International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

AI-Powered Encryption: Innovative Approach for Malware Intrusions in File-Sharing Networks

Dr. Intekab Alam[1], Adlin Jebakumari S[2], Sunil Kumar Jakhar[3], Dr. Karishma Desai[4], Madhur Grover[4]

Submitted: 05/02/2024 **Revised**: 13/03/2024 **Accepted**: 19/03/2024

Abstract: Malware intrusions in file-sharing networks are a prevalent problem, compromising the integrity and security of encrypted data. These disguised attacks took the utilization of flaws in network designs, spreading intentionally and corrupting data shared by users. The dynamic and distributed organization of file-sharing networks complicates the detection and prevention of such attacks. In this study, we developed an optimized encryption model named Adaptive Emperor Penguin tuned Bayesian Belief Networks (AEP-BBN) for enhancing the prediction of malware activities in file-sharing networks. Initially, we gathered a dataset that includes infected shared files from the organizations to train our proposed prediction model. A robust Scaling (RS) algorithm is employed to pre-process the gathered raw data, to improve the quality of the data. We extracted significant features from the pre-processed data using Recursive Feature Elimination (RFE). Adaptive Emperor Penguin Optimization (AEPO) is used to enhance the primary features of the suggested BBN architecture. The recommended approach has been implemented in Python software. The result assessment phase is performed with numerous metrics such as recall, precision, f1 score and accuracy to evaluate the suggested AEP-BBN approach with other conventional approaches. The outcomes of the experiments demonstrate that the proposed AEP-BBN approach performed better than other existing approaches for enhancing the prediction of malware activities in file-sharing networks.

Keywords: *Malware Intrusions, Encryption, Encryption, Adaptive Emperor Penguin tuned Bayesian Belief Networks (AEP-BBN), Artificial intelligence (AI).*

1. Introduction

Malware is a term used to describe a broad range of harmful programming codes, scripts, invasive software or active content that can be used to destroy computer systems, mobile applications and online applications. Examples of these include computer viruses, spyware, ransomware, worms, dialers, root kits, Trojan horses, adware, key loggers and malicious Browser Helper Objects (BHOs) [1]. The words malicious and software are combined to form the term malware. Malicious software is defined as software that does a common task without authorization. There are two components to malware: Payload and exploits (carrier and activity) [2]. Malware appears as malicious software or legitimate software or shows up as real software when it executed, it does a hidden operation [3]. Malware is harmful software that tries to control a user's device, steal their personal information, and interfere with the operating system's ability to function [4]. Malware Intrusions are becoming faster and provide a major risk to computer systems, especially mobile, robotic, and cloud-based ones [5]. Malware has a history of targeting various Operating Systems (OS) components from when it was developed. Numerous techniques have been developed to detect malware and enhance OS resilience to minimize the risk of malware attacks and protect against Malware Intrusions [6]. MI involves malicious software or malware entering a victim's device, allowing unauthorized operations such as data encryption, screen locking, and data exfiltration [7].

The objective of this study aims to identify Malware incidents using feature extraction from encrypted network data. Previous detection techniques were not based on encrypted file-sharing network, and an analytic tool for intricate architectures is needed. An analytic tool that can find patterns in intricate architectures is necessary due to the volume of features this traffic offers. In this study, Bayesian Belief Network model is trained and evaluated. In a network investigation, we assess models that could operate quickly and effectively.

The following portions of this study are organized in the following order: Section 2 contains the related works and Section 3 contains a comprehensive analysis of the situation and approach, including datasets used for model

[1]*Assistant Professor, Maharishi School of Engineering and Technolgy, Maharishi University of Information Technology, Lucknow, India - 226036, Email Id- intekhab@muit.in, Orcid Id- 0000-0001-5473-2408*
[2]*Assistant Professor, Department of Computer Science and Information Technology, Jain (Deemed to be University), Bangalore, Karnataka, India, Email Id- j.adlin@jainuniversity.ac.in , Orcid Id- 0000-0003-0392-4367*
[3]*Assistant Professor, Department of Mechanical Engineering, Vivekananda Global University, Jaipur jakhar.sunil@vitj.ac.in, Orcid Id- 0009-0000-4338-3070*
[4]*Associate Professor, Department of ISME, ATLAS SkillTech University, Mumbai, Maharashtra, India, Email Id- karishma.desai@atlasuniversity.edu.in, Orcid Id- 0009-0009-5171-9588*
[5]*Centre of Research Impact and Outcome, Chitkara University, Rajpura-140417, Punjab, India, Email ID- madhur.grover.orp@chitkara.edu.in, Orcid Id-https://orcid.org/0009-0008-3520-4667*

testing, validation, and training. Section 4 presents the results of tests, validating, and training machine learning models, highlighting the chosen model and design trade-offs and comparing outcomes with other literature tools, and Section 5 presents the conclusions.

2. Related works

The study [8] utilized neural network methods to analyze malicious Windows portable execution files, using EMBER as a benchmark dataset. Techniques like Convolutional Neural Networks and Feed Forward Neural Networks were used to identify malware in 1.1 million binary files. The study [9] uses data from the Drebin project to identify key malware elements using graph-based machine learning techniques. Malware detection software uses Random Forest, K-Nearest Neighbour, Decision Tree, and Logistic Regression algorithms for training and classification. The study [10] demonstrated that ensemble algorithms perform better at predicting malware than conventional machine learning methods. Light GBM reduced the feature count from 215 to 100, resulting in a good accuracy rate. The study [11] conducted a thorough comparative analysis of wired networks' P2P file sharing methods, examining their effectiveness under different conditions and examining advanced P2P networks, including wired and wireless networks, to understand their effectiveness. The study [12] aimed to improve proactive detection, analysis and defense against internal and external threats in financial institutions, an AI-based system leveraging machine learning for cyber security threat identification was developed. The study [13] integrated AI-based solutions into digital health communication to reinforce security measures, improve threat detection, and guarantee quick reaction. Sensitive patient data was protected both during transmission and storage by using algorithms for real-time analysis. The study [14] aimed to develop an AI-based ransomware detection framework using static and dynamic malware analysis methodologies, focusing on email scams and enhancing accuracy with support vector machine and Adaboost with J48 algorithms. The study [15] aimed to explore the transformative potential of artificial intelligence in various applications such as cyber security, wireless communication, and Internet of Things networks. It will analyze AI's widespread adoption, impact on automation, smart communication and cyber security measures across various industries using various methods. The study [16] developed a Deep Learning-based Intrusion Detection System to enhance security in the chemical industry's IT and OT networks. The system successfully detected and prevented various attacks with 87.19% accuracy. The study [17] developed a Linux server safeguard using EBPF and machine learning, demonstrating high detection accuracy in real-world settings and highlighting the potential for malware spread through email scams.

3. Methods

The experimental technique employed in the study is explained in the procedure. This section offers a strategy for implementing the recommended artificial intelligence model into practice, ensuring openness and repeatability in the study's approach to malware intrusion inputs. Fig.1 depicts the Recommended Approach's organizational structure.

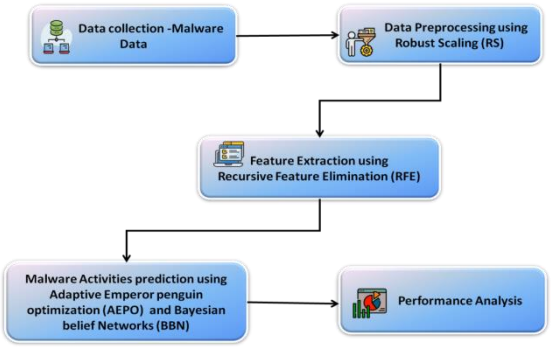

Fig1. Structure of the Recommended Approach

3.1 Dataset

The Microsoft Malware Prediction dataset was utilized for predicting the probability that a Windows PC that are infected with various types of malware. The dataset consisting of 9 million rows of training and 8 million rows of testing has 82 characteristics [17], primarily categorical. To protect data privacy, 23 variables have been numerically encoded. Table 1 depicts the dataset description.

Table 1. Dataset description

Features	Special Qualities	Missing values in Percentage	Value percentage in the largest category	Category
Firewall	2	1.023933	96.856251	Float16
S Mode	2	6.027686	93.928812	Float16
Pua Mode	2	99.974119	99.974119	Category
Default Browsers Identifier	1730	95.141637	95.141637	Float16

Smart Screen	21	35.610795	48.379658	Category
Organization Identifier	49	30.841487	47.037662	Float16

3.2 Data Preprocessing Using Robust Scaling (RS)

A technique known as robust scaling modifies the characteristics according to the median and inter-quartile range, therefore strengthening them against outliers. This approach guarantees that the outliers do not dominate the feature values, which makes it helpful when the data contains outliers. When the data has a high number of outliers, robust scaling can yield superior results, as it is resilient to outliers. It could be more difficult to implement and could lose the data's original structure. Equation (1) shows a robust scaling function.

$$X_{scaled} = \frac{X - X_{median}}{IQR} \quad (1)$$

Where X_{scaled} represents the features resized form. X Represents the initial feature values, X_{median} represents the feature's median, IQR is the feature's inter-quartile range, which is determined as $Q_3(X) - Q_1(X)$ where $Q_1(X)$ represents the first quartile and $Q3(X)$ represents the third quartile.

3.3 Feature Extraction Using Recursive Feature Elimination (RFE)

Better criteria for classifying characteristics cannot always translate to better criteria for classifying subsets of characteristics. $DK(j)$ Or $\left(w_j\right)^2$ a criterion calculates the impact of deleting an item using the objective function one at a time. When numerous functions are eliminated at the same time and the model becomes extremely non-optimal, which is required to acquire a subset of miniature features. The following recurrent process, which we refer to as Recursive Feature Elimination, can be used to fix this issue:

1. Optimize the weights w_j in relation to K to train the classifier.
2. Calculate the ranking criterion for each characteristic ($DK(j)$or $(w_j)^2$).
3. Take out the feature that has the lowest ranking criterion.

This iterative process demonstrates the removal of undeveloped features. Several functions could be excluded for computational reasons, as this could cause a drop in classification performance. In these situations, his approach results in a classification of a subset of traits rather than a classification of all traits. The nested Subset of Features is given by $F_1 C F_2 C \ldots C F..$

3.4 Malware Activities Prediction Using Adaptive Emperor Penguin tuned Bayesian Belief Networks (AEP-BBN)

3.4.1 Adaptive Emperor Penguin Optimization (AEPO)

Three techniques are introduced in AEPO to address the disadvantages of the classic EPO: Gaussian Mutation (GM), Opposition-Based Learning (OBL) and Levy Flight (LF). This allows EPO to maintain an appropriate balance between exploration and extraction. The AEPO method can optimize multi-threshold problems by enhancing penguins' independent hop capacity and strengthening their bonds to address the local optimum problem. The following Equation (2) represents the modified mathematical model:

$$P_{ep}(y+1) = P(y) - B \times D_{ep} \oplus G(\alpha) \quad (2)$$

Where P_{ep} indicates Emperor Penguin's position vector and the D_{ep} is the penguins' separation from one another. The Levy flight enhances individual penguins' update position formula by enhancing their leap capacity and broadening the swarm's search area, allowing the perfect penguin to achieve perfect answer quickly. The formulation is shown in the following Equation (3):

$$P_{eq}^{levy}(y+1) = P(y) \times Levy - B \times D_{ep} \oplus G(\alpha) \quad (3)$$

The updated location i^{th} solution is represented byP_{eq}^{levy}. In the end, the best penguin could appear outside of the search region due to the inclusion of Gaussian mutation and Levy flight. The OBL enhances the AEPO algorithm by limiting its leap capacity, thereby preventing the swarm from entering an endless loop and preventing updates to the optimal solution. This increased search domain exploration enhances the swarm's diversity, as per the following Equation (4).

$$P_{ep} = P_{max} + P_{min} - P_{best} + r\left(P_{best} - P_{ep}\right) \quad (4)$$

Wherein P_{ep} is the i^{th} opposing penguin's location within the search domain. The i^{th} variable's lower and upper bounds are denoted by $P_{max}$and P_{min}. The best penguin's location is denoted by P_{best}, while the random vector r has elements between 0 and 1. Additionally, P_{ep} represents the i^{th} penguin in the population's position vector. The penguin's optimal approach is modified,

enhancing results and convergence speed and enabling quicker adjustment and accurate identification of ideal values. Algorithm 1 provides the structure of AEPO.

Algorithm 1: AEPO

Being

$Initialize\ the\ emperor\ penguins\ population\ O_{ep}(w), w = 1,2,\dots,m$:
$Initialize\ the\ parameters$;
$\boldsymbol{Fitness} =\ the\ best\ search\ agent$;
$\boldsymbol{While}\ (1 < Max\ number\ of\ iterations)$
$\boldsymbol{For}\ each\ search\ agent$
$Update\ the\ position\ of\ the\ current\ search\ agent\ by\ the\ Eq.9$;
End for
$Update\ Fitness\ if\ there\ is\ a\ better\ solution$;
$J = J + 1$
End while
Return fitness
End

3.4.2 Bayesian Belief Networks (BBN)

A graphical link between the causative factors is called a Bayesian belief network (BBN) with a solid mathematical foundation in Bayesian probability and the benefits of an easy-to-understand visual representation, BBNs facilitate reasoning in the face of uncertainty. An acyclic-directed graph is called a BBN graph. Nodes are illogical variables that stand for unclear occurrences. The arcs show a causal link between variables, with probability tables linked to each node, conditional probability tables for child nodes and marginal probability for root nodes. As stated in Equation (5), the complete joint probability, calculated as the conditional probability product for each combination of node and parent is revised when a BBN network event is seen with security.

$$PX1,2,\dots\dots\dots Xn=Xj|ParentsXj \quad (5)$$

The following Equation (6) is the combined probability of every node:

$$P\ (D,,Freq,TPF) = P\ (D)*P\ (N|D)*P\ (Freq\ |D)*P\ (TPF|\ Freq,N) \quad (6)$$

Assume that there are two states true and false for each of the four nodes. The following Equation (7) can be used to obtain a few sample conditional probabilities:

$$(N=\ True\ |\ TPF=\ True)=(P(N=True\ ,TPF\ =True))P(TPF\ =True) \quad (7)$$

3.4.3 Adaptive Emperor Penguin tuned Bayesian Belief Networks (AEP-BBN)

Using a model known as Adaptive Emperor Penguin adjusted Bayesian Belief Networks (AEP-BBN), this concept predicts malware activity. It combines the features of Emperor Penguins and modifies their actions to improve Bayesian Belief Network performance. By utilizing Emperor Penguins' special characteristics throughout the tuning phase, this innovative method seeks to increase malware prediction's precision and flexibility. To predict malware activity, the method probably combines sophisticated artificial intelligence algorithms with adaptive methodologies. To enable the real-time optimization which is based on the shifting of malware activities, the model parameter has dynamically and adaptively adjusted with the help utilization of AEP. By this adaptive feature, which allows the system to acquire knowledge from and modify its predictions is the capacity of the AEP-BBN. This AEP-BBN technique aimed to provide an efficient and reliable method used to predict and reduce malware activities with the integration of the adaptability and durability derived from Emperor Penguin tuning into Bayesian Belief Networks.

4. Result and Discussion

We use the Google Tensorflow-Keras package, which is available as an open-source download, together with Python 3.8 on Anaconda for Training. The laptop is powered by a powerful 11th Gen Intel® CoreTM i7-1165G7 CPU @ 2.80GHz and has improved graphics due to Intel® Iris® Xe. It can multitask with ease because of its enormous 512 GB of disc space and 16 GB of RAM. For the fault recognition of wireless sensor networks, we have taken the existing methods that contain "Support Vector Machine (SVM) [18], K-Nearest Neighbor (KNN) [18], and Iterative Dichotomiser 3 (ID3) [18]". For comparative analysis in this study, we use metrics such as F-measure, accuracy, precision and recall shown in Table 2.

Table 2. Outcomes of existing and proposed methodologies

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SVM [18]	92.77	92.68	92.20	92.44
KNN [18]	92.93	93.87	91.47	92.65

ID3 [18]	92.24	90.49	90.49	90.49
AEP-BBN [Proposed]	**94.75**	**95.54**	**94.36**	**93.97**

4.1 Accuracy

Accuracy is recognized to be advantageous in courses that are evenly distributed and it cannot be suitable in classes that are not balanced. It is utilized in image segmentation for the prediction of malware intrusion. Equation (8) is used to determine the ratio of accurate prediction to all predictions.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \qquad (8)$$

Fig.2 provides the Comparison of accuracy for proposed and existing methods. The study evaluated various classification techniques, with the proposed AEP-BBN method achieves the highest accuracy of 94.75%, where the existing methods such as SVM [18] achieved 92.77%, KNN [18] with 92.93% and ID3 [18] with 92.24%.

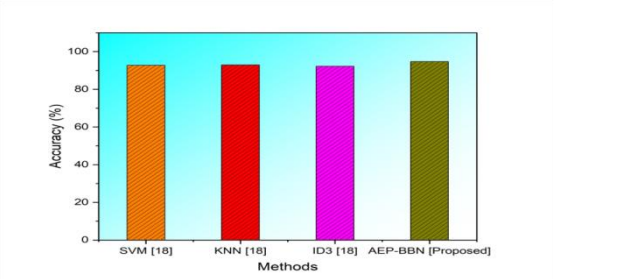

Fig 2. Comparison of accuracy for proposed and existing methods

4.2 Precision

The rate at which the projected positive situations turn out to be positive is determined by Equation (9). It calculates the difference between the total amount of positives predicted by the amount of True Positives (TP) and the model.

$$Precision = \frac{TP}{TP+Fp} \qquad (9)$$

Fig.3 provides the comparison of accuracy for proposed and existing methods. The proposed AEP-BBN method outperforms other methods in precision, with a significant improvement of 95.54% while compared to the existing methods. The existing methods such as SVM, KNN and ID3 achieved the Precision rate of 92.68%, 93.87% and 90.49%, respectively.

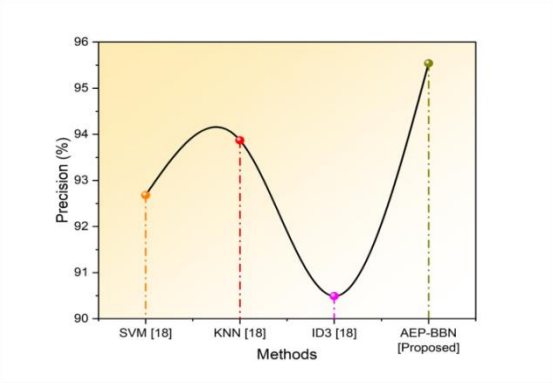

Fig 3. Comparison of Precision for proposed and existing methods

4.3 Recall

Recall, often referred to as sensitivity, quantifies the degree of accuracy between the positive predictions and the actual data. Using the following Equation (10), we can determine the recall for malware intrusion.

$$Recall = \frac{TP}{TP+FN} \qquad (10)$$

Fig.4 provides the comparison of accuracy for proposed and existing methods. The AEP-BBN method achieves 94.36% of recall rate and other previous methods such as SVM, KNN and ID3 attained the percentages of 92.20%, 91.47% and 90.49%, respectively.

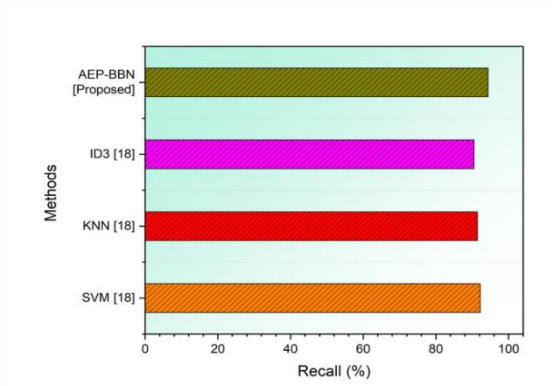

Fig 4. Comparison of Recall for proposed and existing methods

4.4 F-Measure

A popular statistic for assessing the effectiveness of classification models, such as those used for the prediction of malware activities, is the F-Measure. The F1 score can be computed with the following Equation (11):

$$F1\ Score = 2 * \frac{Precision*Recall}{Precision+Recall} \qquad (11)$$

Fig.5 provides the Comparison of accuracy for proposed and existing methods. The F-Measure percentages show the effectiveness of various strategies compared to the

existing method such as SVM [18] achieved 92.44%, KNN [18] with 92.65% and ID3 [18] with 90.49%. The suggested AEP-BBN technique performs significantly better with a better F-Measure (93.87%).

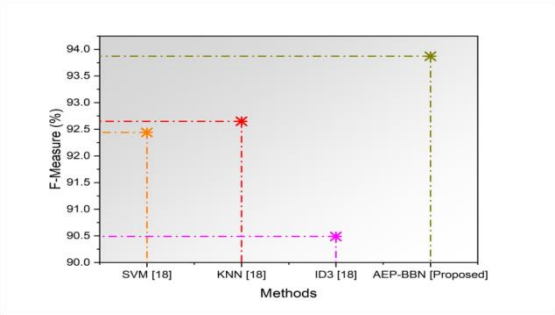

Fig 5. Comparison of F-Measure for proposed and existing methods

5. Conclusion

The study presents the Adaptive Emperor Penguin tuned Bayesian Belief Networks (AEP-BBN) malware prediction model, which effectively addresses malware infiltration in file-sharing networks. The model, which combines AEPO for feature augmentation and Robust Scaling for data preprocessing, outperforms other approaches in malware activity prediction. The comprehensive evaluation, considering various parameters such as Accuracy (94.75%), Recall (94.36%), Precision (95.54%), and F-measure (93.87%) demonstrates that the suggested AEP-BBN technique is effective in enhancing the security of file-sharing networks against unseen assaults. In large-scale file-sharing networks with substantial datasets, the AEP-BBN approach could face issues with scalability and computing efficiency. The goal of future research is to improve the model's real-time malware identification capabilities and its capacity to adjust to new threats.

References

[1] Faruk, M. J. H., Shahriar, H., Valero, M., Barsha, F. L., Sobhan, S., Khan, M. A., ... & Wu, F. (2021, December). Malware detection and prevention using artificial intelligence techniques. In 2021 IEEE International Conference on Big Data (Big Data) (pp. 5369-5377). IEEE.

[2] Yadav, C. S., Singh, J., Yadav, A., Pattanayak, H. S., Kumar, R., Khan, A. A., ... & Alharby, S. (2022). Malware analysis in iot & android systems with defensive mechanism. Electronics, 11(15), 2354. 10.3390/electronics11152354.

[3] Chakraborty, A., Biswas, A., & Khan, A. K. (2023). Artificial intelligence for cybersecurity: Threats, attacks and mitigation. In Artificial Intelligence for Societal Issues (pp. 3-25). Cham: Springer International Publishing.

[4] Almomani, I., Alkhayer, A., & El-Shafai, W. (2022). An automated vision-based deep learning model for efficient detection of android malware attacks. IEEE Access, 10, 2700-2720.

[5] Melvin, A. A. R., Kathrine, G. J. W., Ilango, S. S., Vimal, S., Rho, S., Xiong, N. N., & Nam, Y. (2022). Dynamic malware attack dataset leveraging virtual machine monitor audit data for the detection of intrusions in cloud. Transactions on Emerging Telecommunications Technologies, 33(4), e4287. 10.1002/ett.4287

[6] McIntosh, T., Watters, P., Kayes, A. S. M., Ng, A., & Chen, Y. P. P. (2021). Enforcing situation-aware access control to build malware-resilient file systems. Future Generation Computer Systems, 115, 568-582. 10.1016/j.future.2020.09.035

[7] Al-Hawawreh, M., Alazab, M., Ferrag, M. A., & Hossain, M. S. (2023). Securing the Industrial Internet of Things against ransomware attacks: A comprehensive analysis of the emerging threat landscape and detection mechanisms. Journal of Network and Computer Applications, 103809.

[8] Pramanik, S., & Teja, H. (2019, January). EMBER-Analysis of Malware Dataset Using Convolutional Neural Networks. In 2019 Third International Conference on Inventive Systems and Control (ICISC) (pp. 286-291). IEEE. 10.1109/ICISC44355.2019.9036424

[9] Karrar, A. E. (2022). Adopting Graph-Based Machine Learning Algorithms to Classify Android Malware. IJCSNS International Journal of Computer Science and Network Security, 22(9), 840.10.22937/IJCSNS.2022.22.9.109

[10] Al Sarah, N., Rifat, F. Y., Hossain, M. S., & Narman, H. S. (2021). An efficient android malware prediction using Ensemble machine learning algorithms. Procedia Computer Science, 191, 184-191. 10.1016/j.procs.2021.07.023

[11] Ashraf, F., Naseer, A., & Iqbal, S. (2019, April). Comparative analysis of unstructured P2P file sharing networks. In Proceedings of the 2019 3rd International Conference on Information System and Data Mining (pp. 148-153). 10.1145/3325917.3325952

[12] Kumar, D., & Kumar, K. P. (2023, March). Artificial Intelligence based Cyber Security Threats Identification in Financial Institutions Using Machine Learning Approach. In 2023 2nd International Conference for Innovation in Technology (INOCON) (pp. 1-6). IEEE.10.1109/INOCON57975.2023.10100967

[13] Tyagi, A. K., Hemamalini, V., & Soni, G. (2023). Digital Health Communication With Artificial Intelligence-Based Cyber Security. In AI-Based

Digital Health Communication for Securing Assistive Systems (pp. 178-213). IGI Global.10.4018/978-1-6684-8938-3.ch009

[14] Poudyal, S., & Dasgupta, D. (2020, December). AI-powered ransomware detection framework. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1154-1161). IEEE.

[15] Haider, U., Nawaal, B., Khan, I. U., & El Hajjami, S. AI-Based Secure Wireless Communication Technologies and Cyber Threats for IoT Networks. In Cyber Security for Next-Generation Computing Technologies (pp. 70-83). CRC Press.

[16] Lofù, D., Pazienza, A., Abbatecola, A., Lella, E., Macchiarulo, N., & Noviello, P. (2023, June). Watching against the Unseen: AI-powered Approach to Detect Attacks on Critical Infrastructure. In 2023 8th International Conference on Smart and Sustainable Technologies (SpliTech) (pp. 1-6). IEEE.

[17] bin Asad, A., Mansur, R., Zawad, S., Evan, N., & Hossain, M. I. (2020, June). Analysis of malware prediction based on infection rate using machine learning techniques. In 2020 IEEE region 10 symposium (TENSYMP) (pp. 706-709). IEEE.

[18] Dehkordy, D. T., & Rasoolzadegan, A. (2021). A new machine learning-based method for android malware detection on imbalanced dataset. Multimedia Tools and Applications, 80, 24533-24554. 10.1007/s11042-021-10647-z