[1]Mumtaz Ahmed

Smart End-Point System for Ensuring Efficient Monitoring and Management of Potential Threats in Networks

Journal of Electrical Systems

Abstract: Ransomware attacks involve the malicious encryption of crucial files and subsequent ransom demands for decryption keys, have become a pervasive and destructive threat. These attacks target individuals, organizations, and government bodies, leading to significant financial losses and data breaches. This dissertation focuses on developing a comprehensive system to prevent ransomware attacks, thereby enhancing data security and resilience.

The core objective of this research is to create an advanced software solution that effectively restricts illegal activities, specifically ransomware attacks, within society, institutions, and companies. The software comprises two integral components: an administrator interface and a client-side application. The administrator interface provides a centralized dashboard that allows users to control and customize security parameters, ensuring efficient monitoring and management of potential threats.

A key feature of the proposed solution is the integration of a deep learning model designed to detect and neutralize ransomware activities. The model demonstrated exceptional performance with an accuracy rate of 94.3%, along with high precision and recall metrics, highlighting its robustness in identifying and mitigating ransomware threats. This study involves a thorough analysis of current encryption methods, data protection strategies, and existing cybersecurity measures to develop a resilient framework against ransomware.

Additionally, the research explores innovative approaches, including real-time anomaly detection systems, advanced encryption protocols, and robust data recovery mechanisms, to enhance the overall security of data stored in cloud environments. The proposed system will be validated through simulations and real-world case studies, offering practical insights and actionable recommendations for improving cloud data security.

Keywords: Ransomware attack detection; transfer learning; deep learning ensemble models; cybersecurity.

I.INTRODUCTION

Most work in malware detection, particularly ransomware detection, has traditionally relied on signature-based methods, but recent approaches have explored alternative techniques. Authors in [1] propose an authentication-based access control mechanism called AntiBiotics, which includes three key components. The first component, the Policy Enforcement Driver, serves as an initial barrier that logs and pauses any file modification attempts, such as renames or deletions, requiring users to pass a challenge like CAPTCHA or biometric authentication to proceed. The second component, the Policy Specification Interface, is a GUI program that allows administrators to configure system policies. The third component, the Challenge Response Generator, manages the generation of challenges, controlling factors such as the timeout rate and implementing mechanisms to prevent excessive challenge generation.

The rise of hostile threats against computer networks and digital services poses a significant risk to network infrastructure. To enable linking between websites, the domain name system (DNS) is essential. However, detecting DNS intrusions requires overcoming the challenges of hidden tunnels and conventional detection avoidance. This research paper presents an intrusion detection model utilizing statistical analysis and Bi-directional Recurrent Neural Network (BRNN) methods to identify malicious DNS over HTTPS (DoH) requests through covert channels. The model demonstrates 90% accuracy in detecting malicious DoH searches using data from the Canadian Institute for Cybersecurity's CIRA CIC-DoHBrw-2020 dataset. Additionally, the proposed model outperforms competing methods by using fewer features and achieving higher throughput during both training and testing.

In a comprehensive review of research literature [15], techniques employing machine learning and deep learning for ransomware detection were examined. The devastating impact of ransomware, its persistence once infections occur, and the critical urgency for early detection were primary motivators for this investigation. The role of machine learning in combating ransomware underscores the need to assess current defences and explore avenues for improvement. The proliferation of new ransomware families and strains in cyberspace underscores the severity of the situation. The intricate encryption methods employed by ransomware complicate infection removal.

[1] *Mumtaz Ahmed: Department of Computer Engineering, Jamia Millia Islamia, mahmed1@jmi.ac.in (Corresponding Author)

This state-of-the-art review [7] focuses on recent advancements in machine learning and deep learning for detecting ransomware. The investigation was driven by the urgent need to protect computer systems from ransomware attacks. Machine learning and deep learning techniques have proven effective in identifying zero-day attacks and constructing predictive models based on ransomware behaviour, leading to their widespread adoption.

As ransomware evolves, it continuously exploits new vulnerabilities discovered in computer systems. Both researchers and practitioners rely heavily on machine learning techniques, particularly transfer learning, for the detection and mitigation of ransomware. Transfer learning, which involves leveraging models trained on other tasks, holds promise in enhancing the accuracy and adaptability of ransomware detection systems. With an expanding body of research on ransomware detection, identifying the specific machine learning algorithms and transfer learning methodologies employed in these studies has become increasingly challenging. Ransomware presents a significant security threat to enterprises due to its ability to encrypt data and restrict access to it. Article [12] introduces a novel approach aimed at enhancing the detection of ransomware attacks by analysing traffic from file-sharing platforms. Machine learning algorithms such as deep learning (DL) and transfer learning (TL) are utilized to monitor client-server communication for suspicious behaviour indicative of ransomware activity during file reading and overwriting.

Ransomware poses a significant threat across all levels, impacting individuals and large corporations alike, especially as files are accessible from diverse servers. In their research published in [12], a method is proposed to tackle this issue by monitoring file-sharing traffic for signs of crypto-ransomware. In study [18], the author employs machine learning methods to address the task of identifying maliciously encrypted messages. Detecting harmful encrypted traffic poses a significant challenge, but this paper provides a comprehensive review of current methodologies and datasets. It extensively evaluates and compares various machine learning techniques and datasets for their efficacy in detecting such activities.

Ransomware poses a severe threat by denying access to files and potentially exposing critical information. Victims often face significant challenges accessing encrypted files. Binary analysis of malware is crucial for understanding the encryption techniques employed by different ransomware variants. In article [19], the author explores the landscape of ransomware detection, detailing the criteria, factors, and tools involved in this process while comparing various methods and techniques.

II. PROPOSED METHODOLOGY

Our approach to network packet analysis integrates advanced capture techniques using tools like libpcap and WinPcap. These tools operate in promiscuous mode to intercept and log all network traffic, ensuring comprehensive visibility into data traversing the network. Post-capture, we employ rigorous filtering mechanisms based on IP addresses, port numbers, and protocols to focus our analysis on pertinent traffic, enhancing efficiency and reducing processing overhead. This meticulous filtering prepares packets for detailed header analysis, where we scrutinize essential attributes such as packet length, identification, flags like RST/SET, time to live, checksums, and port details. This deep dive into packet headers not only reveals crucial insights into network behaviour but also serves as a foundation for detecting anomalies and potential security threats, including patterns indicative of ransomware attacks.

Building upon header analysis, we extract and analyse patterns that signify suspicious activities or deviations from expected norms. By leveraging extracted patterns, we strengthen our network defences with proactive measures such as real-time packet blocking or redirection, aimed at mitigating potential threats before they escalate. Our methodology ensures robust security through the integration of reversible information embedding techniques (3.1.4), where data embedding, and recovery processes occur in the spatial domain using LSB embedding. This approach facilitates the secure transmission and retrieval of information while maintaining the integrity and reversibility of embedded data. Together, these methodologies not only enhance network security against evolving threats like ransomware but also optimize network performance by efficiently managing data flows and responses based on real-time analysis.

A. Packet capture

In this section, we describe the process of capturing live network packets and handling them to perform necessary actions such as blocking or allowing network traffic. The packet capture process involves several steps to ensure real-time monitoring and effective decision-making based on the analysis of packet content.

- **Packet Sniffing**:

We utilize packet sniffing tools or libraries (e.g., libpcap, WinPcap) to capture live network traffic. These tools allow us to intercept and log packets traveling over a network. The sniffer operates in promiscuous mode, meaning it captures all packets on the network, regardless of their destination.

- **Packet Filtering**:

Once packets are captured, we apply filters to narrow down the packets of interest. Filters can be set based on various criteria such as IP addresses, port numbers, protocols (e.g., TCP, UDP), and specific packet contents. This helps in focusing on relevant packets while ignoring unnecessary traffic.

- **Real-Time Processing**:

Captured packets are processed in real-time to extract necessary information from the packet headers and payloads. This includes parsing the packet structure, extracting header fields (e.g., source and destination IP addresses, ports), and payload data.

	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.2.6	142.250.195.10	UDP		71 51652 → 443 Len=29
2	0.007768	142.250.195.10	192.168.2.6	UDP		67 443 → 51652 Len=25
3	0.598270	192.168.2.6	192.168.199.166	TCP		66 49727 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
4	0.638659	TPLink_21:27:84	Broadcast	ARP		60 Who has 192.168.2.8? Tell 192.168.2.1
5	1.090855	192.168.2.6	138.199.14.89	TCP		55 61511 → 443 [ACK] Seq=1 Ack=1 Win=512 Len=1 [TCP segment of a reassembled PDU]
6	1.150619	fe80::2a87:baff:fe2…	ff02::1	ICMPv6		78 Router Advertisement from 28:87:ba:21:27:84
7	1.240584	192.168.2.6	13.107.6.158	TCP		55 49697 → 443 [ACK] Seq=1 Ack=1 Win=517 Len=1 [TCP segment of a reassembled PDU]
8	1.258574	138.199.14.89	192.168.2.6	TCP		66 443 → 61511 [ACK] Seq=1 Ack=2 Win=501 Len=0 SLE=1 SRE=2
9	1.277979	13.107.6.158	192.168.2.6	TCP		66 443 → 49697 [ACK] Seq=1 Ack=2 Win=16384 Len=0 SLE=1 SRE=2
10	1.283169	192.168.2.6	150.171.28.10	TCP		55 49690 → 443 [ACK] Seq=1 Ack=1 Win=517 Len=1 [TCP segment of a reassembled PDU]
11	1.287251	150.171.28.10	192.168.2.6	TCP		66 443 → 49690 [ACK] Seq=1 Ack=2 Win=16385 Len=0 SLE=1 SRE=2
12	1.613462	192.168.2.6	142.250.195.10	UDP		71 51652 → 443 Len=29
13	1.621875	142.250.195.10	192.168.2.6	UDP		67 443 → 51652 Len=25
14	2.432793	192.168.2.6	40.79.197.35	TCP		55 49699 → 443 [ACK] Seq=1 Ack=1 Win=515 Len=1 [TCP segment of a reassembled PDU]
15	2.568510	40.79.197.35	192.168.2.6	TCP		66 443 → 49699 [ACK] Seq=1 Ack=2 Win=16384 Len=0 SLE=1 SRE=2
16	3.000041	192.168.2.6	20.190.146.32	TCP		55 49703 → 443 [ACK] Seq=1 Ack=1 Win=517 Len=1 [TCP segment of a reassembled PDU]
17	3.035643	20.190.146.32	192.168.2.6	TCP		66 443 → 49703 [ACK] Seq=1 Ack=2 Win=16385 Len=0 SLE=1 SRE=2
18	3.371325	192.168.2.6	13.105.74.49	TCP		55 49704 → 443 [ACK] Seq=1 Ack=1 Win=511 Len=1 [TCP segment of a reassembled PDU]
19	3.509155	TPLink_21:27:84	Broadcast	ARP		60 Who has 192.168.2.12? Tell 192.168.2.1
20	3.590009	13.105.74.49	192.168.2.6	TCP		66 443 → 49704 [ACK] Seq=1 Ack=2 Win=2053 Len=0 SLE=1 SRE=2
21	3.917625	192.168.2.6	44.207.83.152	TCP		55 49700 → 443 [ACK] Seq=1 Ack=1 Win=63842 Len=1 [TCP segment of a reassembled PDU]
22	4.163604	44.207.83.152	192.168.2.6	TCP		54 443 → 49700 [ACK] Seq=1 Ack=2 Win=59618 Len=0
23	4.824745	192.168.2.6	142.250.195.10	UDP		71 51652 → 443 Len=29

Figure 1. Packet capturing

B. Packet header information analysis

The analysis of packet header information is a fundamental step in the process of detecting ransomware attacks. Packet headers contain crucial metadata about the data being transmitted, such as source and destination addresses, protocol information, and other control information. Here's a detailed explanation of how packet header information is analysed:

- **Header Identification:** Once the packets are captured, the headers are identified and extracted. This includes the IP header, TCP/UDP header, and other relevant protocol headers depending on the packet type.
- **Feature Extraction:** Here, there are major features are used for ransomware and some feature engineering.

Table 1. Features description

Features	**Description**
LenIP	Length of the datagram packets
IdIP	Identification of the datagram packets
FlagsIP	Flags of the datagram packets (e.g., RST, SYN)

TtlIP	Time to live for the datagram packets
ChecksumIP	Checksum value of the datagram
Sport	Source port of the packets
Dport	Destination port of the packets
Tchecksum	Checksum value for the packets to verify complete reception
Tcp_seq	Sequence number of the packet within segments
Tcp_ack	Acknowledgement number of the packet within segments
Tcp_dtaofs	Data offset field in the TCP packet segments
Tcp_flags	Flags within the segments (e.g., SYN, ACK, FIN)
Tcp_window	Window size specified in the packets
Tcp_urgptr	Urgent pointer value in the segments
URL-Type	Type of HTTP request (e.g., GET, POST)
Host-live	Indicator of whether the host is live
User-Agent	User-agent string (e.g., Mozilla, Chrome)
Accept	Indicator of whether the request is accepted
Content-type	Type of content (e.g., JSON, HTML, plain text)
Referrer	Referrer URL
Cookie	Stored cookies
Status	HTTP status code of the request (e.g., 200, 404)
Payload_len	Length of the payload
Duration	Total duration the packet is active
direction	Flow direction of the packet (uni-directional or bi-directional)

- **Data Cleaning:** Remove any irrelevant or redundant information from the header data.
- **Normalization:** Normalize the values of the extracted features to ensure they are on a consistent scale. This helps in improving the performance and convergence of the neural network during training.

C. Extract pattern of packet payload

The process of extracting **patterns** from the packet payload is a critical step in identifying potential ransomware attacks. This involves leveraging advanced deep learning models to capture and interpret complex patterns within the payload data.

Utilizing Pre-trained Models

- **VGG16 and ResNet50 Models:** Two state-of-the-art convolutional neural network (CNN) architectures, VGG16 and ResNet50, are employed to extract features from the payload. These models are chosen for their proven ability to capture hierarchical features and local patterns within the data.

- **Feature Extraction Layers:** The payload data is passed through the convolutional layers of these pre-trained models. These layers act as feature extractors, identifying important patterns and characteristics within the payload.

  - **VGG16:** Known for its simplicity and depth, VGG16 uses small convolutional filters (3x3) and deep architecture to capture intricate details of the payload.

  - **ResNet50:** Utilizes residual connections to enable the training of deeper networks without the vanishing gradient problem. This model captures even more complex patterns by maintaining the integrity of information through deeper layers.

D. Processing the Extracted Features

- **Flattening:** The output from the final convolutional layer of the VGG16 model is flattened into a 1D tensor. This transformation converts the 2D feature maps into a single-dimensional array that can be further processed by dense layers.

- **Dense Layers:** The flattened features are fed into fully connected (dense) layers. These layers learn to interpret the extracted features and identify patterns that are indicative of ransomware payloads.

E. Combining with Header Information

- The features extracted from the payload are later combined with the header information and additional features. This combined approach leverages both the detailed payload patterns and the structural information from the packet header to make a more informed classification.

F. Architecture of proposed model

- **Input Handling**

The architecture starts with two input layers: one for the payload data and one for the packet header and additional features.

- **Feature Extraction**

The payload data is passed through pre-trained VGG16 and ResNet50 models to extract rich, hierarchical features. These models are fine-tuned to capture the intricate patterns specific to ransomware payloads.

- **Payload Processing**

The output from the VGG16 model is flattened and passed through several dense layers interspersed with dropout layers to prevent overfitting. These layers progressively transform and refine the extracted features.

- **Header Processing**

The header information and additional features are processed through a separate dense layer to extract meaningful patterns from this data.

- **Concatenation and Fusion**

The features from both the payload and header are concatenated, merging the information from both parts of the packet. This combined feature vector is further processed through several dense layers to learn the final decision boundary.

- **Final Decision Making**

The final dense layer outputs a single value between 0 and 1 using a sigmoid activation function. A value closer to 1 indicates that the packet is classified as ransomware, while a value closer to 0 indicates a normal packet.

III. EXPERIMENTS & RESULTS

To comprehensively evaluate the performance of our proposed depression detection approach, we analysed the results across several metrics. The classification accuracy provides an overall sense of the model's predictive performance. However, due to the class imbalance in our dataset (with more non-depressed users than depressed), accuracy alone may not tell the full story.

Therefore, we also report the precision, recall, and F1 scores broken down by class. The precision for the depressed class indicates how many of the instances predicted as depressed were truly depressed according to the ground truth labels. Recall measures what proportion of the actual depressed cases were successfully identified by the model. The F1 score combines precision and recall into a single metric.

For the non-depressed class, high precision ensures few users are incorrectly flagged as potentially depressed when they are not. High recall is important to catch as many truly non-depressed cases as possible.

Table 2. Performance analysis of Model

Sr. No.	Feature Type	Model	Epochs	Batch Size	Accuracy
01.	Network based feature	DNN	50	64	0.9775
02.	Payload based feature	RNN	50	32	0.933
		CNN1D	50	32	1.00
03.	Packet Behaviour based feature	VGG16	50	64	0.943
		RESTNET 50	50	64	0.98

3.1 METRICS PLOT

A. Loss curve for training and validation data

Figure 4.1 shows a plot of training and validation loss against the number of training epochs, illustrating the model's performance during training. As the model is trained, it ideally reduces the training loss over time. The validation loss is used to monitor the model's ability to generalize to unseen data, helping to avoid overfitting.

This metric evaluates how well the model's losses in identifying depressed cases generalizes to unseen data beyond the training examples.

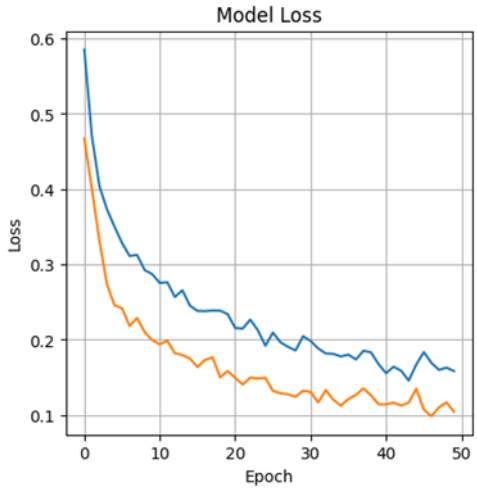

Figure 2. Loss of training and validation data

B. Precision Curve for Training and Validation

The blue curve in the figure below depicts the model's precision on the depressed class for the training data. Ideally, this precision should increase over time, which is observed in the figure 4.2 as the model learns to correctly identify true positive cases of depression in the training examples. The orange curve represents the model's precision on the depressed class for a separate validation dataset.

This metric evaluates how well the model's precision in identifying depressed cases generalizes to unseen data beyond the training examples. Monitoring the validation precision curve helps us avoid overfitting the model to the specific characteristics of the training data.

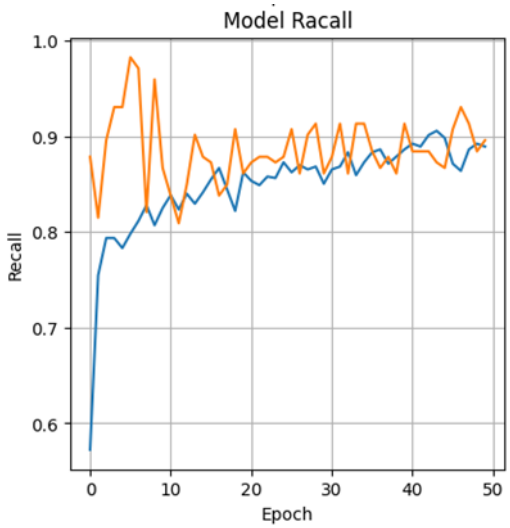

Figure 3. Precision of training and validation data

RECALL CURVE FOR TRAINING AND VALIDATION DATA

The blue curve in the figure below shows the model's recall on the depressed class for the training data. Ideally, this value should increase over time, as the model learns to correctly identify and retrieve a higher proportion of true positive cases of depression in the training examples. The orange curve represents the model's recall on the depressed class for a separate validation dataset. This metric evaluates how well the model's ability to detect depressed cases generalizes to unseen data beyond the training examples. Monitoring the validation recall curve helps to avoid overfitting the model to the specific characteristics of the training data.

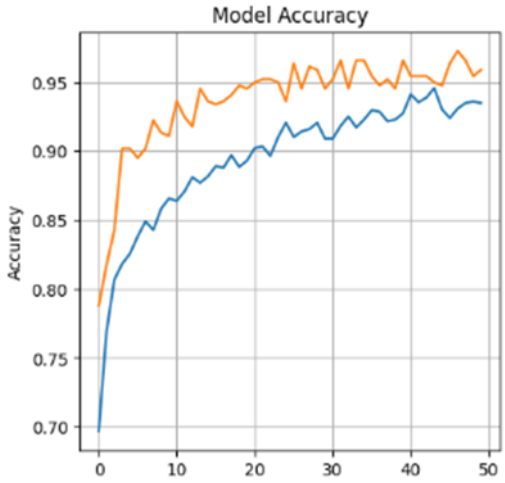

Figure 4. Recall of training and validation data

ACCURACY CURVE FOR TRAINING AND VALIDATION DATA

The blue curve in the figure below illustrates the model's accuracy on the training data. Ideally, this value should increase over time, as observed in the figure, indicating that the model is learning to correctly segment the training examples.

The orange curve represents the model's accuracy on a separate validation dataset, which helps assess how well the model generalizes to unseen data and avoids overfitting to the training data. The perceptual quality matrix of the proposed algorithm is presented below.

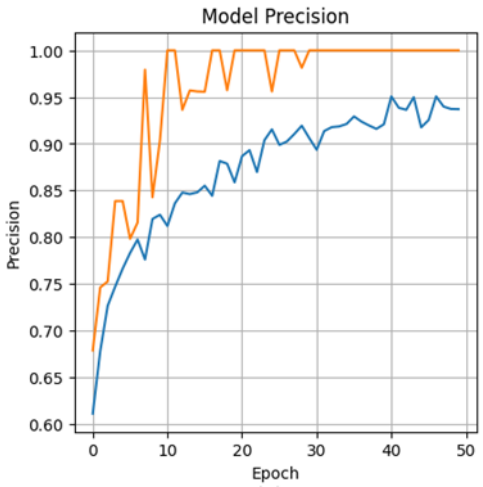

Figure 5. Accuracy of training and validation data

IV. CONCLUSION AND FUTURE WORK

We introduced a novel and powerful hybrid model, the New Model Architecture, to classify ransomware attacks and using packet analysis, addressing a challenging issue in the field. The New Model Architecture combines

Convolutional Neural Networks (CNNs) and pretrained transformers to leverage the hierarchical features and local patterns in the data.

We extensively tested and compared the performance of the proposed model architecture with two other popular models, ResNet50 and VGG16. The proposed model achieved impressive results, with a 99.6% training accuracy and a 99.1% testing accuracy. Its ability to distinguish between ransomware attacks and regular occurrences is supported by its superior performance in terms of precision, recall, and F1 score compared to competing models.

To demonstrate the New Model Architecture's superiority over existing models, we compared it to cutting-edge research in the field. The comparative analysis identified the proposed model as a promising state-of-the-art solution for ransomware detection and classification.

Overall, the proposed model represents a significant advancement in identifying and categorizing ransomware attacks. By capturing both global and local patterns, CNNs and pretrained transformers effectively separate benign from harmful data. The model's high levels of accuracy, precision, and recall, along with its capacity to handle the complexity packet analysis, make it an attractive option for enhancing cloud security and protecting against ransomware.

V. FUTURE WORK

Building upon the findings of this research, future work can focus on enhancing the performance and robustness of the proposed ransomware detection model. One avenue for improvement is to explore more advanced deep learning architectures or hybrid models that combine multiple approaches, such as integrating recurrent neural networks (RNNs) or attention mechanisms alongside convolutional neural networks (CNNs) and pretrained transformers. This approach could potentially capture more complex patterns in encrypted data, leading to improved accuracy and reliability in distinguishing ransomware attacks from normal data activities.

Additionally, conducting extensive experiments with larger and more diverse datasets, including real-world data, would provide further insights into the model's generalization capabilities and scalability.

VI. REFERENCES

[1] Or Ami, Yuval Elovici, and Danny Hendler. Ransomware prevention using application authentication-based le access control. *2018.*

[2] Al-Fawa'reh, M.; Ashi, Z.; Jafar, M.T. Detecting Malicious DNS Queries over Encrypted Tunnels Using Statistical Analysis and Bi-Directional Recurrent Neural Networks. Karbala Int. J. Mod. Sci. 2021, 7, 268–280. [https://doi.org/10.33640/2405-609X.3155]

[3] Jegede, A.; Fadele, A.; Onoja, M.; Aimufua, G.; Mazadu, I.J. Trends and Future Directions in Automated Ransomware Detection. J. Comput. Soc. Inform. 2022, 1, 17–41. [https://doi.org/10.33736/jcsi.4932.2022]

[4] Fernando, D.W.; Komninos, N.; Chen, T. A Study on the Evolution of Ransomware Detection Using Machine Learning and Deep Learning Techniques. Internet Things 2020, 1, 551–604. [https://doi.org/10.3390/iot1020030]

[5] Horduna, M.; Lazarescu, S.; Simion, E. A note on machine learning applied in ransomware detection. Int. Assoc. Cryptologic Res 2023, 17. Available online: https://eprint.iacr.org/2023/045.pdf (accessed on 2 June 2023).

[6] Hsu,C.M.; Yang, C.C.; Cheng, H.H.; Setiasabda, P.E.; Leu, J.S. Enhancing File Entropy Analysis to Improve Machine LearningDetection Rate of Ransomware. IEEE Access 2021, 9, 138345–138351. [https://doi.org/10.1109/ACCESS.2021.3114148]

[7] Smith, D.; Khorsandroo, S.; Roy, K. Machine Learning Algorithms and Frameworks in Ransomware Detection. IEEE Access 2022,10, 117597–117610. [https://doi.org/10.1109/ACCESS.2022.3218779]

[8] Berrueta, E.; Morato, D.; Magaña, E.; Izal, M. Crypto-ransomware detection using machine learning models in file-sharing network scenarios with encrypted traffic. Expert Syst. Appl. 2022, 209, 118299. [https://doi.org/10.1016/j.eswa.2022.118299]

[9] Kapoor, A.; Gupta, A.; Gupta, R.; Tanwar, S.; Sharma, G.; Davidson, I.E. Ransomware detection, avoidance, and mitigation scheme: A review and future directions. Sustainability 2021, 14, 8. [https://doi.org/10.3390/su14010008].

[10] Cohen, A.; Nissim, N. Trusted detection of ransomware in a private cloud using machine learning methods leveraging meta features from volatile memory. Expert Syst. Appl. 2018, 102, 158–178. [https://doi.org/10.1016/j.eswa.2018.02.039].

[11] Yamany, B.; Elsayed, M.S.; Jurcut, A.D.; Abdelbaki, N.; Azer, M.A. A New Scheme for Ransomware Classification and Clustering Using Static Features. Electronics 2022, 11, 3307. [https://doi.org/10.3390/electronics11203307]

[12] Nkongolo, M.; van Deventer, J.P.; Kasongo, S.M.; Zahra, S.R.; Kipongo, J. A Cloud Based Optimization Method for Zero-Day Threats Detection Using Genetic Algorithm and Ensemble Learning. Electronics 2022, 11, 1749. [https://doi.org/10.3390/electronics11111749]

[13] Nenvani, G.; Gupta, H. A survey on attack detection on cloud using supervised learning techniques. In Proceedings of the 2016 Symposium on Colossal Data Analysis and Networking, CDAN 2016, Indore, India, 18–19 March 2016; Volume 175, pp. 21–27. [https://doi.org/10.1109/CDAN.2016.7570872]

[14] Zhao, K.; Jia, F.; Shao, H. A novel conditional weighting transfer Wasserstein auto-encoder for rolling bearing fault diagnosis with multi-source domains. Knowl.-Based Syst. 2023, 262, 110203. [https://doi.org/10.1016/j.knosys.2022.110203]

[15] Jegede, A.; Fadele, A.; Onoja, M.; Aimufua, G.; Mazadu, I.J. Trends and Future Directions in Automated Ransomware Detection. J. Comput. Soc. Inform. 2022, 1, 17–41. [http://dx.doi.org/10.33736/jcsi.4932.2022]

[16] Azmoodeh, A.; Dehghantanha, A.; Conti, M.; Choo, K.K.R. Detecting crypto-ransomware in IoT networks based on energy consumption footprint. J. Ambient Intell. Humaniz. Comput. 2018, 9, 1141–1152. [http://dx.doi.org/10.1007/s12652-017-0558-5]

[17] Talabani, H.S.; Abdulhadi, H.M.T. Bitcoin ransomware detection employing rule-based algorithms. Sci. J. Univ. Zakho 2022, 10, 5–10. [http://dx.doi.org/10.25271/sjuoz.2022.10.1.865]

[18] Paquet-Clouston, M.; Haslhofer, B.; Dupont, B. Ransomware payments in the bitcoin ecosystem. J. Cybersecur. 2019, 5, tyz003. [http://dx.doi.org/10.1093/cybsec/tyz003]

[19] Azmoodeh, A.; Dehghantanha, A.; Conti, M.; Choo, K.K.R. Detecting crypto-ransomware in IoT networks based on energy consumption footprint. J. Ambient Intell. Humaniz. Comput. 2018, 9, 1141–1152. [http://dx.doi.org/10.1007/s12652-017-0558-5]

[20] Celdrán, A.H.; Sánchez, P.M.S.; Castillo, M.A.; Bovet, G.; Pérez, G.M.; Stiller, B. Intelligent and behavioral-based detection of malware in IoT spectrum sensors. Int. J. Inf. Secur. 2022, 22, 541–561. [http://doi.org/10.1007/s10207-022-00602-w]

[21] Azmoodeh, A.; Dehghantanha, A.; Conti, M.; Choo, K.K.R. Detecting crypto-ransomware in IoT networks based on energy consumption footprint. J. Ambient Intell. Humaniz. Comput. 2018, 9, 1141–1152. [http://dx.doi.org/10.1007/s12652-017-0558-5]

[22] Sheen, S.; Asmitha, K.; Venkatesan, S. R-Sentry: Deception based ransomware detection using file access patterns. Comput. Electr. Eng. 2022, 103, 108346. [http://dx.doi.org/10.1016/j.compeleceng.2022.108346]

[23] Kok, S.; Azween, A.; Jhanjhi, N. Evaluation metric for crypto-ransomware detection using machine learning. J. Inf. Secur. Appl. 2020, 55, 102646. [http://dx.doi.org/10.1016/j.jisa.2020.102646]

[24] Gorment, N.Z.; Selamat, A.; Cheng, L.K.; Krejcar, O. Machine Learning Algorithm for Malware Detection: Taxonomy, Current Challenges and Future Directions. IEEE Access 2023, 1. [http://dx.doi.org/10.1109/ACCESS.2023.3256979]