Optimized Multi-Task Evolutionary Artificial Neural Network Fostered Crypto-Ransomware Prevention in File-Sharing Scenarios with Encrypted Traffic

Vyom Kulshreshtha
Department of Computer Science & Engineering, Amity University Madhya Pradesh, Gwalior, India
vyom19@gmail.com (corresponding author)

Deepak Motwani
Department of Computer Science & Engineering, Amity University Madhya Pradesh, Gwalior, India
dmotwani@gwa.amity.edu

Pankaj Sharma
Department of Computer Science & Engineering, Eshan College of Engineering, Mathura, India
topankajsharma126@gmail.com

Received: 18 March 2025 | Revised: 19 May 2025 | Accepted: 31 May 2025

 DOI: https://doi.org/10.48084/etasr.11023

ABSTRACT

This study addresses the substantial threat of ransomware to both home users and enterprises, particularly in corporate settings where shared servers store critical data accessed by users. A novel approach is proposed to prevent crypto-ransomware attacks in File-Sharing Scenarios with Encrypted Traffic (CRP-MTEANN-BWOA-FSET). Ransomware samples are collected from encrypted files shared in the network during infection and from benign activities of office users. Employing Adaptive Self-Guided Filtering (ASGF) for normalization and General Synchroextracting Chirplet Transform (GSCT) for feature extraction, the proposed method extracts critical network traffic features for analysis. Subsequently, MTEANN is utilized to classify ransomware samples as safe or risky, with an emphasis on blocking crypto-ransomware activities in file-sharing scenarios. The proposed method uses the Beluga Whale Optimization Algorithm (BWOA) to optimize the weight parameters in MTEANN, improving the accuracy of ransomware classification. The efficiency of the CRP-MTEANN-BWOA-FSET method was evaluated using accuracy, precision, recall, F-measure, and phi coefficient, and the results showed that it achieved better performance compared to existing methods.

Keywords-adaptive self-guided filtering; beluga whale optimization; file-sharing; multi-task evolutionary artificial neural network; ransomware

I. INTRODUCTION

Crypto-ransomware, malicious software that extorts users by encrypting their files and demanding ransom for decryption, was identified by EUROPOL in 2016 as the most significant malware threat to individuals and businesses [1-2]. Since 2018, crypto-ransomware attacks have targeted various sectors, such as manufacturing, transport, economics, telecommunications, law enforcement, and healthcare, driven by lucrative financial gains for malware developers. In corporate settings, the use of shared network volumes to store valuable company files facilitates backup policies, sharing capabilities, and extended access control [3-5]. However, this architecture also poses risks, as a single infected computer can encrypt accessible files on shared volumes, leading to a compromised environment. A study involving 5,000 IT managers across 26 countries found 65% of ransomware victims experienced data loss in network-shared volumes [6]. Detection tools have been developed to combat the propagation of ransomware, focusing on this specific type of malware. Traditional methods monitor information in network file-access protocol messages, but with the prevalence of public internet cloud services and the growing emphasis on network transaction confidentiality, such protocols are developing into encrypted versions [7-9]. As a result, traffic monitors are unable to attain detailed information about disk-access actions, rendering detection tools that depend on such information less effective. File access operations on valuable files can still be monitored through network traffic

probes, observing traffic exchange among user hosts and file servers. Detection proposals depend on local interception of disk access system calls utilized in file-sharing scenarios, as remote-access protocols provide applications with an interface to shared files utilizing the local file system. It is crucial to distinguish between encryption at the protocol level with encryption at the application level [10, 11]. Even if a ransomware application attempts to overwrite a file with encrypted content, clear-text file-sharing protocol data reveals information about the operation type, file path, and byte position, despite the encrypted content. However, novel file-sharing protocols utilizing encrypted transport lack the required clear-text protocol messages for effective detection [12].

In [13], a novel approach was proposed to detect crypto-ransomware within file-sharing network scenarios characterized by encrypted traffic. This study presented a tool for identifying and obstructing crypto-ransomware activities by analyzing file-sharing traffic. This tool actively monitors communication among clients and file servers, employing ML models to discern patterns indicative of ransomware actions, such as reading and overwriting files, and achieving high accuracy but low precision. In [14], an innovative method was proposed for the early identification of crypto-ransomware through the implementation of a pre-encryption detection algorithm. This algorithm was designed to discover crypto-ransomware at the pre-encryption stage, before any encryption occurs. PEDA incorporates dual levels of detection. The initial level involves comparing signatures with known crypto ransomware utilizing SHA-256, facilitating a swift and accurate assessment of file content to identify potential threats before ransomware activation. The second level employs a learning algorithm that identifies crypto-ransomware depending on pre-encryption API behavior. This dual-level approach enhances the overall efficacy of early detection in combating the ever-evolving landscape of cyber threats, providing a high recall but with a low F1-score. In [15], a novel method was proposed to counter crypto-ransomware on Windows platforms using a honey file-based method by R-Locker, achieving high accuracy and lower precision.

The main contributions of this study are:

- Employs Adaptive Self-Guided Filtering (ASGF) for sample normalization in the preprocessing stage.
- Uses General Synchroextracting Chirplet Transform (GSCT) to extract network traffic features, such as the number of bytes written or read, or control commands, from the preprocessed ransomware samples.
- Employs a Multi-Task Evolutionary Artificial Neural Network (MTEANN) to classify ransomware samples as Safe or Risky, enabling the blocking of crypto-ransomware activities depending on file-sharing traffic analysis.
- Uses the Beluga Whale Optimization Algorithm (BWOA) to enhance the weight parameters of the classifier, ensuring precise classification of ransomware samples.
- Compares the results with traditional ML algorithms.

II. PROPOSED METHOD

This section discusses the proposed method, called Crypto-Ransomware Prevention using Multi-task Evolutionary Artificial Neural Network through Beluga Whale Optimization Algorithm in File Sharing scenario with Encrypted Traffic (CRP-MTEANN-BWOA-FSET). BWOA has been utilized in previous research in diverse contexts [16]. The proposed framework consists of five phases, as shown in Figure 1: data acquisition, preprocessing, feature extraction, classification, and optimization.

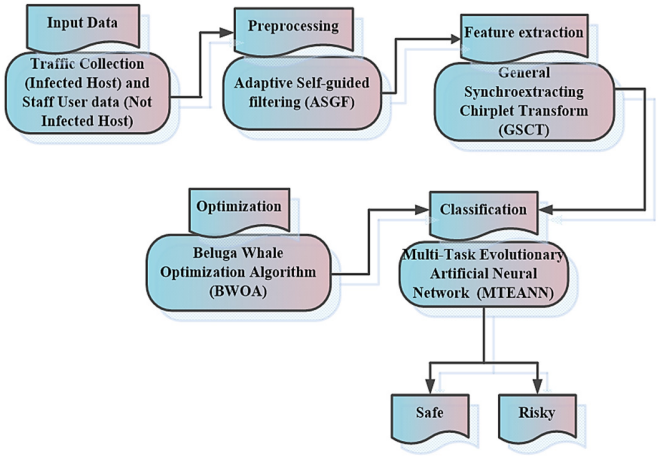

Fig. 1. Diagram of the proposed CRP-MTEANN-BWOA-FSET.

A. Data Acquisition

The study utilizes two sample types: (i) data from traffic captured through encryption network-shared files by crypto-ransomware (referred to as infected), and (ii) data from staff office users who are running benign applications, retrieving shared files (referred to as not infected). These data samples were taken from a publicly available dataset [17]. The data repository contains the results of running more than 70 samples from 31 different ransomware strains. The dataset was divided into 70:30 for training and testing purposes.

B. Preprocessing

The collected samples are preprocessed to normalize them using ASGF [18]. Normalization is used to scale the values of different features within the data. In the context of ASGF, the normalization process is adaptive and self-guided, meaning that the normalization is dynamically adjusted depending on the features themselves. The preprocessing steps are given below.

The dataset values were converted into binary form to obtain a matrix of 16-bit values. Each value of the matrix can be represented as $P(x, y)$ where x and y are the coordinate positions of P in the matrix.

Assume a guided filter as a local linear method among guidance network I and filter output q. The output is expressed as a linear transformation:

$$p_I = b_p I_i + c_p, \quad \forall i \in m_p \tag{1}$$

To establish the linear coefficients of expression (b_p, c_p), a purpose rate in the network m_p is utilized to minimize the difference between the output p_i. The preprocessing step involving ASGF ensures that the ransomware samples are appropriately normalized, and this normalized data is then subjected to feature extraction.

C. Feature Extraction Using General Synchroextracting Chirplet Transform (GSCT)

GSCT [19] is applied to network traffic data to extract features related to dynamic variations in the frequency of bytes read, written, or control commands over time. This allows for a more detailed representation of how these parameters change within the network traffic signals. The integration of GSCT allows for the transformation of complex I/O kinetics time-series into a highly concentrated, two-dimensional Time-Frequency Representation (TFR). TFR enables the subsequent extraction of robust and invariant Haralick features. This synthesis is novel for the detection of ransomware.

In the feature extraction process, various characteristics of the normalized data are analyzed to distinguish between benign user behavior and crypto-ransomware activity depending on network traffic patterns. Network traffic features, such as the number of bytes written, read, or control commands, are extracted using GSCT.

The GSCT algorithm starts by identifying the chirplet that best matches the signal at a certain time point. A network traffic feature extracts the following features: bytes read, bytes written, or control commands. The underlying dynamics of the signal and used for a wide range of applications and time-frequency analysis as:

$$E_{j,t} = \int Q(r)\, \Delta^{y}{}_{j,t}(r-v) ir \quad (2)$$

Here $E_{j,t}$ denote chirplet rate, ir denotes word embeddings of $Q(r)$ acquired by the Q transform, i.e. $(r-v)ir$.

Ξ denotes a complex conjugate of a window function, expressed as:

$$\Xi^{0}_{(j,t)}(r) = o_t(r)\, exp(-c^{\chi_{-e^2}}) \quad (3)$$

Finally, the extracted features are:

- Bytes Client-Server: Represents bytes being written during a write operation. A write operation is identified as a one-packet response to a large (more than one packet) request.
- Bytes Server-Client: Pertains to the bytes being read during a read operation. A read operation is identified as a one-packet request with a large response (more than one packet).
- Control or Short Commands: This encompasses operations such as delete, rename, open, or close a file. Such commands typically fit into single-packet requests from a client to the server or single-packet responses from the server to the client. The number of short commands per second is expressively lower for benign activities, such as directory duplication, compared to crypto-ransomware. Analysis is performed in per-second time intervals, utilizing a temporal window of T seconds to make comprehensive time-samples for the learning process. Each sample represents a time window with $3 \times T$ features (bytes read, written, short commands), serving as input for the classification method.

D. Classification by a Multi-Task Evolutionary Artificial Neural Network

The extracted features are fed into MTEANN [20] for classifying the ransomware samples as Safe or Risky. MTEANN is employed to block crypto-ransomware activity depending on file-sharing traffic analysis. The extracted features (bytes read, bytes written, control commands) are fed into the input layer of the NN. The neural network is trained using a labeled dataset (Safe or Risky). The weights of the network are adjusted through training to minimize the difference between predicted outputs and actual labels.

The formal definition of a multi-task regression paradigm is articulated as follows:

$$F = \left\{ (u_i, v_{i6h}, v_{i12h}, v_{i,24h}. v_{i48h}) \right\};\ i = 1,2,\ldots,N \quad (4)$$

where F represents the dataset utilized, and u_i is the vector containing seven input weighted variables. The variables $\{(u_i, v_{i6h}, v_{i12h}, v_{i,24h}, v_{i48h})\}$ denote the flux of energy at all respective time prediction horizons, while N represents the total number of instances in the dataset.

The input layer of the MTEANN model consists of 3 neurons, one for each input variable, whereas the output layer is composed of four neurons, one for each time prediction horizon (6, 12, 24, and 48).

Regarding the neurons of the hidden layers, three fundamental basis functions are considered. The first of these is the Sigmoidal Unit (SU), recognized as the most commonly employed basis function. Sigmoidal NNs or Multi-Layer Perceptrons (MLPs) are capable of accurately approximating any continuous function. However, it's worth noting that they may encounter challenges such as frequently getting stuck in local minima during the training phase. The SU basis function is expressed as:

$$G_j(u, f_j) = \frac{1}{1+e^{-(f_{j0}+\sum_{i=1}^{d} f_{ji} u_i)}},\ j = 1, \ldots, m \quad (5)$$

where f_{j0} is the bias.

The PU (Product Unit) basis function offers an alternative to SU, introducing multiplicative neurons rather than additive ones. PU-dependent NNs excel in representing robust interactions among input variables, although the training process is further intricate due to the sensitivity of small weight changes, which can lead to significant alterations in the total error. The PU basis function can be expressed as:

$$G_j(u, f_j) = \prod_{i=1}^{d}, j = 1, .., m \quad (6)$$

where $f_j^t = (f_{j1}, f_{j2}, \ldots f_{jd})$, without considering bias f_{j0} in the input.

The Radial Basis Function (RBF) serves as a kernel function, while PU and SU are projection functions. Unlike PU and SU, RBF offers localized behavior for each neuron, as each one is positioned in a distinct area of the search space. This characteristic helps reduce the number of local minima. The output layer is defined as:

$$S_q(u, f, \beta) = \prod_{j=1}^{m} b_j(u, f_j)^{\beta_{qj}}, q = 1, \ldots . Q \quad (7)$$

The output layer predicts the output into two classes, Safe and Risky. Then, BWOA is used to enhance the weight parameters (w_i, w_j) of the MTEANN classifier, enhancing the precision of classifying ransomware samples.

E. Stepwise Procedure for Optimizing MTEANN with BWOA

BWOA [21] is utilized to improve the precision of classifying ransomware samples.

1) Step 1: Initialization

Initialize the population of potential solutions. In this context, this involves randomly making a set of initial weights for the MTEANN classifier as:

$$H_A = \begin{bmatrix} h(a_{1,1}, a_{1,2}, \ldots . , a_{1,e}) \\ h(a_{2,1}, a_{2,2}, \ldots . , a_{2,e}) \\ \ldots \ldots \ldots \\ h(a_{n,1}, a_{n,2}, \ldots . , a_{n,e}) \end{bmatrix} \quad (8)$$

where H_A denotes the objective function of each searching agent in all connections and $h(a_{n,e})$ denotes the quick configuration shifts in every sensor node.

2) Step 2: Random Generation

New candidate solutions are generated in the search space. This step often involves introducing randomness to explore a diverse set of potential solutions. In BWOA, it could involve perturbing the current weights of the MTEANN as:

$$A_j^i - ly_j + (uy_j - ly_j) \times Random \quad (9)$$

where A_j^i denotes the beluga whale and $(uy_j - ly_j)$ denotes the control variables' maximum and lowest limits in the sensor nodes.

3) Step 3: Fitness Function

The fitness function records the passing of sensor nodes, such as the pending energy and the list of adjacent devices on a regular basis as:

$$FitnessFunction = optimize(w_l \; and \; w_j) \quad (10)$$

4) Step 4: Exploration Phase

The search space is explored to detect promising areas that may cover optimal solutions. BWOA, inspired by the natural behavior of beluga whales, includes mechanisms for individuals to explore the solution space efficiently. This exploration phase helps in discovering different areas where the optimal solution might exist.

$$A_i^{T+1} = R_3 A_{Best}^T - R_4 A_i^T + Z_1 . k_e . (A_R^T - A_i^T) \quad (11)$$

where A_i^{T+1} denotes the updated location target nodes of the i^{th} whale in the j^{th} dimension, R_3 and R_4denote random values in the [0, 1] range, T denotes the current iteration in the sensor node, A_i^Trefers to the current location target nodes, k_e denotes the latency, and Z_1 is an adaptive factor used to measure intensity in sensor nodes.

5) Step 5: Exploitation Phase for Optimizing w_i and w_j

The information gained during exploration is exploited to refine the solutions and move towards the optimal one. BWOA employs strategies to focus on regions that show promise and exploit the information gathered. In the context of optimizing the weight parameters of the MTEANN, this involves adjusting the weights to improve classification performance. Then, the BWOA process is utilized to decide the best course of action:

$$A^{T+1} = \begin{cases} A_{best}^T + R_8(A_{best}^T - A_i^T) + \beta(A_{best}^T - A_i^T), & i = 1 \\ A_{best}^T + R_8(A_{i-1}(T) - A_i^T) + \beta(A_{best}^T - A_i^T), & i = 2,3, \ldots , n \end{cases} \quad (12)$$

where $A_{i,j}^{T+1}$ denotes updated location target nodes of the i^{th} whale in the j^{th} dimension, β denotes a weight coefficient of each node, T denotes the current iteration in sensor node passing, A_{Best}^T refers to the best location in the network destination, A_i^T denotes the current location packet's source, and R_8 denotes a random number in the range $[0,1]$.

F. Step 6: Termination

The weight parameter value of the generator (w_i and w_j) from MTEANN is optimized by utilizing BWOA. Step 3 repeats until it obtains its halting criteria. Once the termination condition is met, the optimization process concludes, and the final set of optimized weights is obtained.

III. RESULTS AND DISCUSSION

CRP-MTEANN-BWOA-FSET was implemented in Python and was analyzed using performance metrics such as accuracy, precision, recall, F-measure, and phi coefficient. The proposed technique was compared with other machine learning techniques, such as transfer learning based CRP-TE-FSET, random forest-based CRP-RF-FSET, and SVM-based CRP-SVM-FSET. All these techniques were implemented in a file-sharing environment with encrypted traffic.

Figure 2 shows a comparison of accuracy between the suggested and existing methods. The proposed CRP-MTEANN-BWOA-FSET technique achieved an accuracy that was 10.56%, 18.76%, and 20.67% higher for Safe and 11.46%, 16.58%, and 7.54% higher for Risky compared to CRP-TE-FSET, CRP-RF-FSET, and CRP-SVM-FSET models, respectively.

Figure 3 shows a comparison of precision, where the proposed technique achieved 19.56%, 17.76% and 21.67% higher precision for Safe and 18.46%, 20.58%, and 16.54% higher for Risky compared to CRP-TE-FSET, CRP-RF-FSET, and CRP-SVM-FSET models, respectively.

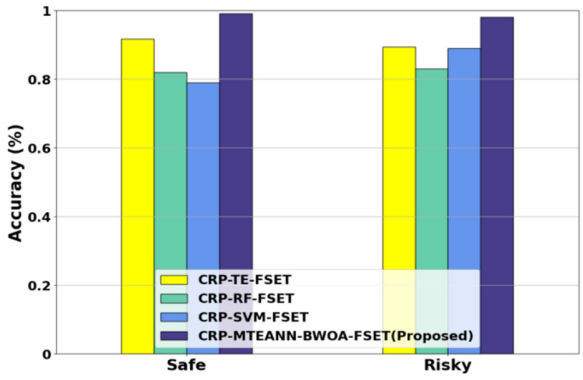

Fig. 2. Accuracy.

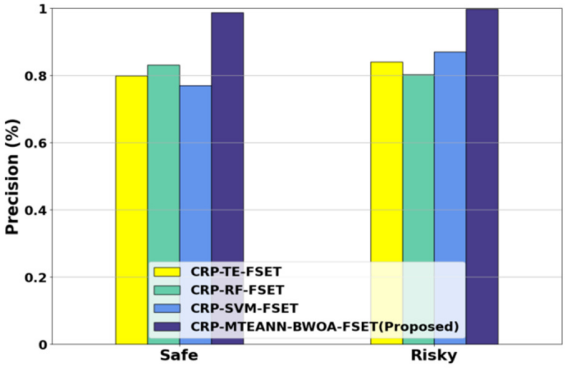

Fig. 3. Precision.

Figure 4 shows a comparison of recall between the suggested and existing methods. The proposed technique achieved a recall of 13.66%, 16.56% and 12.47% higher for Safe and 12.46%, 14.68%, and 11.74% higher for Risky compared to the other models.

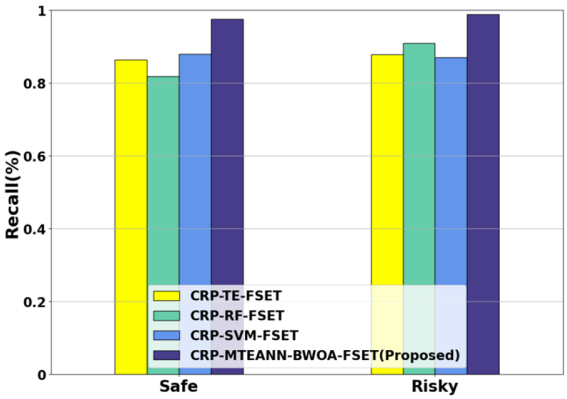

Fig. 4. Recall analysis.

Figure 5 shows a comparison of F1-score, where the proposed CRP-MTEANN-BWOA-FSET technique achieved an F1-score of 16.26%, 19.56% and 12.47% higher for Safe and 17.46%, 16.68%, and 15.74% higher for Risky compared to the existing models.

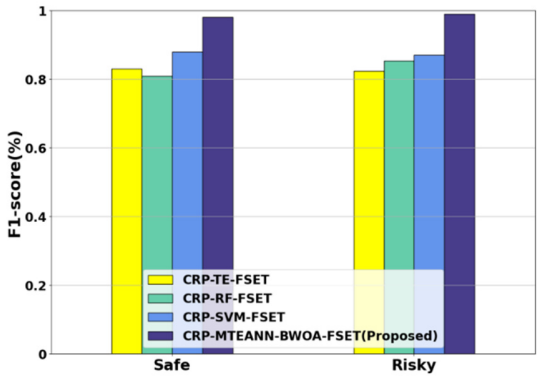

Fig. 5. FI-score analysis.

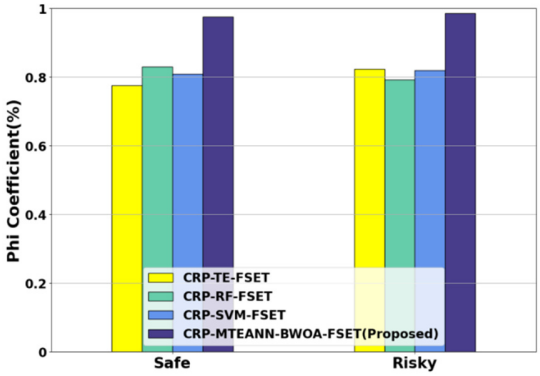

Fig. 6. Phi coefficient analysis.

Figure 6 shows a comparison of the results in terms of the Phi coefficient between the suggested and existing methods. The proposed CRP-MTEANN-BWOA-FSET technique achieved Phi coefficients that were 20.26%, 16.56% and 18.47% higher for Safe and 18.46%, 20.68%, and 18.74% higher for Risky compared to the existing models.

IV. CONCLUSION

The proposed optimized MTEANN for CRP-MTEANN-BWOA-FSET presents a comprehensive approach to enhance security in corporate environments. By leveraging advanced techniques such as ASGF for preprocessing and GSCT for feature extraction, the model demonstrates effective classification of crypto-ransomware samples. The incorporation of BWOA for optimizing MTEANN's weight parameters further enhances accuracy. Implemented in Python, the proposed method exhibited promising results in terms of performance metrics, affirming its potential as an efficient tool for combating crypto-ransomware in file-sharing scenarios through encrypted traffic. The proposed system was developed for the Windows operating system, without considering mobile operating systems, assuming that file-sharing networks are unlikely to run on Android or other mobile operating systems. The proposed system describes a static solution to detect crypto-ransomware that may not be valid for new strains. In the future, better adaptive training models will be investigated, incorporating samples and observing the improvement or degradation of the model.

REFERENCES

[1] B. V. Reddy, G. J. Krishna, V. Ravi, and D. Dasgupta, "Machine Learning and Feature Selection Based Ransomware Detection Using Hexacodes," in *Evolution in Computational Intelligence*, 2021, pp. 583–597, https://doi.org/10.1007/978-981-15-5788-0_56.

[2] 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, https://doi.org/10.1109/access.2021.3114148.

[3] I. Bello *et al.*, "Detecting ransomware attacks using intelligent algorithms: recent development and next direction from deep learning and big data perspectives," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 9, pp. 8699–8717, Sep. 2021, https://doi.org/10.1007/s12652-020-02630-7.

[4] F. Faghihi and M. Zulkernine, "RansomCare: Data-centric detection and mitigation against smartphone crypto-ransomware," *Computer Networks*, vol. 191, May 2021, Art. no. 108011, https://doi.org/10.1016/j.comnet.2021.108011.

[5] 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.

[6] S. Sharma, C. R. Krishna, and R. Kumar, "RansomDroid: Forensic analysis and detection of Android Ransomware using unsupervised machine learning technique," *Forensic Science International: Digital Investigation*, vol. 37, Jun. 2021, Art. no. 301168, https://doi.org/10.1016/j.fsidi.2021.301168.

[7] A. Mantri, N. Singh, K. Kumar, and S. Dahiya, "Pre-Encryption and Identification (PEI): An Anti-crypto Ransomware Technique," *IETE Journal of Research*, vol. 69, no. 11, pp. 8058–8066, Nov. 2023, https://doi.org/10.1080/03772063.2022.2048706.

[8] D. Smith, S. Khorsandroo, and K. Roy, "Machine Learning Algorithms and Frameworks in Ransomware Detection," *IEEE Access*, vol. 10, pp. 117597–117610, 2022, https://doi.org/10.1109/access.2022.3218779.

[9] V. Thangapandian, "Machine Learning in Automated Detection of Ransomware: Scope, Benefits and Challenges," in *Illumination of Artificial Intelligence in Cybersecurity and Forensics*, S. Misra and C. Arumugam, Eds. Springer International Publishing, 2022, pp. 345–372.

[10] J. Singh, K. Sharma, M. Wazid, and A. K. Das, "SINN-RD: Spline interpolation-envisioned neural network-based ransomware detection scheme," *Computers and Electrical Engineering*, vol. 106, Mar. 2023, Art. no. 108601, https://doi.org/10.1016/j.compeleceng.2023.108601.

[11] F. Manavi and A. Hamzeh, "A novel approach for ransomware detection based on PE header using graph embedding," *Journal of Computer Virology and Hacking Techniques*, vol. 18, no. 4, pp. 285–296, Dec. 2022, https://doi.org/10.1007/s11416-021-00414-x.

[12] R. O. Ogundokun, J. B. Awotunde, S. Misra, O. C. Abikoye, and O. Folarin, "Application of Machine Learning for Ransomware Detection in IoT Devices," in *Artificial Intelligence for Cyber Security: Methods, Issues and Possible Horizons or Opportunities*, S. Misra and A. Kumar Tyagi, Eds. Springer International Publishing, 2021, pp. 393–420.

[13] E. Berrueta, D. Morato, E. Magaña, and M. Izal, "Crypto-ransomware detection using machine learning models in file-sharing network scenarios with encrypted traffic," *Expert Systems with Applications*, vol. 209, Dec. 2022, Art. no. 118299, https://doi.org/10.1016/j.eswa.2022.118299.

[14] 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.

[15] J. A. Gómez-Hernández, R. Sánchez-Fernández, and P. García-Teodoro, "Inhibiting crypto-ransomware on windows platforms through a honeyfile-based approach with R-Locker," *IET Information Security*, vol. 16, no. 1, pp. 64–74, 2022, https://doi.org/10.1049/ise2.12042.

[16] M. Waty, H. Sulistio, and A. Prihatiningsih, "The Impact of Change Orders on the Waste Materials of Large-Scale Projects," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 18363–18370, Dec. 2024, https://doi.org/10.48084/etasr.8910.

[17] E. Berrueta, D. Morató, E. Magaña, and M. Izal, "Open Repository for the Evaluation of Ransomware Detection Tools." IEEE DataPort, Feb. 27, 2020, https://doi.org/10.21227/5AXV-3829.

[18] S. Zhu and Z. Yu, "Self-guided filter for image denoising," *IET Image Processing*, vol. 14, no. 11, pp. 2561–2566, 2020, https://doi.org/10.1049/iet-ipr.2019.1471.

[19] Z. Meng, M. Lv, Z. Liu, and F. Fan, "General synchroextracting chirplet transform: Application to the rotor rub-impact fault diagnosis," *Measurement*, vol. 169, Feb. 2021, Art. no. 108523, https://doi.org/10.1016/j.measurement.2020.108523.

[20] D. Guijo-Rubio, A. M. Gómez-Orellana, P. A. Gutiérrez, and C. Hervás-Martínez, "Short- and long-term energy flux prediction using Multi-Task Evolutionary Artificial Neural Networks," *Ocean Engineering*, vol. 216, Nov. 2020, Art. no. 108089, https://doi.org/10.1016/j.oceaneng.2020.108089.

[21] C. Zhong, G. Li, and Z. Meng, "Beluga whale optimization: A novel nature-inspired metaheuristic algorithm," *Knowledge-Based Systems*, vol. 251, Sep. 2022, Art. no. 109215, https://doi.org/10.1016/j.knosys.2022.109215.