comparing performance of malware classification on automated stacking," in *Proceedings of the 2020 ACM Southeast Conference*, Apr. 2020, pp. 307–308. doi: 10.1145/3374135.3385316.
- [27] B. M. Khammas, S. Hasan, N. Nateq, J. S. Bassi, I. Ismail, and M. N. Marsono, "First line defense against spreading new malware in the network," in *2018 10th Computer Science and Electronic Engineering (CEECE)*, Sep. 2018, pp. 113–118. doi: 10.1109/CEECE.2018.8674214.
- [28] B. M. Khammas, A. Monemi, J. Stephen Bassi, I. Ismail, S. Mohd Nor, and M. N. Marsono, "Feature selection and machine learning classification for malware detection," *Jurnal Teknologi*, vol. 77, no. 1, Oct. 2015, doi: 10.11113/jt.v77.3558.
- [29] A. Hussain, M. Asif, M. Bin Ahmad, T. Mahmood, and M. A. Raza, "Malware detection using machine learning algorithms for windows platform," in *Proceedings of International Conference on Information Technology and Applications*, 2022, pp. 619–632. doi: 10.1007/978-981-16-7618-5_53.
- [30] A. Widjajarto, M. Lubis, and V. Ayuningtyas, "Vulnerability and risk assessment for operating system (OS) with framework STRIDE: comparison between VulnOS and Vulnix," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 3, pp. 1643–1653, Sep. 2021, doi: 10.11591/ijeecs.v23.i3.pp1643-1653.
- [31] M. S. Akhtar and T. Feng, "Evaluation of machine learning algorithms for malware detection," *Sensors*, vol. 23, no. 2, Jan. 2023, doi: 10.3390/s23020946.
- [32] C. W. Kim, "NtMalDetect: A machine learning approach to malware detection using native API system calls," *Preprint arXiv.1802.05412*, Feb. 2018.
- [33] B. M. Khammas, "Comparative analysis of various machine learning algorithms for ransomware detection," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 20, no. 1, pp. 43–51, Feb. 2022, doi: 10.12928/telkomnika.v20i1.18812.
- [34] A. M. Bamhdi, I. Abrar, and F. Masoodi, "An ensemble based approach for effective intrusion detection using majority voting," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 2, pp. 664–671, Apr. 2021, doi: 10.12928/telkomnika.v19i2.18325.
- [35] A. Pektaş and T. Acarman, "Malware classification based on API calls and behaviour analysis," *IET Information Security*, vol. 12, no. 2, pp. 107–117, Mar. 2018, doi: 10.1049/iet-ifs.2017.0430.

BIOGRAPHIES OF AUTHORS






Mamady Kante    is an individual of considerable passion and dedication, whose trajectory has traversed both the realms of academia and professional development, with a specific focus on the fields of computer science and cybersecurity. In the year 2015, Mr. Kante achieved the successful culmination of his Bachelor's degree in Computer Science and Engineering from Gamal Abdel Nasser University of Conakry in Guinea. Acknowledging the imperative of remaining at the forefront of the dynamic technological landscape, he opted to further specialize in the domains of cybersecurity and networking. This commitment was manifested through his enrollment in the Master of Technology (MTECH) program at Sharda University. Mamady Kante's aspirations are indicative of a profound commitment to the enhancement of security protocols, active engagement in ethical hacking practices, and a pivotal role in fortifying digital infrastructures. He can be contacted at email: mhdkante@gmail.com.



Dr. Vivek Sharma    graduated from Guru Jambheshwar University in Hisar in 2005 with a B.Tech. in Information Technology, Guru Gobind Singh IP University in New Delhi, India in 2008 with an M.Tech. in Information Technology, and Jamia Millia Islamia in New Delhi, India in 2018 with a Ph.D. in Computer Engineering. His research on MANET, IoT, and networks has resulted in 20 research papers published in prestigious international publications and conferences. He is an active participant in the scholarly debate, actively involved in both research and development. He can be contacted at email: vivek.sharma@sharda.ac.



Dr. Keshav Gupta    after earning a B.Tech. in Computer Science and Engineering from UIET, Kurukshetra University, and an M.Tech. in Software Engineering from UIET, He graduated with a Ph.D. from Delhi Technological University in 2020. He has published extensively in prestigious international journals and conferences on a variety of subjects related to his academic interests in biometric systems, pattern recognition, image processing, and machine learning. He can be contacted at email: keshav.gupta@sharda.ac.in.