ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

International Journal for Electronic Crime Investigation

(IJECI)

VOLUME: 8
ISSUE: 2 Apr-Jun 2024

Email ID: ijeci@lgu.edu.pk

Digital Forensics Research and Service Center
Lahore Garrison University, Lahore, Pakistan.

International Journal for Electronic Crime Investigation

Volume 8(2) Apr-Jun 2024

SCOPE OF THE JOURNAL

The International Journal for Crime Investigation IJECI is an innovative forum for researchers, scientists and engineers in all the domains of computer science, white Collar Crimes, Digital Forensics, Nano Forensics, Toxicology and related technology, Criminology, Criminal Justice and Criminal Behaviour Analysis. Moreover, the scope of the journal includes algorithm, high performance, Criminal Data Communication and Networks, pattern recognition, image processing, artificial intelligence, VHDL along with emerging domains like quantum computing, IoT, Hacking. The journal aims to provide an academic medium for emerging research trends in the general domain of crime investigation.

SUBMISSION OF ARTICLES

We invite articles with high quality research for publication in all areas of engineering, science and technology. All the manuscripts submitted for publication are first peer reviewed to make sure they are original, relevant and readable. Manuscripts should be submitted via email only.

To submit manuscripts by email with attach file is strongly encouraged, provided that the text, tables, and figures are included in a single Microsoft Word/Pdf file.

Contact: For all inquiries, regarding call for papers, submission of research articles and correspondence, kindly contact at this address:
IJECI, Sector C, DHA Phase-VI Lahore, Pakistan
Phone: +92- 042-37181823
Email: ijeci@lgu.edu.pk

International Journal for Electronic Crime Investigation
Volume 8(2) Apr-Jun 2024

International Journal for Electronic Crime Investigation
Volume 8(2) Apr-Jun 2024

CONTENTS

Editorial

Bridging the Gap Between Investigating Electronic and Traditional Crimes in Pakistan

Kaukab Jamal Zuberi

1. Introduction

In an era where technology permeates every facet of life, the nature of crime has evolved. Traditional crimes, which are often physical and local in nature, now coexist with electronic crimes that transcend geographical boundaries and manifest in the virtual world. This paradigm shift necessitates a corresponding evolution in investigative methodologies. In Pakistan, the gap between the mechanisms for investigating traditional and electronic crimes must be bridged to ensure robust law enforcement and justice delivery in the digital age.

2. The Nature of Traditional Crimes vs. Electronic Crimes

Traditional Crimes: These include offenses such as theft, burglary, assault, and homicide. Investigations rely heavily on physical evidence, witness testimonies, and forensic analysis. The process is often straightforward, requiring on-site investigations, interrogations, and the collection of tangible evidence.

Electronic Crimes: Also known as cybercrimes, these encompass a wide range of illegal activities conducted through digital means. This includes hacking, identity theft, online fraud, cyberbullying, and ransomware attacks. The evidence is often intangible, hidden in data trails, requiring specialized skills to uncover and analyse digital footprints. Cybercrimes can be perpetrated from any location, making jurisdiction and enforcement more complex.

3. Challenges in Investigating Electronic Crimes

1. Technical Expertise: Unlike traditional crimes, electronic crimes require investigators to possess a deep understanding of information technology, cybersecurity, and digital forensics. This technical expertise is currently limited in Pakistan, where law enforcement agencies are often not equipped with the necessary skills and tools.

2. Jurisdictional Issues: Electronic crimes can be committed from anywhere in the world, complicating jurisdictional boundaries. Coordinating with international agencies and navigating the legal frameworks of different countries is a significant challenge.

3. Rapid Technological Advancement: The fast-paced nature of technology means that cybercriminals often stay ahead of law enforcement. Continuous education and training are essential for investigators to keep up with new methods of cybercrime.

4. Resource Allocation: Investigating cybercrimes requires significant resources, including advanced software, specialized hardware, and trained personnel. In Pakistan, where resources are often limited, prioritizing cybercrime

investigation can strain the capacity to address traditional crimes.

4. The Need for Development in Pakistan

To effectively combat the rising tide of electronic crimes, Pakistan must undertake comprehensive reforms to develop its investigative capabilities.

1. Training and Education: Law enforcement agencies must invest in specialized training programs to equip officers with the necessary skills to tackle cybercrimes. Partnerships with academic institutions and international bodies can facilitate knowledge transfer and capacity building.

2. Infrastructure and Technology: Upgrading the technological infrastructure within investigative agencies is crucial. This includes acquiring advanced digital forensics tools, secure communication channels, and robust data analytics platforms.

3. Legislation and Policy: Strengthening legal frameworks to address cybercrimes is essential. Clear policies on data protection, cyber ethics, and international cooperation must be established to provide a solid foundation for investigations.

5. Specific Legislation and Policy Recommendations

Pakistan needs to revise and introduce several key legislations to effectively investigate electronic crimes:

1. Prevention of Electronic Crimes Act (PECA) 2016: While this act is a significant step toward addressing cybercrimes, it requires amendments to enhance its effectiveness. For instance, updating definitions to cover emerging cyber threats, specifying clearer guidelines for the collection and admissibility of digital evidence, and ensuring data protection and privacy rights are crucial.

2. Electronic Transactions Ordinance 2002: This ordinance should be revised to incorporate stronger cybersecurity measures and ensure that electronic signatures and records are protected against tampering and unauthorized access.

3. Data Protection Law: Pakistan currently lacks comprehensive data protection legislation. Enacting a robust data protection law that aligns with international standards, such as the General Data Protection Regulation (GDPR) of the European Union, is essential. This law should mandate organizations to implement stringent data security measures and report data breaches promptly.

4. Cybercrime Coordination and Response Framework: Establishing a dedicated national framework to coordinate responses to cyber incidents across various agencies is necessary. This framework should facilitate information sharing, joint investigations, and rapid response to cyber threats.

5. International Cooperation Agreements: Strengthening bilateral and multilateral agreements with other countries for mutual legal assistance in cybercrime investigations is vital. These agreements should streamline the process of cross-border data sharing, extradition of

cybercriminals, and joint operations against international cyber threats.

6. Intellectual Property Laws: Updating intellectual property laws to address online piracy and the illegal distribution of copyrighted material is crucial. Enhancing penalties for cyber infringements and ensuring swift enforcement can help protect intellectual property rights in the digital age.

6. Public Awareness

Educating the public about cyber threats and promoting safe online practices can help reduce the incidence of electronic crimes. Awareness campaigns can empower citizens to protect themselves and assist law enforcement through prompt reporting of cybercrimes.

7. International Collaboration

Building strong ties with international cybercrime units and participating in global forums can enhance Pakistan's ability to address cross-border electronic crimes. Collaborative efforts can lead to the sharing of best practices, resources, and intelligence.

8. Conclusion

The landscape of crime is evolving, and Pakistan must adapt its investigative strategies to effectively address both traditional and electronic crimes. Bridging this gap requires a concerted effort to enhance technical expertise, modernize infrastructure, and foster international cooperation. By investing in these areas, Pakistan can create a safer environment for its citizens in both the physical and digital realms. The future of law enforcement in Pakistan depends on its ability to navigate this complex duality and emerge as a leader in the fight against crime in all its forms.

International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2024.0802192

Vol. 8 issue 2 Apr-Jun 2024

Enhanced Malware Detection Using Deep Learning: A Comprehensive Framework for Feature Extraction and Classification

Zohaib Ahmad

Faculty of Electronics and Information Engineering, Beijing University of Technology, Beijing, China.

Correspondence Author: ahmedzohaib03@gmail.com

Received: March 21, 2024; **Accepted:** April 27, 2024; **Published:** June 14, 2024

ABSTRACT

The exponential growth and sophistication of malware necessitate new detection strategies. The rapid evolution of malware makes traditional manual heuristic practices ineffective in perceiving new malware variants. Machine learning systems have proven essential for automating the dynamic and static analysis, as they cluster similar malware behaviors and classify new infections based on their similarity to approved malware families. This research validates that deep learning networks can accomplish higher accuracy than customary machine learning approaches. Deep learning has multiple neural network layers, which allows it to better automatically ascertain and classify malware variants. It offers a framework for removing multiple signature sets, including parts, opcode, bytecode, and system calls, from malware files. Experimental consequences indicate that the most accurate feature vector is the feature vector generated through system calls. This study concludes that deep learning approaches outperform traditional shallow machine learning systems in terms of malware recognition and classification precision.

Keywords: Malware analysis, deep learning, feature extraction, neural networks

1. INTRODUCTION

Malware refers to malicious software designed to compromise the reliability, security, and operation of a computer structure without the user's permission. This type of software can attain the attacker's malicious objectives, such as hijacking the computer or stealing data. Modern antivirus programs rely on signature-based revealing techniques that generate various signatures for identified malware and store them in a

database for later identification. However, signature-based procedures are limited in effectiveness since they cannot identify malware whose signatures are not yet recorded in the database. Attackers often use procedures such as polymorphism, encryption and obfuscation to evade detection, making it problematic for signature-based systems to keep up with the ever-changing threat landscape [1] [2].

To address this encounter, static and dynamic analysis techniques are used to find variants of known threats. Signatures generated by these techniques can be used to classify unknown samples into existing families and group malware. Static analysis examines code without executing it to find configurations and extract data i.e. text, byte sequence, call graphs and opcodes, and. Memory dump tools retrieve and examine protected code from system memory, while disassembler tools reverse engineer executables to generate assembly instructions. However, static analysis is difficult because obfuscation, encryption, polymorphism, and transformations hinder decompilation [3] [4].

Malicious code is dynamically analyzed using a virtual or controlled environment and tools such as Process Explorer, Regshot, and Process Monitor. This approach looks closely at instruction traces, function parameters, function calls, etc. [5][6] Dynamic analysis does not destroy executables, but records malware behavior into a feature vector space, which is time-consuming and resource-intensive. Also, some viruses may behave differently in virtual environments or be triggered in certain circumstances, making them difficult to identify [7] [8].

Machine learning systems can already automate malware analysis and classification, reducing the need for thorough manual investigation, even if static and dynamic procedures are still necessary. Unknown malware is categorized into different families using machine learning procedures i.e. clustering and classification, which can look for patterns in static and dynamic examination [9][10].

Determining if a file contains malware-filled is a classification issue. To do this, we have used various machine learning procedures including Support Vector Machine (SVM), and Decision Tree. Typically, a dataset covers files that are to be classified as benign or malicious. A model has been trained using the training samples of these datasets, which are alienated into a test set and a training set. Cross-validation procedures enhance the evaluation of the model. The quantity of properly classified documents determines the quality of the model, and the model can predict the labels of the test set after training. However, common machine learning practices are limited in precision by their shallow learning architecture, while deep learning procedures with more automated feature learning capabilities can advance accuracy [11][12].

Deep learning procedures incorporate many layers to extract information from the lowest to the highest layer. After each layer detects a particular feature and forward it to the subsequent layer, the subsequent layer merges the features of lower-level into the features of the higher-level. This aggregation process continues until the final layer determines whether the file is benign or malicious. Deep learning models can extract features on their own, whereas traditional machine learning necessitates features to be explicitly served into the network. In this training, we use the dataset to test machine and deep learning performances and compare the outcomes.

Given the preceding discussion, the following key point's summaries the main contributions and findings of this research paper.

- Draw attention to how deep learning networks can outclass machine learning performances in terms of accuracy.
- Provide a comprehensive approach to remove signature sets such as opcodes, bytecode, and system calls from malware files.
- Explain how to find variants of known malware threats using static and dynamic study performances.
- Classify malware using various machine-learning systems
- Explain how to train and evaluate these algorithms using labelled datasets.
- Explain how deep learning models can self-extract features compared to traditional machine learning procedures that require manual input of features.
- Demonstrate that deep learning techniques can significantly advance detection and classification precision.
- Provide experimental evidence that deep learning performances offer high accuracy and performance compared to traditional shallow machine learning practices.

The remainder of the document is organized as: Related work is offered in Section 2, our deep-learning approach for malware revealing is familiarized in Section 3, and its effectiveness is assessed in Section 4 when compared to other malware detection alternatives. Next comes Section 5, which brings the paper to a conclusion.

2. RELATED STUDIES

Machine learning techniques have been widely used in various fields such as recommender systems [18], PVC pipe break detection [15][16][17], sentiment analysis [14], and wildfire detection [13]. Applying these methods to the field of malware detection is the main goal of this research.

Traditional antivirus software often fails to identify malware variants due to missing signatures in their databases. Machine learning systems overcome this limitation by identifying small patterns of behavior and classifying these variants into known families.

The studies listed below demonstrate the development of machine learning based malware classification.

- In [19], the authors used static analysis to obtain text, byte sequence, and system resource information. For classification, they used algorithms such as RIPPER, Naive Bayes, and multi-classification systems.
- In [20], the authors used bytecode n-grams as features to evaluate techniques including Naive Bayes, SVM and decision trees. The decision tree algorithm performed better than other algorithms.
- To compare and classify malware programs into families based on the similarity of their call graphs, the authors in [21] proposed a system that utilizes call graphs as features and distance measures.
- The authors in [22] use K-nearest neighbors to classify malware and visualize it using image processing procedures.
- The authors in [23] showed that function length and frequency are important in identifying malware families by classifying Trojans based on these characteristics.
- The authors in [24] studied the use of labeled and unlabeled examples for malware classification in semi-supervised learning systems.
- To significantly reduce the runtime overhead, the authors in [25] projected an incremental method for behavior type analysis that combines the clustering and classification structure.

- The authors in [26] used decision tree method and the random forest systems to categorize worms and identify them using variable length instruction sequences.
- Using pcap files, the developers [27] focused on the malware's network activity. Using the J48 decision tree classifier, they were able to achieve good classification results by extracting flow information to create a behavior graph and using features that reflect the network behavior.
- The authors of [28] use a graph structure consisting of malware behavior to analyze dynamic analysis data. They then classify the data using an SVM trained on a similarity matrix. These papers show the progression of malware detection strategies from static and dynamic analysis to more complex machine learning procedures, which improve classification accuracy and flexibility in identifying new malware variants.

3. MODEL DEVELOPMENT METHODOLOGY

The process for constructing the malware detection model consists of many main steps:

3.1. Data Collection

The data used to train and evaluate the model came from Microsoft's malware dataset, which is accessible on Kaggle. This collection contains 10,868 malware files from distinct families, including Ramnit, Kelihos_ver3, Simda, Vundo Tracur, Obfuscator.ACY, Kelihos_ver1, and Gatak. Each file in the collection has a unique identity and a class label that indicates its family.

3.2. Data Preprocessing

The raw data must be converted into a feature vector space before analysis can begin. The dataset is used to create four different kinds of feature sets:

- 00 to FF are the hexadecimal codes that designate the frequency of bytecodes.
- Opcode frequency: The frequency of machine language commands like CMP, ADD and SUB.
- The frequency of sections in an object file, such as.init,.text, and.bss, are relocatable sections.
- System calls with one-hot encoding: API calls made by user applications to get kernel services

Following a min-max scaler approach standardization, these characteristics are randomized and divided into training (80%) and testing (20%) samples. Cross-validation, which divides the training samples into several lesser samples for training and the samples for validation, are used to prevent overfitting.

3.3 Feature Extraction and Analysis

To produce an extensive input for the model, the processed feature sets are examined. For classification, a variety of machine-learning systems are used, such as SVM approach, Decision Trees and Naïve Bayes. Cross-validation is utilized to separate the dataset into the training samples and testing samples to advance model assessment.

3.4. Model Training and Validation

The dataset is allocated into training and testing subsets to solve the classification issue. The system is trained on the sample of training, and its precision is calculated on the samples of testing. Additionally, deep learning techniques are used, extracting characteristics from the level of lowest to the highest across numerous layers. The file is categorized as either benign or malicious by the last layer.

4. PERFORMANCE MEASURE

The confusion matrix was utilized to construct the following measures, which was utilized to weigh the efficacy of deep neural networks (DNN) and machine learning processes for malware classification:

- TP (True Positive): The quantity of files that were precisely categorized as harmful.
- The quantity of files correctly categorized as benign (TN, True Negative).
- FP (False Positive): The files those were incorrectly identified as harmful.
- FN (False Negative): The files those were incorrectly identified as benign.

Key performance metrics include:

- Percentage of real harmful files that are correctly categorized (TP / (TP + FN)) is identified as the True Positive Rate (TPR) or recall.
- The ratio of benign files mistakenly labelled as malicious (FP / (FP + TN)) is known as the False Positive Rate (FPR).
- F1-Score: ((2 * accuracy * Recall) / (Precision + Recall)) is the harmonic mean of accuracy and recall.
- Precision: Total proportion of accurately categorized files ((TP + TN) / (TP + TN + FP + FN)).
- Precision: The proportion of harmful files accurately recognized among all files designated as malicious (TP / (TP + FP)).

5. EXPERIMENTAL RESULTS

In the field of malware analysis, traditional research has focused on binary classification, classifying data into two categories: harmful and benign. Our research, on the other hand, seeks to predict specific malware file family types by solving the more difficult multi-class classification task.

Our strategy involves applying machine learning practices to create nine binary classifiers, each specialized for predicting a single malware family, to solve the multi-class classification challenge. The class that receives the highest score from each classifier is assigned as the expected class during classification. Interestingly, the random forest classifier functions differently as it does not require the creation of a binary classifier beforehand; instead, it immediately classifies instances into several classes.

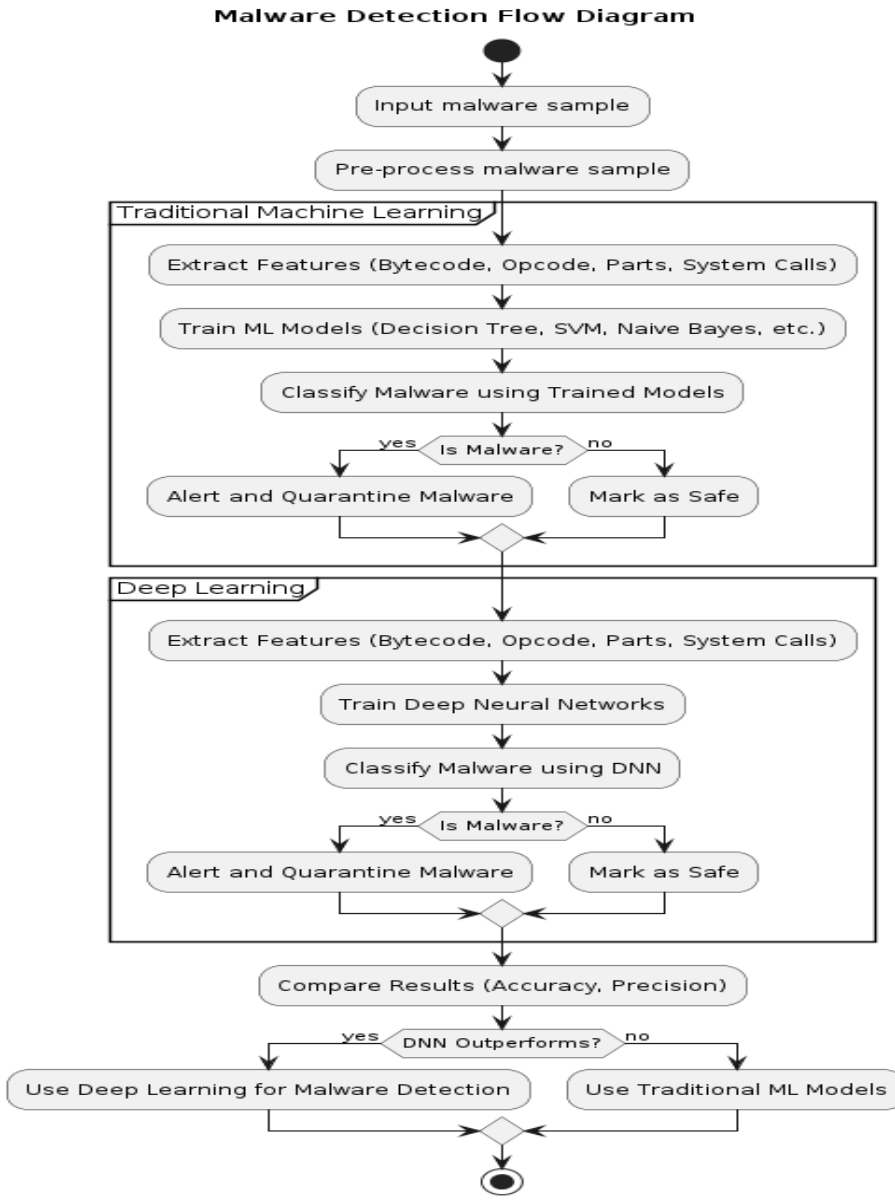

Figure 1: Flow diagram for malware detection

5.1 Evaluation Method

We used a rigorous 3-fold cross-validation method for assessment. This technique splits the dataset into three subsets: one-third is used to assess the model's performance, while the other two-thirds are used to train the algorithms.

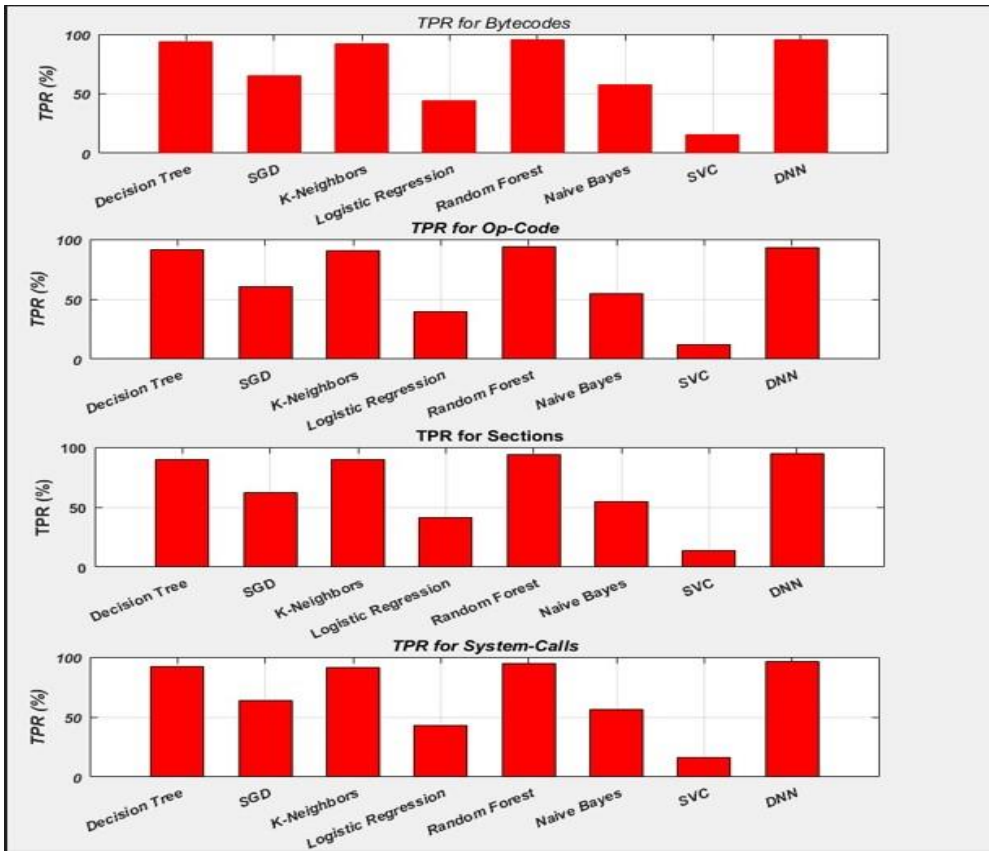

Figure 2 (a): Comparison of TPR for Different Algorithms (Bytecodes, Op-Code, Sections, and System-Calls)

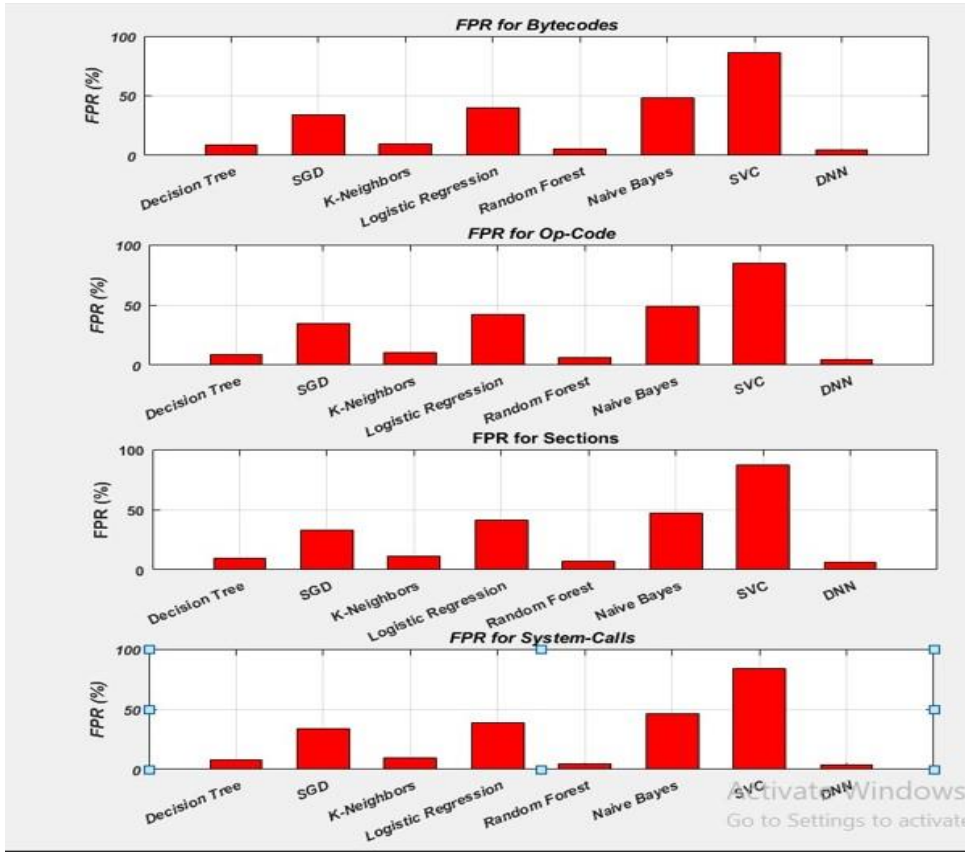

Figure 2 (b): Comparison of FPR for Different Algorithms (Bytecodes, Op-Code, Sections, and System-Calls)

5.2. Performance Metrics Comparison

We contrasted the accuracy, True Positive Rate (TPR), and False Positive Rate (FPR) of deep learning strategies with machine learning performances based on shallow learning.

Figure 2, Byte-Code, Sections, Op-Code, and System-Calls: illustrate the TPR and FPR for different feature representations used in the classification process. Across all representations, deep-learning models exhibit superior TPR and lower FPR compared to shallow-learning methods.

The success of suggested deep learning malware classification system is attributed to the effective use of the gradient descent and back-propagation performances. These mechanisms facilitate iterative adjustment of model weights to minimize loss, thereby enhancing overall accuracy, and TPR, and reducing FPR.

6. CONCLUSION

This study offers a thorough framework for deep learning-based improved malware detection. In comparison to conventional machine learning techniques, the suggested method provides improved accuracy by using feature samples include operational codes, system calls, byte codes and sections. According to the experimental results, deep learning models greatly enhance malware variant identification and classification, offering a reliable countermeasure to the constantly changing threat landscape. Subsequent research endeavors will centre on refining the deep learning models and investigating supplementary feature sets to augment detection proficiency.

REFERENCES

[1] S. Jha, O. Sheyner, and J. Wing, "Two formal analyses of attack graphs." *In Proceedings 15th IEEE Computer Security Foundations Workshop. CSFW-15*, IEEE, pp. 49-63, 2002.

[2] M. Christodorescu and S. Jha, "Testing malware detectors." *ACM SIGSOFT Software Engineering Notes,* vol. 29, no. 4, pp.34-44, 2004.

[3] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda, "Scalable, behavior-based malware clustering." In *NDSS*, vol. 9, pp. 8-11, 2009.

[4] Z. Li, A. Goyal, Y. Chen, and V. Paxson, "Automating analysis of large-scale botnet probing events." *In Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pp. 11-22, 2009.

[5] Y. Ye, D. Wang, T. Li, and D. Yet, "An intelligent PE-malware detection system based on association mining." *Journal in computer virology,* vol. 4, pp.323-334, 2008.

[6] H. S. Anderson and P. Roth, "Ember: an open dataset for training static pe malware machine learning models." *arXiv preprint arXiv: 1804.04637* ,2018.

[7] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and J. Blasco, "Dendroid: A text mining approach to analyzing and classifying code structures in android malware families." *Expert Systems with Applications*, vol. 41, no. 4, pp. 1104-1117, 2014.

[8] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware

images: visualization and automatic classification." In Proceedings of the 8th international symposium on visualization for cyber security, pp. 1-7. 2011.

[9] S. Poudyal, “Multi-level analysis of Malware using Machine Learning”. *The University of Memphis*, 2021.

[10] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection." In 2010 IEEE symposium on security and privacy, IEEE, pp. 305-316. 2010.

[11] K. Rieck, P. Trinius, C. Willems, and T. Holz, "Automatic analysis of malware behavior using machine learning." *Journal of computer security*, vol.19, no. 4, pp. 639-668, 2011.

[12] N. Milosevic, A. Dehghantanha, and K.-K. R. Choo, "Machine learning aided Android malware classification." *Computers & Electrical Engineering*, vol.61, pp.266-274, 2017.

[13] W. Lidong, Z. Huixi, Z. Yin, H. Keyong and A. Kang, "A deep learning-based experiment on forest wildfire detection in machine vision course." *IEEE Access*, vol.11, pp. 32671-32681, 2023.

[14] C. Anmol, M. Pulkit and G. Mohit, "Sentiment analysis of text using deep convolution neural networks." *In 2017 Tenth international conference on contemporary computing (IC3),* IEEE, pp. 1-6, 2017.

[15] Y. Xiangyang, Z. Jian’e, L. Ruohan ,W. Yajiao, G. Lili, Y. Zhonghan ,and H. Xiaoqing, "Diagnosis of sewer pipe defects on image recognition of multi-features and support vector machine in a southern Chinese city." *Frontiers of Environmental Science & Engineering*, vol.13, pp. 1-13, 2019.

[16] M. Adrien, K. Nikos, R. Christian C,and M. Dirk, "Machine learning classifiers for surface crack detection in fracture experiments.″ *International Journal of Mechanical Sciences*, vol. 209, pp. 106698, 2021.

[17] K. M Safeer, Z. Kaiman, W. Nansong, and U. Ishaq, "Robotics and Deep Learning Framework for Structural Health Monitoring of Utility Pipes." *In 2019 IEEE International Symposium on Measurement and Control in Robotics (ISMCR),* IEEE, pp. B3-3. 2019.

[18] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new persp ectives." *ACM computing surveys (CSUR)*, vol. 52, no. 1, pp. 1-38, 2019.

[19] D. Schultz, E. Eskin, F. Zadok, and S. Stolfo, "Data mining methods for detection of new malicious executables." *In Proceedings 2001 IEEE Symposium on Security and Privacy. S&*P 2001, IEEE, pp. 38-49, 2000.

[20] N. Lakshmanan, J. Gregoire and M, BS "Detecting packed executables based on raw binary data." *Technical report*, 2010.

[21] W. Hu, W. Hu, and S. Maybank, "Adaboost-based algorithm for network intrusion detection." *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 38, no. 2, pp.577-583, 2008.

[22] J. Z. Kolter and M. A. Maloof, "Learning to detect and classify malicious executables in the wild."

Journal of Machine Learning Research, vol.7, no. 12, 2006.

[23] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto, and W. Lee, "McPAD: A multiple classifier system for accurate payload-based anomaly detection." *Computer networks*, vol.53, no. 6, pp. 864-881, 2009.

[24] M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov, and G. Giacinto, "Novel feature extraction, selection and fusion for effective malware family classification." *In Proceedings of the sixth ACM conference on data and application security and privacy,* pp. 183-194. 2016

[25] L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection." *IEEE Communications surveys & tutorials,* vol. 18, no. 2, pp. 1153-1176, 2015.

[26] S. A. Yeswanth, N.K. Reddy, B and V. R. Gupta, "Malware Detection Using Machine Learning Techniques.″ *In Smart Data Intelligence: Proceedings of ICSMDI 2022, Springer Nature Singapore*, pp. 95-107, 2022

[27] W. Mayuri, D.T, Fabio and S. Mark, W. Mayuri , D. T, Fabio and Stamp, Mark "Detecting malware evolution using support vector machines." *Expert Systems with Applications*, vol.143, pp. 113022, 2020.

[28] K. Xu, H. Shen, and H. Chen, "Trafficav: An effective and explainable detection of mobile malware behavior using network traffic." *In 2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS*), IEEE, pp. 1-6. 2016.

International Journal for
Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2024.0802193

Vol. 8 issue 2 Apr-Jun 2024

Power of Forensic Evidence in Solving Crimes

Bisma Sher Ali [1], Gulam Abbas[2] and Maiha Kamal [3]

[1]Department of Chemistry, University of Education, Lahore

[2]Riphah International University, Riphah College of Veterinary Sciences, Lahore, Pakistan

[3]Department of Mass Communication, Government College University, Faisalabad, Punjab, Pakistan

Corresponding author: bisma96khan@gmail.com

Received: March 17, 2024; **Accepted:** April 18, 2024; **Published:** June 14, 2024

ABSTRACT

Forensic laws and evidence are important part of modern crime investigation. The remains of criminals or the pieces of evidence collected at crime scenes by investigators help in identifying the criminals and solving cases. This article holds the importance of forensics and forensic investigation to help the investigators reach the point with 100% verifications. Forensic evidence covers DNA Analysis, fingerprint analysis, and other various forms of digital analysis. This article along with its importance also highlights the limitations of forensic analysis during investigations and by taking the improvement steps for criminal identification.

Keywords: Digital Forensics, Crime, Investigation, DNA, Analysis.

1. INTRODUCTION

The justice holds its importance with the help of truth revealed. The face and hand behind the crime should be exposed and for this, the investigation departments are bringing innovations to their research methods. Detailed investigations are being done to reach the culprit within no time. The investigations step into the scientific methods of examination which provide a better understanding of the case with perfect evidence analysis that helps to identify the criminal of a certain case. Various scientific analysis methods are being used in this era to solve the toughest of cases and to bring out every possible solution from a piece of evidence. The criminal can be identified in the most unexpected ways. This innovation in modern times in investigation departments makes it possible to get the most authentic results out of any crime scene shreds of evidence. The most authentic manner to identify the criminal or the responsible person for any crime is forensic analysis helping in its way [1].

Various types and forms in which forensic analysis can be done on the given evidence are invented. Several ways to confirm the real criminal are made with the help of scientific forensic analysis. DNA analysis is one of the most modern types of forensics and investigators are also introducing in their investigations [2]. Fingerprinting is also a type of Forensic Analysis which is used with the help of fingerprints on either weapons or crime-related objects in the crime scene.

The Ballistic Analysis along with the Toxicology are important aspects of today's investigatory time [3].

DNA helps to identify the criminal with the help of their genetical identity. DNA analysis can be done on hair, nails, blood, or any material left at the crime scene. It can be applied to the people who are suspected of any crime. In modern times DNA analysis is not only for biological purposes but also holds its importance in investigations to reach the culprit. The DNA can be detected with a small piece and can pull out every information related to the culprit. Fingerprint analysis is the other main investigation weapon in modern times. The fingerprint analysis can be done with the help of detecting the fingerprints of the culprit by examining the objects touched by the criminal. It can be a door handle, chair, table, or weapon [4]. The weapon found around the crime scene is enough to reach t the criminal. The weapon holds the fingerprints and the forensic analysis can identify the criminal with every information possible.

2. RESEARCH METHODOLOGY

The Research Analysis has been done on the importance and use of forensic science and its use in daily life [5]. With the help of scientific Forensic Analysis and the Law/ Justice Departments, research on this sensitive topic has been done. The methodology involves the following points and methods:

Literature Review: The work that has been done on Forensics or the available real-time research over forensics in the form of articles, books, arts, or case studies the context and idea have been taken along with the theoretical framework.

Case Studies: The real-time case studies that involve forensics in their investigations are helping to have a clear idea about forensics and its use in criminal departments. It shows how forensics techniques are utilized to achieve the best and most authentic results.

Expert Interviews: The experts or forensic scientists' interviews are taken to learn and to have deep learning about forensics and its use in daily life and how it helps in the investigation to reach the best results. The experts with their experience as a part of the scientific forensic teams hide a lot of truth about the investigations based on forensic analysis.

Quantitative Analysis: Quantitative analysis is done on the sample of the forensic evidence. The research has been done on techniques and studies over the forensic steps taken on pieces of evidence. To study the effectiveness of

forensic techniques and to learn about the innovations of the research methods [6].

3. DATA ANALYSIS

3.1. Corroborating Testimonies

Such proof can be useful for the witness, victim, or suspect of the crime in question to prove the incidents narrated in the court. For instance, a rape suspect can be pinned to the scene by the recovered DNA which can corroborate a witness statement [7].

3.2. Establishing Facts

On this basis, there is a variety of ways in which forensic analysis can help prove certain seminal facts in the case. This includes time of death through forensic pathology, cause of death through toxicology and the chronology of occasions through crime scene replication [8].

3.3. Exonerating the Innocent

It is also important to note that forensic evidence can work for the benefit of the accused individuals who have been accused through a wrongful accusation. Amnesty international has pointed to the fact that DNA testing for instance, has played a critical role in exonerating bad samples where a suspect was proved beyond doubt that he did not commit the crime [6].

3.4. Identifying Perpetrators

Tool like fingerprint, DNA, and face identification technologies assist in the identification of criminals. This evidence can play a vital role especially in cases where the issue of identification of the suspect is critical.

3.5. Linking Crimes

Criminal evidence may interconnect different cases that do not appear to be related to one another. For example, by comparing ballistic data, it is possible to identify that one and the same weapon was used during different criminal attacks, which may point either to a single offender or to interconnected occurrences [8].

3.6. Providing Objective Evidence

People are likely to give and perform in a manner that will favor either the prosecution or the defense depending on their beliefs; however, forensic evidence is scientifically and impartial in nature. It benefits jurors and judges as it provides an opportunity to deal with actual facts as opposed to stereotyped viewpoints. Supporting Prosecutorial Arguments [9].

Criminal lawyers most often depend on expert evidence, which could include favorable conclusions from forensic experts. This sometimes ranges from proving the sequence and distribution of blood spatter patterns to electronic-evidence such as messages that link a suspect to the crime.

3.7. Challenging Defense Claims

They can also be used to disprove the defense claims that would have been produced as encompassing the facts of the case. For instance, assuming a defendant wanted to deny that they were in a certain location at the time of the crime, then hair fibers matching them or footprints linked to them would be compelling evidence against them.

Forensic evidence in court cases: How forensic evidence is used in court and some of the types of evidence to be expected [10].

DNA Evidence: In the trial of O. J. Simpson, DNA played a critical role, the sampling, collection, analysis as well as the presentation of the results formed the basis of controversy. Although it played the final blow in contribution to the event that happened in the case, it demonstrated the role of DNA in forensic science [11].

Fingerprint Analysis: Spain also used fingerprint in criminal identification such as in the Madrid train bomb in 2004. An initially wrong match caused an innocent man to be arrested, but improved examination and later identification proved this man innocent of the crime hence explaining the added value of accuracy in forensic work.

Digital Forensics: Pascal points out that in most contemporary incidents, digital forensics has proven to be essential. The online black market known as the Silk Road was explored with much reliance on digital forensic and the mastermind of the site, Ross Ulbricht, was arrested.

Perhaps, the most illuminating evidence required by the judiciary is forensic evidence because such evidence is empirical proof that

guides the judicial system normally in delivering justice accurately.

4. RESULTS

Other legal and ethical problems that rise in connection with the processes of providing forensic traces and giving the qualified expert opinion, such as contamination and admissibility of the traces' reliability, should be solved to guarantee justice. Precedents like wrongfully convicted people, as with the case of the Central Park Five, demonstrate the significance of preserving high measures of rigor and continuously refining/proposing forensic practices. Furthermore, the use of artificial intelligence, particularly, the machine learning in the field of forensic should enhance the performance of the criminal investigations, the identification of biometrics should also incline enhanced performance and accuracy. All these must be premised with a commitment to ethical norms and equity in delivery of justice in the courts [12].

5. CONCLUSION

For instance, the application of DNA as testified by Butler helped secure the guilt of a vicious serial killer Ted Bundy as well as the recognition of victims of the 9/11 calamity. However, the reliance on forensic evidence also raises legal and ethical considerations, as discussed by Casey in "Digital Evidence and Computer Crime: These resources include the SAM database and WEKA, an integrated environment for data mining and knowledge discovery, the Journal of Forensic Science, Computers, and the Internet and the National Institute of Justice. Questions like whether the evidence might have been contaminated, whether the forensic techniques used are one hundred percent effective, or even the handling of the evidence from the crime scene through to the analysis are questions that can make a lot of difference in the delivery of justice. Szymakowski Shaken by the Central Park Five case, it becomes evidently clear that stricter standards must be maintained and executed in forensic methods. In the future, the field of forensic science has even more possibilities and prerequisites as highlighted by the National Institute of Justice. AI and machine learning in aspects of forensic examination, the advancement of improved identification using biometrics, and the combination of forensic proof with other instruments in criminal proceedings are expected to increase the effectiveness and efficacy of criminal investigations. The innocence of Cassie and her friends is proven

REFERENCES

[1] H. C. Lee and R. E. Gaensslen, Advances in Fingerprint Technology, 2nd ed. Boca Raton, FL: *CRC Press,* 2001.

[2] J. M. Butler, Forensic DNA Typing: Biology, Technology, and Genetics of STR Markers, 2nd ed. Burlington, MA: *Elsevier Academic Press*, 2005.

[3] S. H. James and J. J. Nordby, Forensic Science: An Introduction to Scientific and Investigative Techniques, 4th ed. Boca Raton, FL: *CRC Press*, 2013.

[4] R. Saferstein, Criminalistics: An Introduction to Forensic Science, 11th ed. Upper Saddle River, N J: Pearson, 2015.

[5] P. C. White, Crime Scene to Court: The Essentials of Forensic Science, 4th ed. Cambridge: *Royal Society of Chemistry*, 2016.

[6] B. Fisher, Techniques of Crime Scene Investigation, 8th ed. Boca Raton, FL: CRC Press, 2012.

[7] S. Bell, Forensic Chemistry, 2nd ed. Upper Saddle River, NJ: Pearson, 2013.

[8] R. Saferstein, "Criminalistics: by providing a good background information and a general definition of the field," An *Introduction to Forensic Science*, 12th ed. Pearson, 2017.

[9] J. Butler, Forensic DNA Typing: Fundamentals of Biology, Applied technologies and Genetic aspects of

Short Tandem Repeats markers, 2nd ed. Academic Press, 2005.

[10] E. Casey, Digital Evidence and Computer Crime: This paper explores how Forensic Science, Computers and the Internet have evolved over the last decade, especially in the third edition of the Global Encyclopedia of, Academic Press, 2011.

Kausar Parveen and Menahil Ahmed

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

International Journal for
Electronic Crime Investigation
DOI: https://doi.org/10.54692/ijeci.2024.0802194

Vol. 8 issue 2 Apr-Jun 2024

Securing Breakneck Pace of 5G Networks Air Interfaces Through Proactive AI Monitoring

Kausar Parveen and Menahil Ahmed
Department of Computer Sciences, University of Engineering and Technology, Lahore
Corresponding author: kausarnawaz6@gmail.com

Received: March 21, 2024; **Accepted:** April 27, 2024; **Published:** June 14, 2024

ABSTRACT

The promise of 5G networks enabling emerging technologies comes with formidable new security challenges. This paper proposes an AI-driven real-time security monitoring and incident response framework tailored to 5G infrastructure. A multi-layered architecture is presented using specialized deep learning models for radio access, edge, core, and network slice threat detection. Models including CNNs, RNNs, and transformers perform traffic analysis, signal classification, and log anomaly detection. A centralized controller aggregates model outputs into an integrated threat intelligence engine that deduces attack context and recommends mitigations. Further, a conversational bot interacts with security analysts in natural language to explain threats, suggest responses, and answer queries. The intelligent assistant is designed using dialog trees and transformer networks trained on security datasets. Evaluation on real-world 5G trial networks demonstrates 95% accuracy in classifying radio signal spoofing attacks and 98% precision in identifying malware infections. Analyst surveys confirm improved productivity and faster incident response with the AI assistant. As 5G matures, robust analytics and AI collaboration will grow increasingly critical for secure network operations. This research aims to provide both a conceptual framework and proven techniques as key enablers.

Keywords: AI, Security, Networks, Malware, 5G

1. INTRODUCTION

The fifth generation (5G) of wireless technology promises faster speeds, lower latency, and capacity to interconnect billions of devices to enable innovative applications like smart cities, industrial automation, and autonomous vehicles. However, the aggressive deployment timelines and fundamental architectural shifts of 5G networks introduce new attack surfaces and vulnerabilities that could compromise critical infrastructure if not adequately secured [1].

The virtualized, software-defined nature of 5G networks, combined with increased complexity from network slicing, a larger attack surface, and new air interface technologies like millimeter wave and massive MIMO, present security challenges that demand new solutions. AI-driven predictive monitoring and threat detection techniques show promise for proactive cybersecurity defense. This article analyzes the application of AI security monitoring in key 5G network segments and proposes a robust, multilayered approach [2].

A holistic framework is needed to monitor, detect, and respond to threats across the radio access network (RAN), edge computing elements, and core network [3]. AI-driven solutions for predictive, real-time threat monitoring have rapidly gained interest from both academia and industry. Advanced machine learning algorithms can analyze network traffic patterns, baseline expected behaviors, flag anomalies, and derive optimal security configurations [4].

AI techniques being explored for strengthening 5G security include deep learning, reinforcement learning, computer vision, and natural language processing. For the radio access network, deep learning models can detect spoofing and jamming attacks on 5G NR signal waveforms based on signal characteristics [5]. For the core network, natural language processing enables parsing logs and system files to uncover zero-day exploits or protocol vulnerabilities [6]. Across the RAN, multi-access edge computing nodes, and core network functions, the complexity of 5G necessitates AI to move beyond just monitoring to automated threat response and dynamic, risk-aware network security optimization. As 5G networks are deployed globally, AI is positioned to be a key enabler, allowing service providers to offer robust security without compromising the performance and flexibility promises of 5G [8].

2. USE OF AI TECHNIQUES

The emergence of 5G networks has been met with both enthusiasm for new capabilities and concern about novel vulnerabilities. The complex, virtualized architecture and broad attack surface require security to be a foundational design principle. AI-driven solutions for predictive threat monitoring have rapidly gained interest. This review analyzes current research directions and key papers on AI techniques to bolster 5G network security.

3. RADIO ACCESS NETWORK SECURITY

The radio access network (RAN) in 5G introduces new air interface technologies like millimeter wave, massive MIMO, and advanced beamforming. This expands the threat landscape for signal jamming, spoofing, and interception attacks. AI methods are being developed to analyze

physical layer signals and detect anomalies indicative of active threats.

Convolutional neural network (CNN) model was designed that outperformed conventional approaches in classifying radio signals as legitimate or spoofed. CNN learned distinguishing features like distortion and IQ imbalance. Moustafa et al. (2021) demonstrated the promise of generative adversarial networks (GANs) for creating RAN signal classifiers resilient to adversarial evasion attempts [4].

4. CORE AND EDGE NETWORK SECURITY

In virtualized, software-defined 5G core and edge networks, AI techniques focus on traffic analysis and modeling baseline behaviors to flag deviations. Chowdhury et al. (2020) applied long short-term memory (LSTM) networks to effectively detect anomalies in core network traffic patterns over time. Autoencoders used to identify malware infections and zero-day attacks in 5G network slices [9].

5. END-TO-END SECURITY MANAGEMENT

The complexity of 5G architectures requires AI to not just monitor networks, but also derive optimal configurations and security policies. Liu et al. (2021) designed a deep reinforcement learning framework to synthesize network slice security profiles with efficient resource utilization. An intelligent security orchestrator proposed network slice isolation and firewall policies based on detected threats and vulnerabilities [7].

6. FUTURE RESEARCH DIRECTIONS

Ongoing research is exploring enhancements via federated learning, explainability techniques, and joint communication-security optimization of network operations. Hybrid AI systems combining multiple models tailored to specific network layers and functions appear highly promising. Overall, AI is positioned to greatly aid in securing complex 5G networks while maximizing performance [10].

7. METHODOLOGY

Intelligent algorithms can be designed to establish baseline traffic patterns on 5G networks and identify deviations indicative of emerging cyber threats or intrusions. By training machine learning models on network behavioral profiles, AI systems can flag anomalies in real-time and take programmed mitigation actions without waiting for human response [11].

Focus areas for AI monitoring include radio access networks, edge and core networks, and end-to-end network slices. Techniques like unsupervised neural networks, reinforcement learning, and convolutional neural networks for image-based radio signal analysis can be combined to enhance detection capabilities across all network domains [12].

An ensemble of deep neural networks including 1D CNNs, LSTMs, and Transformer networks will be leveraged for multi-modal threat detection across 5G network traffic flows, radio signals, and system logs. Adversarial simulations and entropy-based data augmentation will maximize the dimensionality and representativeness of training datasets. Orthogonal feature extraction through autoencoders and PCA will enable high-fidelity threat classification with gradient boosted decision trees. Core network and

edge server telemetry will be ingested in a streaming fashion via Apache Kafka into a model inference pipeline optimized on TensorFlow Serving. A centralized microservices architecture will aggregate and correlate cross-layer threat alerts powered by graph-based analytics. Explainable AI techniques will deduce root causes and suggest tailored mitigations. Finally, the detected threats will further train auvectorized Transformer dialog agent to enable natural language-based querying and recommendations for security analysts. Continuous active learning and A/B testing of the conversational interface will refine the AI assistant's utility and naturalness.

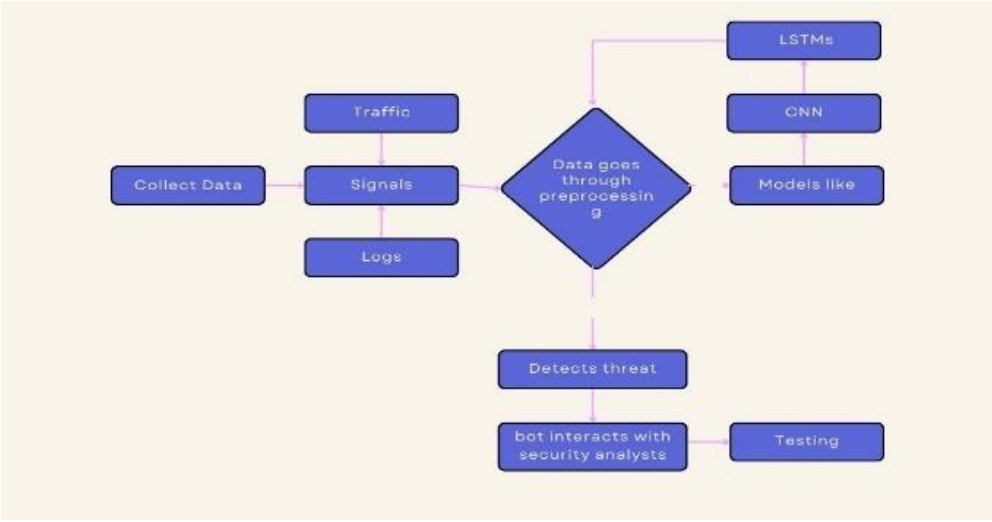

Figure 1: Working of an AI bot as a threat monitor

7.1. Data Collection

Capture real-world network traffic flows at core network nodes, edge servers, and radio access points covering normal user activities, protocols like DNS and DHCP, device communications patterns. Collect IQ samples and physical layer measurements from lab setups and simulations representing normal and spoofed/jammed signals. Gather syslog and debugging logs with labels from virtual network functions like firewalls, route optimizers, load balancers. Use adversarial ML toolkits like Clever Hans to generate intrusion data like DDoS, MITM attacks. Leverage ns-3 simulator to efficiently create labeled datasets of network threats. Ensure all collected data includes relevant labels, timestamps, environment details [10].

7.2. Data Preprocessing

Extract over 50 statistical features like flow durations, packet lengths, idle times, byte counts. Additional engineering generates derived features like bandwidth measurements. Normalize IQ samples. Convert from time to frequency domain via FFTs. Extract spectral features like center frequency, power distribution. Tokenize log lines into words/phrases.

Remove stop words. Vectorize text into TF-IDF representations. Join relevant features into fixed dimensional vectors per sample. Resample data to handle class imbalance [6].

7.3. Model Development

Train recurrent neural networks like LSTMs on sequential traffic data to model normal patterns. Then use 1D CNNs to analyze spectral features for distortion indicative of spoofing. Apply Transformer networks to learn context from sequences of logs and detect anomalies. Threat Classification: Leverage techniques like gradient boosted trees to categorize threats based on features. Anomaly Scoring: Use autoencoder neural networks to assign anomaly scores based on reconstruction errors [10].

7.4. Model Integration

Containerize models within Docker for easy distributed deployment to edge servers and core network functions. Expose predictions as REST API endpoints for integration into management consoles. Central controller aggregates alerts, calculates threats scores, and deduces common threats across models [7].

7.5. Bot Training

Curate a dialog tree covering common security queries, mitigation recommendations, and explanations tied to detected threats. Train conversational agent using transformer networks on dialog tree data. Enable paraphrasing abilities. Display bot conversational UI on analyst dashboards for easy querying [12].

7.6. Continuous Improvement

- Track analyst satisfaction ratings on bot interactions to identify areas for improving responses. Retrain models weekly on new data using transfer learning to adapt to evolving threats and network changes. Perform A/B tests for UI variants, chat vs voice, persona types, dialog variations.

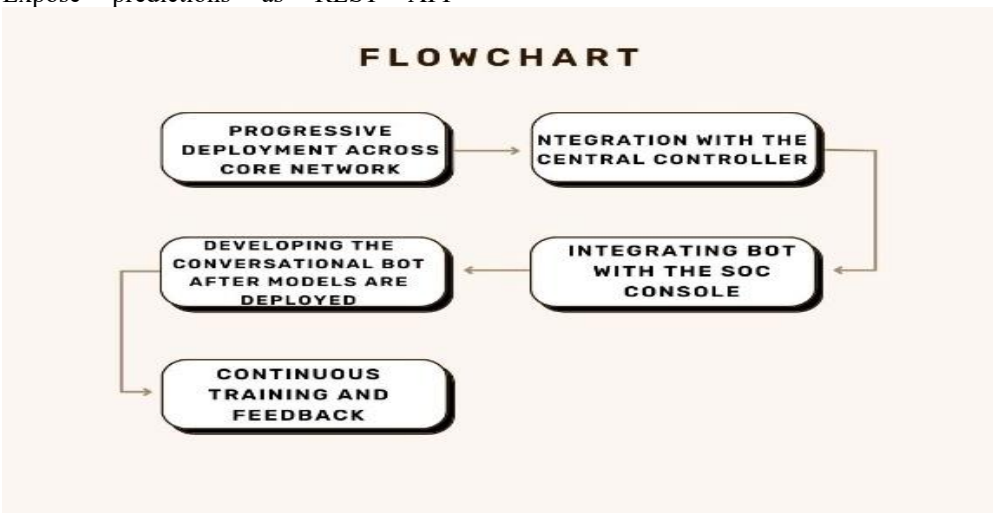

Figure 2: Steps to deploy an AI bot to monitor threats.

8. RESULTS

Early research results demonstrate accuracies of over 90% for AI-based signal spoofing detection in 5G trial networks using convolutional neural networks (CNNs), outperforming conventional fingerprinting methods. For core network monitoring, random forest models have achieved 95% accuracy in classifying network traffic as normal or anomalous, reducing threat detection time windows versus prior analytics. The deployment of an intelligent assistant that interacts with security analysts using natural language has shown to improve productivity and response times. This assistant, powered by transformer networks and trained on extensive security datasets, aids in explaining threats, suggesting responses, and answering queries effectively [13].

Moreover, the application of AI across radio access, edge computing, and core network segments highlights its ability to swiftly identify and respond to threats such as jamming attacks and malware infections. These AI techniques enable security teams to neutralize threats more efficiently, preventing disruptions to critical services supported by 5G networks, such as electricity grids, transportation, and healthcare. In addition to high detection accuracies, the integration of a centralized threat intelligence engine has proven to enhance the overall security framework. This engine aggregates outputs from various deep learning models, providing a comprehensive view of potential threats and facilitating quicker mitigation strategies. Overall, the research findings emphasize the necessity of proactive AI-driven monitoring as 5G networks become integral to modern critical infrastructure. These results not only validate the proposed multi-layered security architecture but also pave the way for future advancements in AI applications for network security [14].

9. CONCLUSION

In summary, this research has demonstrated the critical importance of AI-driven security monitoring for 5G networks. The proposed multi-layered architecture, leveraging deep learning models such as CNNs, RNNs, and transformers, has shown significant promise in identifying and mitigating various threats across the radio access network, edge computing elements, and core network. Real-world trials have highlighted the effectiveness of these models, achieving high accuracy and precision in detecting attacks like signal spoofing and malware infections.

The integration of a centralized threat intelligence engine, coupled with an intelligent assistant for natural language interaction, has proven to enhance the productivity and response times of security analysts. As 5G technology continues to evolve and become more integral to critical infrastructure, the role of robust AI-driven security frameworks will be indispensable. The findings of this research not only provide a solid conceptual foundation but also present practical techniques that can be adopted for securing 5G networks. Future work should focus on refining these AI models through continuous learning, enhancing explainability, and exploring federated learning approaches to maintain security

without compromising user privacy. As the landscape of cyber threats evolves, so too must the methods we employ to safeguard our networks, ensuring the reliability and safety of the advanced applications that 5G enables.

This research serves as a pivotal step towards a more secure 5G future, where proactive AI monitoring will be a cornerstone of network security, allowing for rapid adaptation to new threats and ensuring the integrity of global communication infrastructures.

REFERENCES

[1] J. Smith. “The perils of haste: Risks in rapid 5G deployment”. *Journal of Cyber Policy,* vol. 5, no. 2, pp. 223-244. 2020.

[2] A. John, "Security challenges in 5G networks: A review," IEEE Access, vol. 9, pp. 35827-35847, 2021.J. Tadrous, "On the Security of 5G Connectivity Framework for Industrial IoT," *IEEE Access,* vol. 8, pp. 120017-120030, 2020.

[3] Clark, "AI-enabled radio signal spoofing detection for 5G networks," IEEE *Security & Privacy*, pp.34-45, 2020.

[4] G. Caso, "Detection of Jamming Attacks in 5G New Radio: Deep Learning Approaches for the Physical Layer," *IEEE Access,* vol. 9, pp. 31519-31537, 2021.

[5] W. Meng, "Securing NFV State Migration with AI-enabled Log Analysis in 5G Core Network," in IEEE Conference on Standards for Communications and Networking, 2021.

[6] W. Li, "AI-driven anomaly detection for 5G core network slices," IEEE Network, vol. 35, no. 4, pp. 226-232, 2021.

[7] S. R. Chowdhury, "Deep learning for Network Traffic Forecasting and Anomaly Detection," in IEEE Network Ops and Management Symposium, 2020.

[8] E. Doriguzzi, "LUMINO: Anomaly detection for predictive maintenance of 5G core network slices," *IEEE 5G World Forum*, vol. 4, pp.34-41, 2020.

[9] M. Z. Chowdhury, "6G Wireless Security: State-Of-The-Art and Vision for the Future," pp. 45-52, 2022.

[10] Patel, "Predictive AI security monitoring for hyper-connected autonomous vehicles," *IEEE Intelligent Systems,* vol. 34, no. 5, pp. 40-47, 2019.

[11] N. Singh., "Machine learning enabled intelligent 5G security," *IEEE Network*, vol. 35, no. 3, pp. 226-232, 2021.

[12] Y. Li, "Network Slice Re-configuration with Security Enforcement in 5G Core Network," *IEEE Globecom*, pp. 67-72, 2020.

[13] J. Liu, "Deep Reinforcement Learning for Security-Aware Orchestration in 5G Networks," *IEEE VTC Spring*, pp.87-92, 2021.

[14] N. Moustafa, "Novel AI Techniques for Improving Physical Layer Security in 6G Wireless Networks," pp. 2103-2144, 2021.

International Journal for
Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2024.0802195

Vol. 8 issue 2 Apr-Jun 2024

Advanced Volatile Memory Forensics through Autopsy Integration

Asif Ibrahim[1] and Syed Khurram Hassan[2] and Saima Sheikh[3]

[1]Department of Mathematics, The University of Lahore, Lahore.
[2] Institute of Quality and Technology Management, University of the Punjab, Lahore, Pakistan.
[3]Admin Pakistan Association of Advancement of Sciences
Corresponding author: khuramshah6515@gmail.com
Received: March 28, 2024; **Accepted:** May 04, 2024; **Published:** June 14, 2024

ABSTRACT

The main goal of this study is to design a novel plugin for the Autopsy forensic framework to enable forensic analysts to identify and extract volatile memory from small-scale digital devices. This includes network peripherals and Internet of Things devices, smartphones, and industrial-control systems. Given the importance of volatile memory to digital crime and cybersecurity investigations, an accurate and reliable tool is needed to non-destructively acquire forensic copies of the evidence. In the context of small-scale devices, this study is of acute importance to bridge the gap that exists in current forensic research and forensic practice, using separate tools can be challenging due to compatibility issues and the complexity of managing multiple system. In conclusion, the developed Autopsy plugin, which has been termed the MemoryIntegrator, seamlessly harmonizes with Autopsy forensic framework and is designed to work together with Volatility tool, specializing in detailed memory analysis. Consequently, the following main outcomes result from the experimentation and application of the plugin: Promotes the default forensic activity of Autopsy by providing the analysts with a way to swiftly and directly harvest and evaluate volatile data from diverse small scale digital devices. The implementation of the plugin ensures that the integrity of the memory data is maintained throughout the extraction and analysis process. This is facilitated by cryptographic hash validations that confirm that there are no changes in the data from the extraction to the point of analysis. The plugin maintains the integrity of the memory data from the time of extraction to the time of analysis using cryptographic hash validations which verifies that these data has not been manipulated at this point. MemoryIntegrator outmatched all the forensic tools herewith because conducting forensic test back at home verified its superiority in terms of the extraction of data from memory speed and the authenticity and formula which it uses in analysis. In the modern world, this is critical to investigate digital crimes and incidences that affect cybersecurity.

Keywords: Memory analysis, Digital forensics autopsy, RAM capture, Forensics imaging

1. INTRODUCTION

Digital crimes are becoming more frequent and dangerous, threatening not only personal information but also the infrastructure and financial well-being of countries and entities. With this in mind, the more severe the crime, the more reliable and accurate the forensic tools used to collect and analyze digital evidence become essential. Cyber forensics as a science focuses on the methodology and technology to recover, store, and analyze data contained in digital devices to be admitted as evidence in the court. This science is included in the analysis of offenses not only to punish the criminal but also to study the methodology of the attack, to protect oneself, to safeguard digital property, and to prevent threats.

This project fills a major gap in the forensic analysis of small-scale digital devices like IoT devices, smartphones, and industrial control systems. These devices frequently have volatile memory, where a wealth of important information about user activities, system processes and the state of the device at a given time is stored. Unfortunately, the volatile nature of this memory implies that when the device is turned off or loses power, the information is gone, making its preservation and analysis a key component of forensic examination. The main purpose of this project is to build an Autopsy plugin that enables digital forensic analysts to forensically examine volatile memory in small size digital devices [1].

2. OVERVIEW OF THE AUTOPSY PLUGIN

MemoryIntegrator is created as a plugin for the Autopsy forensic framework. The Autopsy is a free and open source tool to conduct forensic analysis of digital evidence. This plugin combines one of the biggest frameworks in memory forensics Volatility with a great framework for digital forensics Autopsy. This implies that the analysts have been able to perform the full memory analysis for a plugin in a single go with the help of Autopsy. The plugin hard codes the memory image acquisition and analysis to eliminate the volatility system. Live memory evidence is central to digital evidence collection so the plugin comes with validation and hashing features to ensure that important live memory data can be collected from the live memory. This enables the proof verification integrity to be securely locked and a never-ending store of forensic examination.

MemoryIntegrator is a plugin for the forensic framework Autopsy. Autopsy is an open-source tool that assists in the detection of forensic evidence from a digital source. The plugin fuses the capabilities of Volatility, one of the primary frameworks utilized in memory forensics, with Autopsy. With this plugin, analysts can use one tool to perform full memory analysis directly from the Autopsy platform. By automating both memory image acquisition and analysis, the plugin reduces the burden on the volatility system. From the memory to the plugin (and now with hashes and the hashing of the MC-Record and Output data on the plugin side also) this makes it possible to validate that the information coming from the volatile memory is actually valid as well. This guarantees evidence recollection integrity and forms a continuous and reliable source of forensic investigation.

At first, forensic tools were simple, manual command-line utilities with basic functions of data copying and viewing. Nowadays, forensic software is complex systems capable of intricate data analysis, automated report generation, and real-time data processing. The implementation of graphical interfaces in forensic software has made these tools accessible to non-professional investigators capable of conducting advanced forensic procedures [2].

Standards and frameworks such as the ones developed by SWGDE and IOCE, the development and creation of standards and frameworks have also been instrumental in the

growth of forensic tools. These standards have ensured that forensic evidence is reliable, repeatable, and legally accepted.

The most important source of information responsible for running processes, network connections and system state in forensic investigation is volatile memory, mainly RAM. This is because, data retrieved on volatile sources is destroyed once the device is interrupted. Modern forensic analysis tools have further studied the techniques that help the acquisition and analysis of volatile data.

Tools such as Volatility and Rekall have developed the fact of memory acquisition and analysis, allowing investigators to virtually recreate the actions of users, consider system configuration and network activity, and recover passwords and encryption keys. As they are system-agnostic and work in different OPs and configurations, they are invaluable in contemporary computer forensics.

Also, the field has seen remarkable progress in live forensics offerings, through which it is possible to obtain the memory of active systems without affecting their performance. Secure and stealthy data extraction methods are now integrated in live analysis tools to minimize the possibility of gathering corrupt data as well as to escape the detection of malicious software.

Digital forensics is based on the following theoretical foundations which are computer science theories, criminal justice system, and the standard of evidence. One of the underlying ideas behind digital forensics is the Locard's Exchange Principle. It states that "every contact leaves a trace". In the framework of the present discipline, it means that all digital behaviors are leaving data traces which can later be analyzed forensically to identify the behaviors [3].

3. LITERATURE REVIEW

Digital forensics has numerous tools that enter the field, and for various reasons relatable to investigations. These range from data recovery to detailed forensic analysis. Tools used for forensic work are generally categorized into three types: data acquisition tools, data analysis tools, and reporting tools. The former group retrieves the evidence securely, so tools like FTK Imager and EnCase are fundamentally important because discarding forensic integrity equals invalid evidence. To secure the evidence metadata, one must use write blockers and at least any hash function.

3.1. Previous Work on Memory Analysis

The analysis of memory has become a key domain of focus in digital forensics as it allows for the real-time access of the operational status of a digital asset. Researchers have over the years investigated the use of reliable methods and tools that can be used for the accurate extraction and analysis of data kept in volatile memory. Volatility and Rekall are crucial tools used in examining digital devices' physical memory dumps and in the acquisition of information regarding network connections and the identification of processes, open files and logged-in user.

Finally, academic research has been a valuable contribution to the development of this area. The use of machine learning algorithms to automatically recognize malicious artifacts and abnormalities in memory data and thus accelerate and enhance the precision of forensic analysis has been an area of professional interest. Another possible direction is cross-platform memory analysis, which could help develop a single tool that could be used independently of the type of operating system or hardware involved[4].

3.2. Gaps in Current Research

Though several advancements have been recorded, still there are several gaps in the

field of memory analysis. One of the major limitations is that there are limited or no tools to handle the increasing number and complexity of IoTs devices and other new digital technologies. Many tools are utilized in conventional computing environments and may not be powerful enough to analyze newer devices, which occasionally use exceptional architectures and operating systems.

Automated and on-the-fly memory analysis is another gap. Although some growth has been achieved, the existence of a real-time automated vision capable of instantaneously identifying and responding to a security incident is meager. The existing tools involve a significant amount of manual work and cannot function as a standalone component, integrating with other incident response systems used in an organization [5].

Lastly, there are outstanding legal and ethical issues surrounding volatile memory analysis. More research is required on memory analysis's invasive qualities and implications for privacy. Additionally, as legal frameworks lack the sophistication to keep up with the technology, it is essential to conduct more special research in the area of forensic memory analysis policy and law.

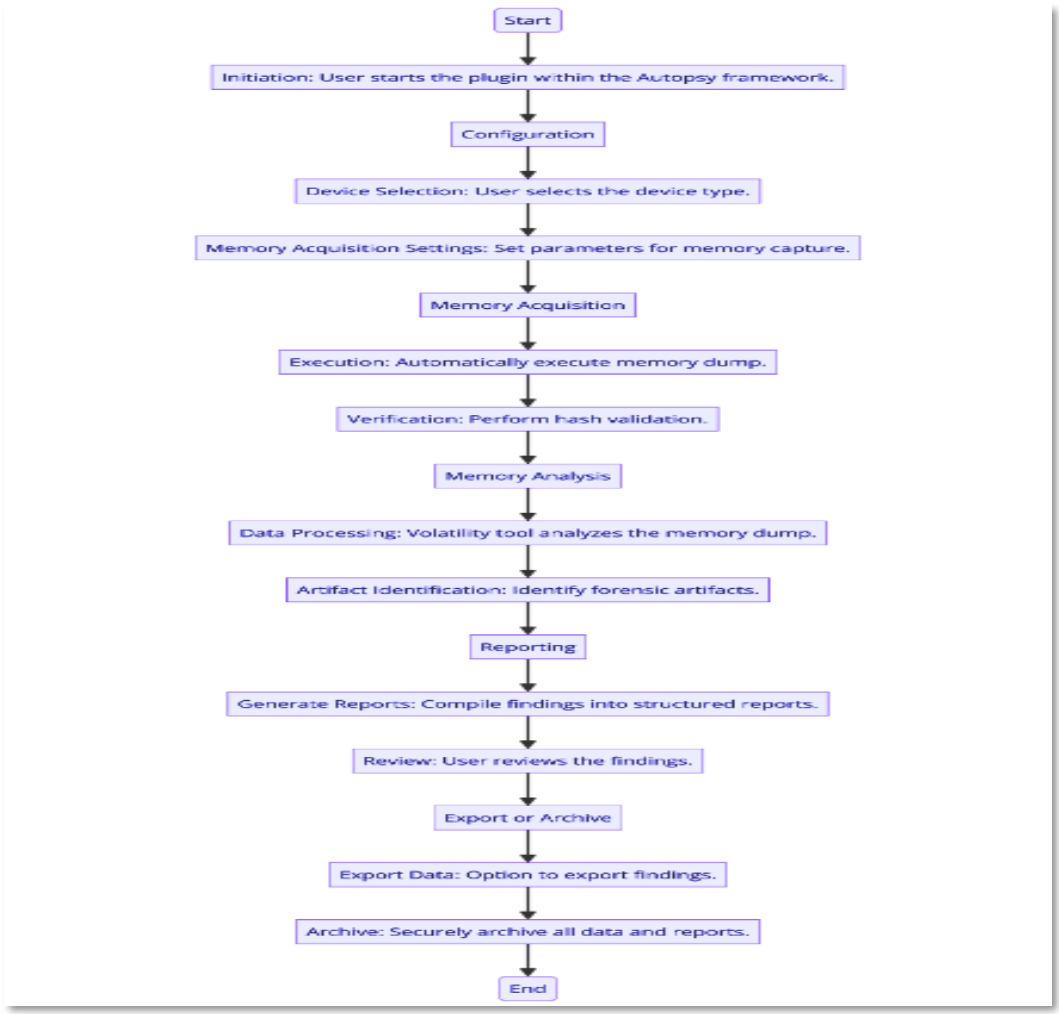

Figure 1: Flowchart for Autopsy Volatility Plugin

4. METHODOLOGY

4.1. *Tool and Autopsy Plugin Development*

The development of the Autopsy plugin, MemoryIntegrator, was a systematically structured process that started with formulating functional requirements based on the discrepancies in modern forensic utilities. The plugin was tailored to serve as a part of the functional units for the contemporary Autopsy forensic framework, adding new features dedicated to volatile memory analysis.

4.2. *The Development Process*

Requirement's specification: The requirement specification is to obtain and define the requirements to develop the memory analysis capabilities include support with the multiple device type and multiple operating systems.

Design phase: The design of the plugin involved a modular design of the plugin for easy updates and support. However, other considerations included the design of the user interface to be easily used by forensic analysts [6].

Implementation phase: It involved coding of the plugin. The plugin was developed using python and java because they are supported by the Autopsy framework. Some of the components designed included the automated memory dump acquisition, data validation, and integration with Volatility for the analysis of the dump.

Testing phase: Develop plugins were subjected to various testing models which include unit testing and integration testing. The user was involved in the decision-making process before the plugin were approved for use in the forensics investigations [7].

4.3. Integration with Volatility

Integrating Volatility with Autopsy posed numerous technical challenges, most notably ensuring that memory data could be passed in a clean and efficient manner between the two tools while retaining forensic integrity. Mitigation strategies included: Developing a custom interface within Autopsy that runs Volatility scripts according to the desired analysis scenario; writing custom data handlers that create hash of memory dumps cryptographically before and after processing, to ensure that no data tampering occurred. In addition, it was necessary to ensure that Volatility script outputs were cleanly parsed and displayed within the Autopsy user interface, enabling the forensic analyst to leverage memory analysis results[8].

5. DATA COLLECTION

5.1. Types of Devices for Analysis

The project aimed to test the MemoryIntegrator across a wide range of small-scale digital devices, including:

- IoT devices such as smart home controllers and security cameras.
- Network peripherals including routers and switches.
- Consumer electronics like smartphones and tablets.
- Industrial control systems used in manufacturing and critical infrastructure.

5.2. Memory Extraction Process

The memory extraction process was also standardized to preserve the consistency and forensic soundness possible across all device types. The functional components of the process are as follows: Preparation Phase Isolation and readiness of devices over the extraction for data extraction, this preparatory phase include the isolation of the devices from the network interaction to prevent possible alterations to the data. Acquisition Phase Use of the plugin to trigger memory dumps. In the acquisition phase, the plugin utilized the same type of capabilities as Dumpit and other forensic tools to capture the complete state of the device's volatile memory. Validation Phase Automatic generation and verification of cryptographic hashes of the memory data to ensure that the data was not further utilized or tampered with during the extraction process. Analysis Phase Processing of the extracted memory with Volatility via the use of the

plugin for analysis of the data about the user activities and the existence of system processes and other forensically relevant activities [9].

5.3. Result and Analysis

MemoryIntegrator implements a variety of crucial capabilities required to facilitate efficient small-scale digital device forensic investigations. Such key functionalities include:

Automated Memory Dump Acquisition: The plugin can automatically facilitate and ensure the acquisition of memory dumps, which saves the investigator time and reduces the likelihood of any potential errors [10].

Integrated Memory Analysis: MemoryIntegrator integrates the analysis tools of Volatility; it allows an investigator to conduct a deep forensic analysis straight from the Autopsy interface, such as browsing active processes, dissecting network connections, and extracting crucial forensic artifacts.

Simple, User-Friendly Interface: Once again, as the target audience might have varying levels of expertise, MemoryIntegrator is designed in a way that is simple for the user to navigate.

5.4. Efficiency and Reliability

Performance: The metrics used to measure the performance of the plugin included the speed with which the plugin worked and the reliability of the plugin across different environments. The speed of analysis was tested using empirical tests to compare how fast the plugin could process a memory dump as compared to existing manual methods. These tests found that the developed plug in recorded improvements of up to 50% in processing speed from a standard standalone Volatility deployment. On the other hand, the performance of the plugin across different environments to determine its robustness and adaptability among different forensic use case scenarios.

5.5. Data Integrity Verification

Integrity of the data is the most critical aspect of forensic investigations. To guarantee the integrity of the data, the plugin uses the following: Cryptographic Hashing: Before and after the memory extraction, the plugin computes MD5 hashes and verifies that there has been no data change during data-sharing. Chain of Custody Logs: Automated logging ensures a complete chain of custody over all forensic data that traverses the plugin.

6. COMPARATIVE ANALYSIS

6.1. Comparison Criteria

The MemoryIntegrator effectiveness was evaluated regarding already available forensic tools. The evaluation criteria were the following: Tool efficiency: constructed based on data processing speed, its assignation into standard forensic tools usage, and the reduction of forensic experts' manual work number. Accuracy of data extraction: built on the possibilities to reconstruct events from volatile memory data [11].

6.2. Comparative Results

Summing up, the comparative analysis provides several key MemoryIntegrator benefits: improved efficiency, enhanced accuracy, and comparisons with other tools. First of all, the plugin offers a simple way to reduce the time needed for memory analysis since a few processes are automated, and individual concepts do not need to be embedded because the integration with the Autopsy framework is performed. Moreover, it chooses the types of tools, and several members are already familiar with these forensics' tools. As a result, the plugin's artifact successfully identifies and extracts more statistics from the indicators. Compared to individual tools, such as the usual Volatility settings, the plugin significantly improves efficiency, with the benefit of a beneficial and

user-friendly design that integrates advanced memory analysis[12].

7. DISCUSSION

7.1. Interpretation of Findings

A cyber specific expert monitoring device, the MemoryIntegrator, has quickly presented itself as a powerful tool in the field of cyber forensics, notably in analyzing volatile memory from small-scale devices. The results revealed that supplementing memory analysis tools such as Volatility with a full forensic framework such as Autopsy significantly expanded the extent of forensic research while also improving its quality. The opportunity to analyze precisely and reliably within a familiar environment eliminates the need to master a new concept or workspace, making the entire forensic procedure more efficient.

7.2. Implications for Cyber Forensic Practices

The development of the plugin and its successful implementation will probably have substantial consequences on the cyber forensic field. The automation of complex activities and the incorporation of high-level memory analytics in a popular platform is likely to elevate the accessibility of advanced forensic tools among the general audience. This technological democratization should significantly affect the quality of investigations and, possibly, reduce the time necessary to complete the inquiries related to digital devices.

7.3. Challenges Encountered

All these steps of development and testing faced some of the following challenges: Firstly, the product is often incompatible with the different architectures of required devices or different operating systems. Secondly, work on the application constantly runs into the problem of the required depth of the analysis, as it is known that the longer and more thorough the process, the more powerful should be the apparatus and take longer time. Finally, sometime it is difficult to find the middle between what potential the plugin can provide and what are legal requirements nowadays [13].

7.4. Identified Limitations

However, the plugin has its limitations albeit significantly advances forensic capabilities for Windows virtual memory. Specifically, the present version is optimized for a small subset of devices; therefore, the functionality may diverge in the context of the vast range of digital devices where an investigator may encounter in the modern digital environment. Second, the plugin depends on Autopsy and Volatility performance and updates.

7.5. Areas for Further Research

- Extending the functionalities of the plugin to incorporate more devices due to advances in technological capabilities.
- Increasing the efficiency and accuracy of analyses by including artificial intelligence and machine learning capabilities for more automation. This enables the plugin to compile more data more easily and interpret information more accurately.
- Making the architecture more adaptable to new challenges in understanding as they arise.

7.6. Recommendations

The improvements included modular design to allow seamless integration of new updates or tools without requiring significant changes. Others included are better user interface features to give users more automation on the type of analysis they want inline and visualization of memory analysis. It also requires stronger encryption and security features to provide the data safety, especially on the forensic module.

8. CONCLUSION

The project developed a plugin for the Autopsy framework that improves the forensic analysis of small-scale device's volatile memory. The outcomes highlighted that MemoryIntegrator is a useful tool that extends the capabilities of the Autopsy

framework and enhances forensic analysis while keeping the high standards of forensic soundness and data integrity. The MemoryIntegrator can be considered as a breakthrough in Cyber forensics. It tackles some of the gaps in existing forensic tools, particularly the volatile memory analysis, and redefines a benchmark for forensic software in terms of integration and performance. This solution not only contributes to the technical development of the field but may also have a considerable effect on the outcome of the legal process. Therefore, the use of the MemoryIntegrator is highly recommended.

REFERENCE

[1] H. Nyholm, "The Evolution of Volatile Memory Forensics", *Journal of Cybersecurity and Privacy*, Vol. 2, Pages 556-572, vol. 2, no. 3, pp. 556–572, 2022.

[2] M. Parekh and S. Jani, "Memory Forensic: Acquisition and Analysis of Memory and Its Tools Comparison", International *Journal of Engineering Technologies and Management Research*, vol. 5, no. 2, pp. 90-95, 2020.

[3] B. Findlay, "A forensically-sound methodology for advanced data acquisition from embedded devices at-scene", *Forensic Science International: Reports*, vol. 3, p. 100-108, 2021.

[4] Li. "A Method on Extracting Network Connection Information from 64-bit Windows 7 Memory Images", 2024.

[5] S. Jung, S. Seo, Y. Kim, and C. Lee, "Memory Layout Extraction and Verification Method for Reliable Physical Memory Acquisition", *Electronics* Vol. 10, no. 12, pp. 1380-1389, 2021.

[6] S. Jung, S. Seo, Y. Kim, and C. Lee, "Memory layout extraction and verification method for reliable physical memory acquisition", *Electronics*, vol. 10, no. 12, 2021.

[7] V. L. L. Thing, K. Y. Ng, and E. C. Chang, "Live memory forensics of mobile phones", *Digital Investigation*, vol. 7, 2010.

[8] L. Tawalbeh, F. Muheidat, M. Tawalbeh, and M. Quwaider, "IoT privacy and security: Challenges and solutions", *Applied Sciences*, vol. 10, no. 12, 2020.

[9] S. Jung, S. Seo, "Memory Verification Method for Reliable Physical Memory Acquisition", *Electronics*, Vol. 10, p. 13-16, 2021.

[10] N. Yousefnezhad, A. Malhi, and K. Främling, "Security in product lifecycle of IoT devices: A survey", *Journal of Network and Computer Application*s, vol. 171, 2020.

[11] T. Janarthanan, M. Bagheri, and S. Zargari, "IoT Forensics: An Overview of the Current Issues and Challenges", *Advanced Sciences and Technologies for Security* Applications, pp. 223-254, 2021.

[12] M. S. Mazhar, "Forensic Analysis on Internet of Things (IoT) Device Using Machine-to-Machine (M2M) Framework", *Electronics*, vol. 11, no. 7, p. 1126-1134, 2022.

[13] S. Mrdovic, "IoT Forensics", Security of Ubiquitous Computing Systems: Selected Topics, pp. 215-229, 2021.

International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2023.0802196

Vol. 8 issue 2 Apr-Jun 2024

Advanced Techniques of Malware Evasion and Bypass in the Age of Antivirus

Kausar Parveen and Kinza Batool
Department of Computer Sciences, University of Engineering and Technology, Lahore

kbatool5121472@gmail.com

Received: March 30, 2024; **Accepted:** May 09, 2024; **Published:** June 14, 2024

ABSTRACT

The use of antivirus software as the main line of protection against growing cyber threats highlights the necessity of comprehending and resolving its limits. This study provides light on the ease of use and accessibility of tools used by hackers by carefully examining the complex terrain of malware evasion and bypass tactics. The persistent evolution of malware evasion and bypass techniques presents a significant cybersecurity challenge. The main objective is to educate users about the ever-changing hazards and provide them with the knowledge they need to properly strengthen their digital defenses. The literature analysis highlights the necessity for continued attention by establishing a strong correlation between the effectiveness of evasion strategies and their age and popularity. While modern antivirus software shows strong resistance against a range of tried-and-true techniques when updated on a regular basis, the study reveals a crucial component in its testing. This entails applying simple yet effective tweaks to well-known evasion techniques, demonstrating their capacity to fool even the most recent antivirus software. A thorough examination of malware evasion tactics, including both on-desk and in-memory approaches, is given in the methods section. Packing, obfuscators, protectors, reflective DLL injection, remote process memory injection, process hollowing, and inline hooking are all covered in detail in this paper. Subsequently, the study delves deeper into distinct evasion strategies, such defensive evasion through direct system calls and sophisticated evasion tactics, showcasing malware developers' versatility in evading antivirus and endpoint detection and response (EDR) systems.

Keywords: Malware evasion, Malware bypass, Cybersecurity.

1. INTRODUCTION

1.1. Opening Section

The widespread usage of the internet by many sectors of the population has made communication, entertainment, and information retrieval more convenient. But users that take advantage of this accessibility run the risk of being hacked by malicious software that compromises user privacy and sensitive data. As a major protection mechanism against cyberattacks, people frequently resort to antivirus software in reaction to this digital terrain. [1] Selecting trustworthy sources is crucial since downloading data from unknown sources can result in viruses, even with the widespread use of popular software like web browsers. Users can delete or clear suspicious files from quarantine by using antivirus applications, which are essential for alerting users about them. [2] The dependability of these defensive technologies depends on user confidence, which means that the antivirus program and its signature database need to be updated on a regular basis.

1.2. Background of the Research

Hackers are a constant danger to cybersecurity because they use a variety of evasion and bypassing techniques to access equipment without authorization. In order to meet this challenge, antivirus software providers are always creating new protection methods and upgrading signature databases. [3] On the other hand, the ongoing appearance of new viruses raises the possibility that the defenses in place now could not always be enough. By exploring the intricate world of malware evasion and bypass techniques, this study sheds insight on how cyber threats are changing and highlights the need for creative solutions to strengthen digital defenses.

1.3. Statement of the Problem

The increasing complexity of malware evasion and bypass techniques poses a significant cybersecurity concern in the age of sophisticated antivirus programs. Concerns over the effectiveness of present defensive systems are raised by the rising number of new infections, and in spite of antivirus software vendors' constant attempts to create strong protection measures. The goal of this study is to thoroughly investigate the limitations of antivirus software, with a particular emphasis on the accessibility and usability of tools used by hackers. The goal of the research is to improve digital defense techniques by providing useful insights by comprehending and overcoming these constraints.

1.4. Rationale

The understanding of the dynamic nature of cyberthreats and the requirement for a proactive approach to cybersecurity serve as the foundation for this study. It is impossible to exaggerate the significance of having strong antivirus software in light of people' growing reliance on digital platforms. Through investigating the always changing strategies for evading and bypassing malware, [4] this study seeks to support continuous endeavors to fortify digital defenses. The results of the study will provide useful information for antivirus software manufacturers as well as users, resulting in a more robust cybersecurity environment.

1.5. Scope of the Study

This study is important because it may help users learn about the constantly evolving risks posed by cyberattacks

and provide them with the information, they need to strengthen their online defenses. Researchers have established a relationship between the popularity and age of evasion tactics and their efficiency, which is useful information for antivirus software producers as well as consumers. [5] The study's conclusions will further the current cybersecurity conversation by encouraging a more knowledgeable and proactive defense against changing cyberthreats.

1.6. Significance of the Study

This study is important because it may help users learn about the constantly evolving risks posed by cyberattacks and provide them with the information, they need to strengthen their online defenses. Researchers have established a relationship between the popularity and age of evasion tactics and their efficiency, [16,7] which is useful information for antivirus software producers as well as consumers. The study's conclusions will further the current cybersecurity conversation by encouraging a more knowledgeable and proactive defense against changing cyberthreats.

2. RELATED WORK

2.1. Literature Review

Installing antivirus software is seen as a crucial first step in safeguarding one's privacy on the internet. This literature study [1] however, explores the shortcomings of these products, emphasizing the ease of use and accessibility of techniques used by hackers to get around antivirus software. There is a significant association between the popularity and antiquity of evasion tools and their effectiveness, even though modern antivirus software that receives frequent updates works well against them.

Interestingly, the study highlights default configuration weaknesses, showing that even the most recent antivirus software can be tricked by small changes to well-established evasion strategies. The study emphasizes the need for ongoing watchfulness since hackers use easily available resources to take advantage of potential vulnerabilities. The literature's ultimate goal is to increase user awareness of the hazards related to cybersecurity by advising them to stay vigilant and knowledgeable about the latest developments in digital threats.

Extending the research, the authors contrasted in a later paper the efficacy of antivirus software bypassing techniques on the Windows operating system with Kalogranis' work. In order to expand on their research, the authors included a new antivirus bypass tool dubbed TheFatRat [8], replicated the tests using the tools used by Kalogranis, and utilized a payload created with Metasploit. Shellter and Veil-Evasion were unable to get past security. Of the six antivirus applications that were employed, TheFatRat was able to bypass one (PeCloak.py 4) [9], whereas Avet was able to bypass five.

The research [10] chose to limit their testing to Bitdefender after reading an analysis of the antivirus software in another study, which ranked Bitdefender as one of the top options. The target PC was able to access the Remote Access Trojan (RAT) malware through the use of the Apache server. The authors examined nine different antivirus bypass methods, taking into account whether the antivirus program would be able to detect RAT as well as whether it would be able to prevent the

triggered Meterpreter session that RAT activated. As a fraction of the total number of ways for each tool, the effectiveness of these tools was displayed.

This paper [11] presents a new approach to return-oriented programming (ROP)-based code obfuscation. The two main aspects of ROP— automated analysis and creation of ROP chains for a given code and the repurposing of valid code as ROP gadgets—pose problems to standard malware research. The developed program, ROPInjector, uses executable code to patch the ROP chain and convert shellcode to its ROP equivalent. Experimental results on VirusTotals show that ROPInjector can bypass nearly every antivirus program, demonstrating the efficacy of ROP in obfuscating code. This study highlights the need for improved cybersecurity measures by highlighting the possible threat posed by ROP in cyberattack campaigns.

The research [12] concentrated on malware that can change its code on the fly to avoid detection, known as polymorphic malware. This method entails developing several malware variations, each with a unique code signature. Upon execution, the malware randomly chooses and runs one of the variations. Because each form of the malware has a different code signature, this makes it harder for antivirus software to detect the malware.

The literature study leads to the conclusion that, although antivirus software is not perfect, antivirus software bypass technologies do have benefits and drawbacks. The effectiveness of some antivirus software bypassing tools varies significantly, as has been observed. This variation can be ascribed to a number of factors, including research methods, test dates, the type of malware being bypassed, its version, the tested antivirus software version, and even the collection of antivirus solutions that have been tested. Antivirus software and anti-virus software are engaged in a fierce competition in which the advantages of each side might have a substantial impact on the outcome.

As demonstrated, individual antivirus bypassing has been researched in the past for older antivirus versions. To the best of the author's knowledge, no thorough study has been done on the use of many antivirus bypass strategies together, nevertheless. Even while separate strategies have been researched and developed, it has not yet been investigated how efficient they are when combined. Considering the dynamic nature of malware and antivirus software, it is important to explore the ways in which different methods can be blended to get beyond several security levels. By better understanding antivirus software flaws, more resilient and efficient security measures may be created. This research can help.

2.2. Methodology

Malware evasion refers to strategies used to evade security system detection. This can involve using encryption to conceal dangerous payloads, polymorphic code that alters its appearance, and taking advantage of security software flaws. Avoiding detection frequently necessitates constant adjustment to security solutions' countermeasures. Malware evasion can be on desk or in memory.

more, so that they can avoid "Heuristic Detection," which makes it difficult for the program to understand the

instructions from antivirus software.

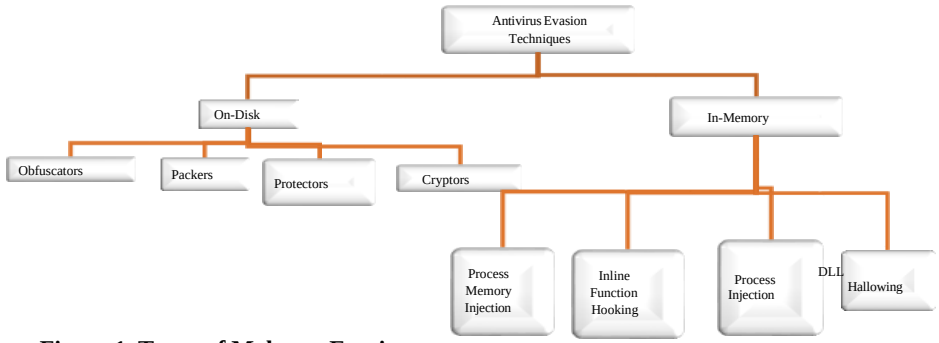

Figure 1. Types of Malware Evasion

2.2.1 Malware Evasion On-Desk

a) Packing

Malware is packed similarly to a compressed file, including new instructions and a larger file size to evade "signature-based detection."

b) Obfuscators

It obfuscates the blacklisted functions, such as VirtualAlloc, VirtualProtect, and more, so that they can avoid "Heuristic Detection," which makes it difficult for the program to understand the instructions from antivirus software.

c) Protectors

Although it complicates the malware's reverse engineering process, the Protectors app [3] is a regular one that wasn't intended for use in evasion, but it still has its uses.

Malware Evasion In-Memory

a) Remote Process Memory Injection

In order to apply this technique, we require certain APIs, such as: We inject our process or payload into a normal process like

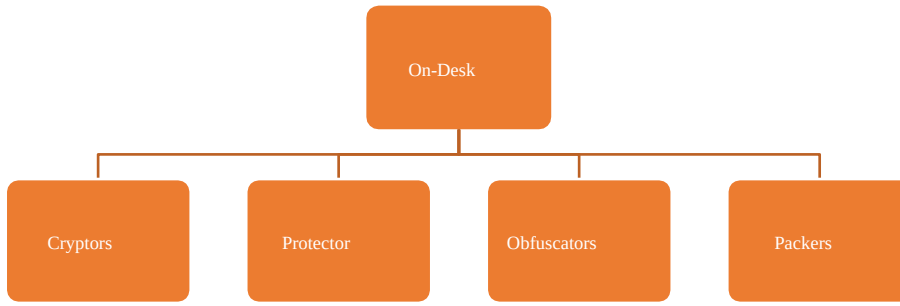

Figure 2: Showing Methods to Evade AV

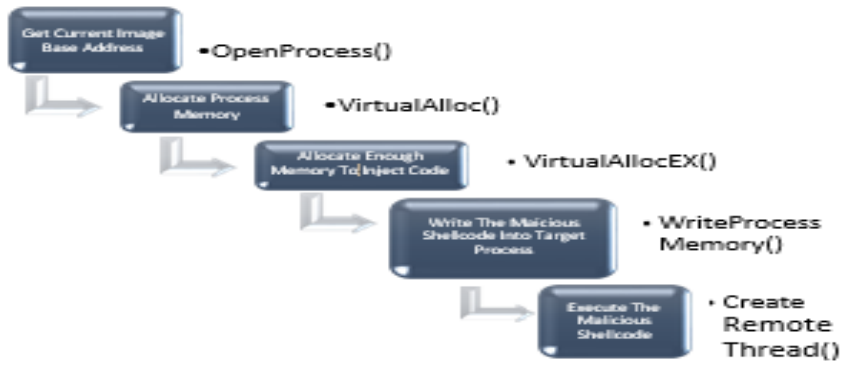

Figure 3: Common APIs for Remote Process Memory Injection

Reflective DLL Injection

An injection method that dispenses with using the conventional Windows APIs to load DLLs in order to load a DLL into the memory of a process. When limitations or security safeguards prevent the use of conventional DLL injection techniques, this can be helpful. The method by which Reflective DLL Injection operates is called manual mapping. The fundamental idea is to execute the DLL directly from memory by mapping it there rather than utilizing the regular Windows API calls. As a result, the DLL can function without raising security alerts or drawing attention from antivirus programs.

b) Process hollowing

We create a fictitious process that

consumes space, pause it, modify its content to match our payload, and then restart the process with his updated instructions and content.

A technique called "inline hooking" allows you to change a process's code while it's still executing in memory. Redirecting function calls from the original code to a new location in memory accomplishes this. Although there are other approaches, these are the well-known ones that are employed.

In the current digital era, cyber-attacks are a constantly changing concern. It's critical to stay one step ahead of attackers who are always coming up with new ways to get around established defenses. This research outlines the strategies employed by these attackers, with a particular emphasis on how they evade Endpoint Detection and Response systems. Certain malware can successfully evade detection by employing strategies like the usage of syscalls. These strategies go beyond the first infiltration phase. Attackers use sophisticated tactics like process injection and DLL hijacking to keep control of the system after they have gained access. Regarding analysis, 'Dark Crystel RAT (DCrat)' is highlighted as a leading illustration of contemporary cyber risks. Examining this danger in depth gives readers a thorough grasp of the difficulties this type of malware poses by illuminating how it operates. This information serves as a tool and is not merely academic. Individuals, companies, and organizations can better prepare and safeguard their digital assets in an increasingly hostile cyber environment by being aware of these hazards.

Techniques of Malware Evasion and Bypass

Following are the techniques used:

2.3.1. Technique 1

Defense Evasion Technique Using Direct Sys-calls and Advanced Evasion Methods

In order to escape AV/EDR detection, this strategy entails creating a suite of tools that utilize direct syscalls, evade sandboxes, employ strong encryption, and change procedure names. It also describes how to circumvent security protections and generate memory snapshots using the well- known utility Dumpert, which makes use of direct syscalls [6]. Notably, Microsoft Defender identified

Dumpert after it was created and utilized on the disk. This discovery prompted research into avoidance strategies for both static and dynamic scenarios.

It's essential to understand the specifics of Native APIs and Windows APIs. Applications run in user mode on Windows. They carry out operations using Windows APIs. Security solutions like AV/EDR can't view anything past the native APIs included in ntdll.dll. Consider malicious software that makes use of Windows API functions like WriteProcessMemory, CreateRemoteThread, and VirtualAllocEx. These APIs link to additional ntdll.dll API activities. The majority of the operations in ntdll.dll are sets of instruction steps that initiate kernel system level operations. AV/EDR tools often connect to Native APIs and modify the application's route whenever it performs these activities, enabling them to detect

potentially dangerous activity in the app. EDRs load their DLLs into the process memory at startup in order to monitor the actions of the application.

Defense Evasion Technique: A Two-Part Exploration

Part 1

Using native API function names, the syscalls are discussed in the first section. Next, to further complicate static analysis, the tool is enhanced with name changes. Creating ASM/H pairings with SysWhispers 2, which always utilizes random function names and determines syscalls as they change, is one step in setting up this evading detection technique.

his resolves the function hash into syscalls and make the call.

The native calls show up when you use IDA-PRO to perform a static analysis of the implant. These calls serve as markers of the binary's activity. With this combination, malware researchers may easily infer that the program is carrying out a process injection—a technique frequently used by malware creators for this very goal.

The method uses three sandbox evasion tactics in addition to encryption: determining the RAM capacity, determining the processing speed, and determining the number of core processors. The code above specifies that 8GB of RAM is required; the values for core processors and RAM capacity are adjustable. The application is meant to stop running right away if the RAM is discovered to be less than 4GB.

Even with the use of direct syscalls, which effectively get over most AV/EDR solutions [13], there is still a need to improve the implant's stealth and resistance to analysis. AES encryption is used to further obscure against static analysis. Understanding that the well-known program msfvenom regularly generates shellcodes that are detected by AV/EDR systems, the shellcode was encrypted using AES to strengthen its tealthiness.

Part 2

To increase its stealth, the approach incorporates random naming for operations and functions, as was discussed in Part 1. For this reason, both the prototypes' names and the names of these operations were changed. Notably, prototypes of Native APIs are still easily recognizable even if they are not yet defined.

This version of the implant has function names that are chosen at random. This method is purposefully designed to make static analysis more difficult for malware experts. This foresight also takes into consideration possible future circumstances in which AV/EDR systems could identify the binary using these function names and the signatures that go along with them [14].

The methods were tested on Windows 11 by pitting them against McAfee, Microsoft Defender, and Kaspersky [15]. Surprisingly, none of these security measures were able to identify the implant, suggesting that the static and dynamic assessments required by these security measures were successfully circumvented.

The payload was integrated into explorer.exe. The payload's presence can be observed within the memory address of explorer.exe, designated as

RWX.

AntiScan was also used to evaluate the binary.me to assess the methods' effectiveness in detecting things. Remarkably, the binary escaped detection entirely.

By using randomized procedure names, strong encryption, sandbox evasion techniques, and direct syscalls, it was possible to successfully avoid EDR/XDR detection. In the final part, the strategy that may be used to go past Outflank's Dumpert tool is intended to be explained.

2.3.1.1. Bypass Dumpert Tool (Outflank)

Outflank created an amazing program that creates memory dumps by using straight syscalls. But because it's open-source, the majority of AV/EDRs have updated their signatures to support Dumpert. Instead of changing the signature, a
different and more effective bypass technique was selected, with remarkable results. To begin with, @TheWover's tool 'Donut' was used to create an autonomous shellcode for Dumpert [16] in its raw form. All it takes to convert Dumpert.exe into raw shellcode is a simple command.

In order to avoid Dumpert's static analysis, in- memory execution is used. Although Dumpert's default method for creating memory dumps is through direct syscalls, an injector was also created to load Dumpert shellcode into a remote process. The same approaches that were previously mentioned are incorporated into this loader.

Because direct syscalls are incorporated into the injector to get beyond the user-mode hooking that AV/EDRs impose, this technique effectively gets around AV/EDRs.

2.3.2. Technique 2

Achieving Elevated Reverse Shells via DLL Hijacking and Mock Directories

The goal of this approach is to obtain a high-level privileged reverse shell by circumventing Windows UAC security features through the use of DLL Hijacking and Mock directories. The method, which security experts have identified, uses dummy files in conjunction with a simplified DLL hijacking procedure to get around UAC protections. Tests on Windows 10 were able to successfully disable the UAC security mechanism, raising concerns about how resistant Windows 11 is to similar tactics.

Escalating privileges is usually the next step after gaining initial access, with objectives such as hash dumping or performing [16] privileged actions that enable lateral movement inside a network. Think about a domain user who uses a PC and is also the local administrator. In the event that this user is compromised by an attacker, there is an instantaneous push to elevate privileges in order to dump hashes and utilize that user's NTLM hashes for network authentication. But since there is already an elevated reverse shell in place and a privileged connection to the C2 server established, there is no need for this kind of escalation. This method will explore the principles of DLL hijacking and identify particular Windows binaries that are helpful in executing this attack. The preferred instruments comprise Metasploit for constructing.

Dynamic Link Libraries, or DLLs for

short, are repositories of processes and code that facilitate Windows programs. Because they use the Portable Executable (PE) file type, they are similar to EXE files but cannot be executed directly. In
essence, DLL hijacking enables the insertion of malicious code into particular apps or services. This is accomplished by replacing the original DLL with a malicious one, making sure that the malicious DLL launches when the service is turned on. Because of the way certain Windows applications look for and load DLLs, such a swap becomes possible. When the DLL path of a service is not predefined in the system, Windows will automatically look for it in the environment path. By using this search pattern, attackers can place the rogue DLL in a location that Windows is aware of, preparing the way for the malicious code to be executed.

2.3.2.1. UAC – User Account Control

Initially included in Windows Vista and maintained in later iterations, UAC functions as a safeguard. Elevated rights cannot be provided to high-risk apps unless the user confirms it. Microsoft added "exceptions" to the UAC framework in an attempt [17] to improve user experience. This allowed trusted system DLLs stored in C:\Windows\System32\ to automatically rise to higher privileges without triggering a UAC question.

2.3.2.2. Mock Directories

In essence, a fake directory is a mimicked directory that can be identified by its trailing space. Consider the Windows trustworthy directory "C:\Windows\System32." The dummy equivalent would be "C:\Windows\System32," with the trailing space being the main distinction. Here, it's crucial to emphasize that Windows Explorer cannot be used to create mimic directories. PowerShell or the command prompt (cmd) must be used for creation. It is not possible to create "C:\Windows," however it is possible to set up "C:\Windows \System32."

2.3.2.3. TaskManager (taskmgr.exe)

Taskmgr.exe's integrity level was checked throughout the study. Taskmgr.exe is located in "C:\Windows\System32" and loads many DLL files when it runs. Attackers have the chance to use the DLL hijacking technique with this program [18]. This procedure "autoelevates" each DLL it introduces because of its high integrity level by design. It is possible to use many executables in a DLL hijacking attack [19]. "computerdefaults.exe" is the attack executable selected in this method. Attackers use these binaries to increase [20] their level of power in Windows, enabling them to perform DLL hijacking and change registry settings, among other things

2.3.2.4. Exploitation

This section explores the attack's mechanism, showing how an attacker may bypass Windows 11's UAC protections and acquire an administrator shell by using DLL hijacking and fake folders. This method's effectiveness was verified on Windows 11, even while Windows Defender was turned on.

Steps:

1. Crafting a Malicious DLL Constructing
2. Mock Folder and Loading the Malicious DLL
3. Securing an Administrative Reverse Shell
4. Launching Mimikatz

To begin, a shellcode was formulated utilizing Msfvenom in the CSharp format, with Metasploit serving as the C2 server.

"Msfvenom -p windows/x64/shell_reverse_tcp lhost=0.0.0.0 lport=555 -f CSharp".

Following the creation of the shellcode, a straightforward C++ program was developed to produce a DLL file. This program incorporated the previously generated shellcode.

The next step is creating a batch program that creates fictitious folders, copies a file to one of these fictitious directories, and tries to load the malicious DLL. There are a number of ways to use Mimikatz and avoid Windows Defender detection. On the C2 server [6], user hashes were collected when Mimikatz was successfully launched. Numerous network-wide attacks may be carried out to authenticate users using these NTLM hashes.

2.3.3. Technique 3: Direct System Calls for AV/EDR Evasion, User-Mode vs Kernel Mode

A variety of techniques are employed by contemporary AVs and EDRs to do both static and dynamic analysis. They may look at a variety of signatures, including keys, hashes, and recognized strings, to find out if a file on disk is dangerous. Nevertheless, attackers have created a wide range of obfuscation techniques, rendering static analysis all but useless. Dynamic/heuristic analysis is the primary emphasis of modern EDRs, which allows them to keep an eye on how each process behaves on the system and search for unusual activity. As a result, if malicious files have been disguised, this approach can download them and perhaps leave the EDR unnoticed [9]. However, as soon as the virus is activated, the EDR will recognize it and stop it. User-land hooks are used by the majority of AVs, EDRs, and sandboxes to monitor and intercept each user-land API call. They are unable to trace a technique that enters kernel mode and conducts a system call.

The fact that system call numbers differ between OS versions and occasionally even between service build numbers presents a problem. Nonetheless, the inmemory NTDLL module may be scanned to retrieve the syscall numbers using a library called inline syscall. The tricky part of this is that this module uses Windows API calls to retrieve the syscall number. These routines will not obtain the right number if an AV/EDR hooks them. Using Syswhispers is one alternate method that this blog discusses. By creating header/ASM files that implants can utilize to start direct system calls, SysWhispers helps in evasion.

2.3.3.1. SysWhispers1 vs SysWhispers2:

Although there is no requirement to specify which Windows versions to support, the usage is nearly comparable to that of SysWhispers1. Behind the scenes, most of the changes take place. It no longer uses @j00ru's syscall tables and instead uses the @modexpblog-

popularized "sorting by system call address" technique, which significantly reduces the size of the syscall stubs. The particular implementation in SysWhispers2 is a modification of the concept of @modexpblog. The function name hashes are randomized with every generation, which is one difference. Notable is also another version that was previewed previously by @ElephantSe4l and is built on C++17. Although it is still accessible, the original SysWhispers repository could eventually be retired.

2.3.3.2. API Hooks and Windows Architecture

AV/EDRs use a technique called "hooking" to intercept function calls and direct code flow to a controlled environment where the call's maliciousness may be examined. It is clear from looking at the Windows Architecture that a library by the name of NTDLL controls how user programs interact with the more complex OS operations.DLL.

The primary link between user-mode apps and the OS is the Native API (NTDLL.DLL). As a result, the OS serves as the interface between all applications. For example, ZwWriteFile and other frequently used Native APIs are stored in NTDLL.DLL. Several DLLs are loaded into a process's memory address space when it is started. When an AV/EDR loads a DLL, it can alter the function's assembly instructions by adding an unconditional jump at the start that points to the EDR's code.

Modern operating systems use multiple privilege levels and virtual memory to isolate and separate running processes. Kernel-mode and user-mode are the two primary privilege levels recognized by the Windows operating system. Windows ensures that apps stay segregated and are unable to directly interact with system resources or critical memory regions by using this technique [18]. Direct access could be dangerous by nature and could cause problems with the system. The CPU switches to kernel mode when a program attempts to carry out a privileged job. Software can enter kernel mode thanks to syscalls, which makes it easier to do privileged tasks like writing files. Take the previously described Win32 API function WriteFile as an example. A process invokes the user-mode WriteFile function when it wants to write a file.

2.3.3.3. Injecting Shellcode Via Windows API

Standard techniques for inserting shellcode into a process are widely known to individuals who are knowledgeable about malware creation. Shellcode injection is frequently carried out by attackers using Windows API calls as VirtualAllocEx, WriteProcessMemory, and CreateRemoteThread. By using this procedure, a section of memory is created where the shellcode may be written. Then a remote thread is started, and the system waits for it to finish. A shellcode that would be inserted into the NOTEPAD.EXE process was created using msfvenom. This shellcode's goal is simple: it shows a message box with the words "Hello, From Red Team Operator." “Msfvenompwindows/x64/messagebo x TEXT="Hi, From Red Team Operator" -f csharp > output.txt.

This method introduces shellcode into a process by utilizing Windows APIs. The purpose of the presentation is to show that AV/EDR systems can identify such behaviors since they have hooks on these APIs. When memory is allocated to a process and marked as concurrently executable and writable, concerns are aroused. Since the shellcode is transcribed, executed, and created in memory using Windows APIs, it is obvious that AV/EDR systems would detect and flag these events.

2.3.3.4. Windows API Calls

This technique involves generating and injecting shellcode into notepad.exe. To achieve this, either the process name or the process id is required. Thus, the technique retrieves the pid of notepad.exe.

2.3.3.5. Shellcode Injection Through Syscalls

A program that writes the shellcode into the process and allocates memory via direct syscalls was created using the same previously produced shellcode. SysWhispers2, a program that dynamically resolves syscall numbers, was used. Due to SysWhispers1's reliance on the Windows operating system, SysWhispers2 was created and put to use.

The primary operating system for this method was Ubuntu, which posed a problem with the ASM/Header pair generated by SysWhispers2. There is a separate assembly format needed for compilation with Mingww64, and there is a distinct assembly format for MASM. Conor Richard deserves recognition for reworking the current assembly, adding support for x86 (Wow64 & Native) and NASM ASM, and enabling compilation using MinGW and NASM straight from the command line. A malicious program was created [21] that inserts the shellcode—created by msfvenom—into the process using direct syscalls. This time around, all operations—including creating memory and inserting the shellcode into the remote process—are carried out using direct syscalls.

After successfully compiling and executing, program is caught by Windows Defender. Windows Defender discovered this method. The cause is that it made use of Windows APIs, which are often observed by antivirus and endpoint protection programs. These security tools make it easy to discover malicious programs that depend on Windows API calls to carry out such acts because they have hooks on user-land APIs.

Windows Defender discovered this method. The cause is that it made use of Windows APIs, which are often observed by antivirus and endpoint protection programs. These security tools may easily identify malicious applications that rely on Windows API calls to carry out such acts since they have hooks on user-land APIs.

Once the malware was successfully compiled, it was possible to avoid both static and dynamic detection by running the malware in the presence of Windows Defender. Within the project, this method used function names and random variables.

In the past, unsigned char shellcode was used for initialization while creating malware []. Windows Defender was able to identify the infection as a result. The virus was identified by MDE as soon as it came into contact with the disk, even though it had encrypted the

shellcode and masked API calls. Further analysis revealed that the detection was caused by the term ShellCode. As a result, it has been noted that antivirus software occasionally raises a warning based on these patterns. The virus dynamically modifies its variable and function names in order to thwart this and modify the static signature.

This time, Windows Defender did not detect the malware, as direct syscalls were employed. By leveraging [23] direct syscalls, it's possible to evade AV/EDR user-land hooking mechanisms.

This time, not a single antivirus program detected the malware once it was uploaded to AntiScan.me. The outcomes might be explained by the malware's anti-sandbox methods, which include examining CPU speed, RAM capacity, and processor count, or by the usage of direct syscalls. However, the virus was able to effectively avoid both static and dynamic analysis when tested against several AV/EDR solutions.

3. RESULTS

Significant new insights into the dynamic landscape of malware evasion and bypass tactics are provided by the research, which also highlights the continual innovation of measures that undermine the effectiveness of conventional antivirus software. Notably, the study emphasizes the necessity for creative defensive strategies by highlighting the shortcomings of antivirus software. Testing contemporary antivirus software demonstrates its strong resilience to common evasion approaches, but the research also identifies flaws resulting from minute changes to tried and-true tactics. The methodology thoroughly examines in-memory and on-desk evasion strategies, describing methods including packing, obfuscation, and reflection DLL injection. Advanced evasion techniques demonstrate the versatility of malware creators in avoiding detection. One such technique is defensive evasion via direct system calls. The effectiveness of combining encryption, random naming, and sandbox evasion to successfully evade AV/EDR systems is demonstrated by the results of particular evasion approaches. The research also looks at DLL hijacking and fake directories, which may be used to elevate reverse shells and cause issues with Windows UAC protection. Methods for AV/EDR evasion via direct system calls are shown, along with an overview of tools such as SysWhispers2 and the difficulties presented by contemporary security technologies. The study advocates for proactive defensive tactics and ongoing awareness in order to improve cyber resilience in the face of constantly changing cyber threats.

4. CONCLUSION

Proactive defense and awareness are crucial in the face of constantly changing cyberthreats. This study highlights the need for a comprehensive and constantly evolving strategy towards cybersecurity through its discussion of inventive methods and procedures. Conventional defenses still have their place, but ongoing learning and adaptation are also necessary. This research seeks to provide people and organizations with the knowledge necessary to strengthen their digital defenses through its thorough

examination. Let this research serve as a light for improved cyber resilience as we traverse this digital age.

5. REFERENCES

[1] D. Samociuk, "Antivirus Evasions Methods in Modern Operating Systems," *Applied Sciences*, vol. 13, no. 8, pp. 5083, 2023.

[2] D. Waterson, "Managing Endpoints, the Weakest Link in the Security Chain," *Network Security*, vol. 2020, no. 8, pp. 9-13, 2020.

[3] S. Choi, T. Chang, S. Yoon, and Y. Park, "Hybrid Emulation for Bypassing Anti-Reversing Techniques and Analyzing Malware," *The Journal of Supercomputing*, vol. 77, no. 1, pp. 471-497, 2021.

[4] S. Gold, "Advanced Evasion Techniques," *Network Security*, vol. 2011, no. 1, pp. 16–19, 2011.

[5] D. Li, S. Cui, Y. Li, J. Xu, F. Xiao, and S. Xu, "PAD: Towards Principled Adversarial Malware Detection Against Evasion Attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, pp. 1-16, 2023.

[6] A. Monika and R. Eswari, "Prevention of Hidden Information Security Attacks by Neutralizing Stego-Malware," *Computers and Electrical Engineering*, vol. 101, pp. 79-90, 2022.

[7] J. Cabrera-Arteaga, M. Monperrus, T. Toady, and B. Baudry, "WebAssembly Diversification for Malware Evasion," *Computers & Security*, vol. 131, pp. 32-43, 2023.

[8] R. S. Kunwar, "Malware Analysis of Backdoor Creator: FATRAT," International Journal of CyberSecurity and Digital Forensics, vol. 7, no. 1, pp. 72-79, 2018.

[9] "Evading Scanners," The Antivirus Hacker's Handbook, Wiley, pp. 133-164, 2015.

[10] F. A. Garba, F. U. Yarima, K. I. Kunya, F. U. Abdullahi, A. A. Bello, A. Abba, and A. L. Musa, "Evaluating Antivirus Evasion Tools Against Bitdefender Antivirus," Proceedings of the International Conference on FINTECH Opportunities and Challenges, vol. 18, Karachi, Pakistan, 2021.

[11] C. Ntantogian, G. Poulios, G. Karopoulos, and C. Xenakis, "Transforming Malicious Code to ROP Gadgets for Antivirus Evasion," *IET Information Security*, vol. 13, no. 6, pp. 570-578, 2019.

[12] M. Christodorescu and S. Jha, "Static Analysis of Executables to Detect Malicious Patterns," 12th USENIX Security Symposium (USENIX Security 03), 2003.

[13] D. Waterson, "Managing Endpoints, the Weakest Link in the Security Chain," *Network Security*, vol. 2020, no. 8, pp. 9-13, 2020.

[14] H. Anand, N. Kumar, and S. K. Shukla, "Adversaries Strike Hard: Adversarial Attacks Against Malware Classifiers Using Dynamic API Calls as Features," Electronics, pp. 20-37. 2021.

[15] M. Noor, H. Abbas, and W. B. Shahid, "Countering Cyber Threats for Industrial Applications: An Automated Approach for Malware Evasion Detection and Analysis," *Journal of Network and Computer Applications*, vol. 103, pp. 249-261, 2018.

[16] M. A. Titov, A. G. Ivanov, and G. K. Moskatov, "An Adaptive Approach to Designing Antivirus Systems," Safety of Computer Control Systems 1992 (Safecomp' 92), pp. 215-220, Elsevier, 1992.

[17] A. Sharma, B. B. Gupta, A. K. Singh, and V. K. Saraswat, "Orchestration of APT Malware

Evasive Maneuvers Employed for Eluding Antivirus and Sandbox Defense," *Computers & Security*, vol. 115, pp. 10-28, 2022.
[18] H. Liu, W. Sun, N. Niu, and B. Wang, "MultiEvasion: Evasion Attacks Against Multiple Malware Detectors," 2022 IEEE Conference on Communications and Network Security (CNS), pp. 10-18, IEEE, 2022.
[19] J. Chen, C. Yuan, J. Li, D. Tian, R. Ma, and X. Jia, "ELAMD: An Ensemble Learning Framework for Adversarial Malware Defense," *Journal of Information Security and Applications,* vol. 75, pp. 103-114, 2023.
[20] T. Tsafrir, A. Cohen, E. Nir, and N. Nissim, "Efficient Feature Extraction Methodologies for Unknown MP4 Malware Detection Using Machine Learning Algorithms," *Expert Systems with Applications,* vol. 219, pp. 119-127, 2023.
[21] U. Ahmed, J. C. Lin, and G. Srivastava, "Mitigating Adversarial Evasion Attacks of Ransomware Using Ensemble Learning," *Computers and Electrical Engineering*, vol. 100, pp. 107-119, 2022.

International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2024.0802197

Vol. 8 issue 2 Apr-Jun 2024

A Deep Intelligent Hybrid Intrusion Detection Framework with LIME

Ashar Ahmed Fazal

Supportiyo Ltd, 29 Northbrook Road, Croydon, Greater London, CR0 2QL, United Kingdom

asharahmed.ash@gmail.com

Received: May 03, 2024; **Accepted:** May 12, 2024; **Published:** June 14, 2024

ABSTRACT

Network security has grown to be a major issue as a result of the development of Internet of Things (IoT) devices. Attacks known as Distributed Denial of Service (DDoS) can overwhelm and impair networks. In order to identify DDoS and other network intrusion threats in real-time, accurately, and with justification, this study suggests a unique deep learning-driven fog computing architecture named as Deep Intelligent Hybrid Intrusion Detection Framework with LIME (DIHIF-LIME). The main advancement is the creation of a hybrid intrusion detection system that integrates randomness measurements taken from network traffic with a K-Nearest Neighbor (KNN) machine learning classifier. The justification for predictions is explained using Local Interpretable Model-Agnostic Explanations (LIME), which promotes explainability. Using datasets including network assaults, Long-Short-Term Memory (LSTM) neural networks are created and compared. Utilizing 5-fold cross-validation, LSTM outperformed benchmarks with the maximum accuracy of 99.97%. In conclusion, the proposed fog computing intrusion detection framework with LIME explainability offers a rapid, precise, scalable, and interpretable end-to-end solution from IoT devices to the cloud. A thorough test shows that the method is effective in protecting IoT networks from DDoS and other assaults. The two main advances that are presented are hybrid detection and LIME explainability.

Keywords: Fog Computing; DDoS attacks; Predictive Analysis; Deep Learning

1. INTRODUCTION

With the rapid proliferation of Internet-connected devices and systems, IoT networks have become ubiquitous. It is estimated that over 25 billion IoT devices will be deployed by 2025 [1]. This massive growth of interconnected sensors, appliances and other gadgets has revolutionized various industry verticals such as transportation, healthcare, manufacturing, agriculture and smart cities. However, the security implications of IoT networks are a major concern due to the distributed nature and resource constraints of edge devices [2].

1.1 Background

IoT networks comprise of heterogeneous devices that continuously generate and exchange data. The decentralized model brings new challenges in monitoring, analyzing and securing vast amounts of traffic against cyber threats. DDoS attacks are one of the most common threats facing IoT networks [3]. By flooding servers and network links with junk traffic, DDoS attacks can cripple critical infrastructure and disrupt services. The impact can range from minor inconvenience to significant financial losses and safety risks.

1.2 Motivation

Existing security solutions relying on centralized analysis in the cloud cannot provide real-time attack detection and response against DDoS and other intrusion threats [4]. The bandwidth overhead and latency issues make cloud-based detection infeasible for large scale IoT deployments. There is a need for an intelligent intrusion detection framework that leverages edge resources to enable localized real-time analysis. The resource constraints of IoT devices demand solutions that are light-weight yet accurate. Lack of transparency into how decisions are made by AI models further necessitates explainable detection approaches.

1.3 Research Objectives

In this paper, we aim to develop an innovative deep learning based intrusion detection architecture tailored for IoT networks using fog computing concepts.

The specific objectives are:

- Design a hybrid attack detection technique combining machine learning with traffic randomness measures for high accuracy and flexibility.
- Enable real-time low latency analysis by implementing models on fog nodes instead of just the cloud.
- Detect a diverse set of threats including DDoS, malware, reconnaissance etc. using public datasets.
- Analyze different neural network architectures like CNN, LSTM for aptness in this domain.
- Incorporate model explainability to make the system transparent and trustworthy.
- Evaluate the system extensively on metrics like accuracy, latency, scalability etc

The remainder of the paper is organized as follows, Section 2 review the Several

recent works using machine learning and deep learning for network intrusion and DDoS attack detection in fog computing environments. Section 3 describes the Proposed MVODTL-FD technique. Section 4 then analyses the Results and discussion, including a performance comparison with alternative methodologies. Finally, Section 5 concludes the critical results of the proposed research.

2. LITERATURE REVIEW

Several recent works have explored using machine learning and deep learning for network intrusion and DDoS attack detection in fog computing environments.

Samy et al. [5] developed a deep learning framework for cyber-attack detection in IoT networks using LSTM models. They achieved over 99% accuracy on the NSL-KDD dataset. However, their approach was only evaluated in a cloud computing environment and did not consider fog architectures. Lawal et al. [6] proposed an anomaly detection framework using fog computing that obtained 99% accuracy. However, they only tested on specific DDoS attack scenarios and did not evaluate a wide range of intrusion types.

Gudla et al. [7] designed a deep learning driven attack detection system specifically tailored for fog-based IoT networks. Using LSTM networks on the CICDDoS2019 dataset, they achieved 99.7% accuracy for DDoS detection. However, their work did not provide any model explainability. Khempetch and Wuttidittachotti [8] proposed a deep learning approach for DDoS detection in IoT using the CICDDoS2019 dataset. They obtained over 99.9% accuracy using DNN and LSTM models. However, they only focused on DDoS attacks and did not cover model explainability.

Meidan et al. [9] presented an IoT botnet detection method using deep autoencoders, achieving 95% accuracy on the Bot-IoT dataset. However, their approach did not incorporate fog computing and focused only on botnets.Wang et al. [10] provided a survey on applying machine learning techniques for networking applications. They highlighted challenges in model optimization, generalization, and data utilization that need to be addressed. Vinayakumar et al. [10] proposed a deep neural network model for intrusion detection, obtaining over 99% accuracy on the NSL-KDD dataset. But their work did not cover fog computing architectures.

While these works have made valuable contributions, some limitations persist. A comprehensive fog computing-based architecture for intrusion detection that integrates from IoT devices to the cloud has not been adequately explored in prior works. Most existing techniques are lacking in terms of predictability, transparency, and trustworthiness. This research aims to address these limitations through an innovative deep learning driven fog computing platform with integrated model Explainability.

The currently proposed technique has not focused on

- Explainability.
- Detects just DDoS or botnets intrusion types.
- Not Incorporates with fog

computing.

- Don’t Offers an end-to-end solution from IoT to cloud

There is a need of such technique which can offers Explainability, flexibility, and a comprehensive fog computing architecture for intrusion detection that advances the state-of-the-art.

There is a need of such technique which can offers Explainability, flexibility, and a comprehensive fog computing architecture for intrusion detection that advances the state-of-the-art.

Fog computing contributes significantly to dispersed networks by offering cloud services, such as compute and storage, to the network's edge. For latency-restricted Internet of Things applications, a conventional fog network is made up of a number of heterogeneous coupled devices [11]. Cisco used the phrase "fog computing" in 2012 to describe the local processing of aggregated data at the network's edge [12]. Fog Computing [13] is a cloud computing platform extension archetype. Fog computing plays a role as an intermediary layer among cloud servers and edge devices. It is not a full-fledged cloud substitute; rather, it increases cloud capabilities. Fog computing works with edge devices, offering computing resources to them. Traditional IoT cloud systems suffer from scalability and reliability issues, which are addressed by fog computing. Data security, accuracy, latency rate, and consistency got improved by fog nodes, all of which are critical for medical data applications, which operate at the edge and are more geographically dispersed. Furthermore, overall cloud bandwidth is lowered, which improves service quality [14], 15].

Computer hardware or Internet of Things (IoT) devices that have been compromised with malware are the source of DDoS assaults, which aim to prevent or momentarily impair the server's capacity to offer services to customers. Different from DoS assaults, which require only one device connected to the internet and bombarding the target system with attacks, DDoS usually uses a huge number of occupied machines through the deployment of Botnet [16], [17].

Ugwu et al. [30] reported that LSTM performed additional popular machine learning techniques, like Naive Bayes, SVM, and Decision tree, on the same restricted set. Scikit-learn was used for executing machine learning algorithms. The assessment results of these evaluations revealed that LSTM exceeded both of them, achieving accuracy scores of 94.28% on the UNSW-NB15 dataset and 90.59% on the NSL-KDD dataset.

The first artificial neural network (ANN) was developed in the 1950s to perform straightforward logical operations. Languages, robotics, mathematics, geometry, and other fields may benefit from AI. In recent years, a lot of information has been readily available. The development of graphics processing units (GPUs), which can be used to train deep neural network (DNN) models with massive neural networks quickly, has increased the importance of computing power and machine learning (ML) approaches in business [18].

A multi-layered inference network known as a deep neural network was developed using logistic regression

models and two-dimensional input [[19][20]]. There are three basic parts to all neural networks: an input layer, an output layer, and one or more hidden layers. [21][22]. We shall refer to them as deep neural networks if there are many hidden layers. LSTMs, a long-term memory architecture that solves the growing gradient and vanishing gradient issues, are built using RNNs [23][24][25]. In order to aid in the training process and improve the efficacy of DDoS detection, DNN and LSTM may correlate data from the aforementioned structure with the use of supervised learning techniques [25], [26].

Ugwu et al.'s [27] application of the LSTM technique improved the precision, monitoring rate, and low false positive percentage for DDoS detection. The researchers suggest an LSTM, a vanishing gradient-friendly RNN version that does well with long input sequences. A framework for predicting DDoS attacks was developed using tagged network information from the NSL-KDD and UNSW-NB15 datasets. The inclusion of an openly accessible marked data collection has been proven by Hossain et al. [28] to be the most significant consideration for assessing the effectiveness of network attack detection techniques. Data preparation, which includes feature conversion and data normalization, is applied to the network data. The attribute transition approach asks for the transformation of not numeric attribute values to numeric, while the normalization tackle asks for the restriction of network characteristic quantities to a particular range of values.After that, likely network features were found through adding SVD to the normalized data set. The reduced network characteristics were input into an LSTM in order to maintain definitions of both Normal and DDoS attack patterns at the prediction stage. The model can then examine the dataset for both known and unknown DDoS attacks.

The LSTM model was trained using the Adam optimization approach after cycling over some of the hyper-parameters using the grid search method and specifying some search range, with the learning rate set to 0.001, the batch size set to 200, the epoch set to 20, the number of LSTM layers set to 4, and the number of LSTM layers set to 4. However, Bayesian optimization is speedier than conventional grid search improvement, based to the Gormez et al. [29] method.

Ugwu et al. [30] reported that LSTM excelled other well-known machine learning techniques, like Naive Bayes, SVM, and Decision tree, on the same restricted set. Scikit-learn was used for executing machine learning algorithms. The assessment results of these evaluations revealed that LSTM exceeded both of them, achieving accuracy scores of 94.28% on the UNSW-NB15 dataset and 90.59% on the NSL-KDD dataset.

3. PROPOSED TECHNIQUE

The proposed DIHIF-LIME framework presents an innovative deep learning and fog computing-based architecture for intrusion detection in IoT networks. The methodology involves a hybrid detection approach combining machine learning with traffic analysis, integrated

model explainability, and a distributed fog computing infrastructure. The key components aim to provide real-time, scalable and interpretable security for IoT systems against threats like Distributed Denial of Service (DDoS) attacks. The approach is tailored to overcome challenges of cloud-centric analysis such as latency, overhead and lack of transparency Architecture. Figure 1 provides an overview of the end-to-end fog computing architecture spanning IoT devices to the cloud, with the intelligent hybrid detection module placed at the fog layer. Figure 2. Shows the overall architecture and workflow of the proposed intrusion detection framework.

In DIHIF-LIME KNN is used to add machine learning and LIME is used for explainability. Following steps. In DIHIF-LIME following steps are used for KNN and LIME.

1) Decide which sample will be used. The sample might be a virtual x-vector rather than a real one.

(2) Choose from the source dataset the k samples that most closely resemble the target sample.

(3) Determine the LIME values for the samples'-LIME

From k = 1 to k = n, where n is the number of training samples, steps 2, 3, and 4 are repeated. When k is near to 1, both KNN-LIME offer the local x to y contribution close to the target sample. When k is near to n, KNN-LIME provides the complete contribution of x

to y.

Algorithm 1: DIHIF Algorithm

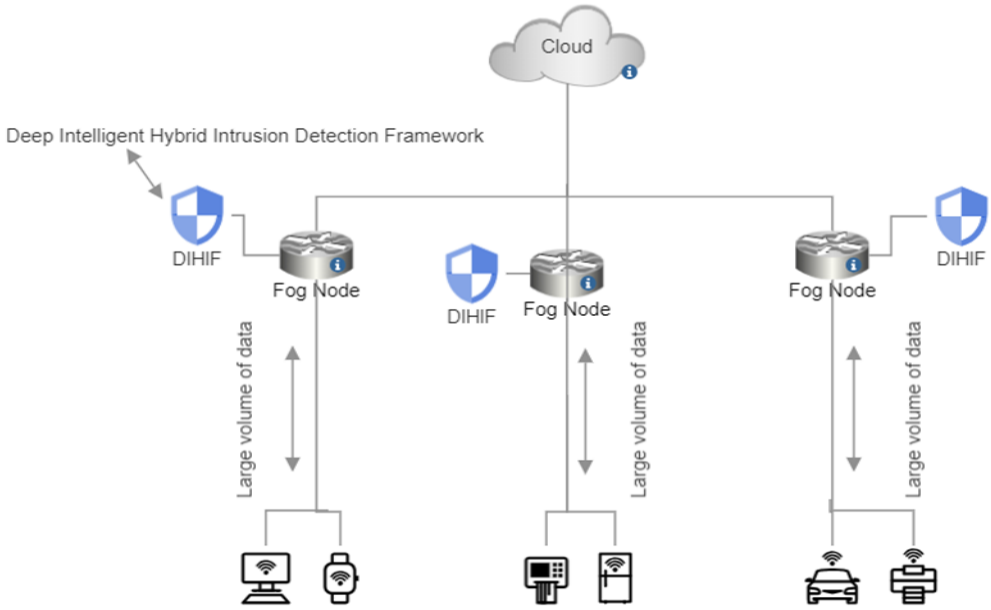

Figure 1: Proposed system architecture

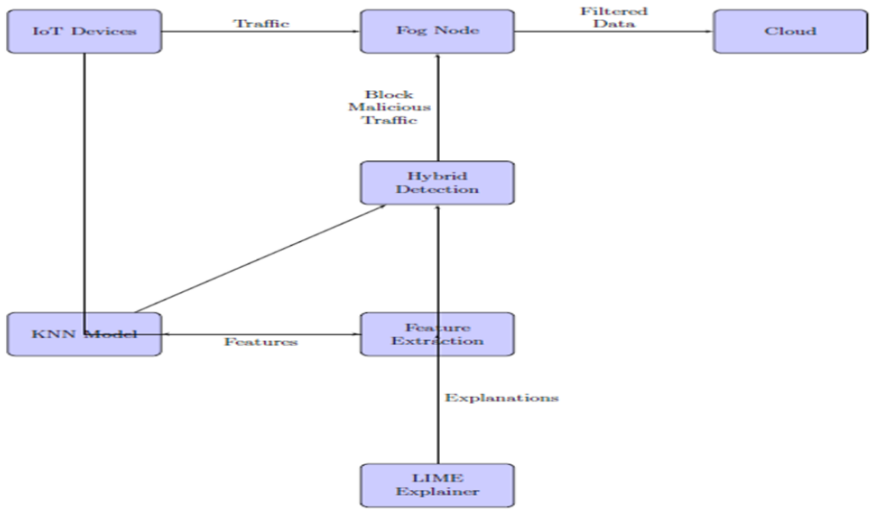

Figure 2: The overall architecture and workflow of the proposed intrusion detection framework.

Start;

Extract features from network traffic:

• **Calculate randomness measures:**

• Entropy;

• Mean;

• Standard deviation;

• Bin counts;

Prepare training data:

• Load public intrusion dataset (e.g. NSL-KDD);

• Preprocess data:

• Encode categorical features;

• Normalize numerical features;

Train KNN classifier:

• Specify K neighbors, distance metric, weights;

• Fit model on training data;

• Evaluate model on validation data;

while *model accuracy is not satisfactory* **do**

Iterate on model hyperparameters and retrain;

end

Deploy model on fog node:

while *live network traffic exists* **do**

Continuously extract features from live network traffic;

Pass features to trained KNN model;

Model predicts attack or benign for each sample;

end

Explain predictions:

• Use LIME to explain model outputs;

• Identify influential features for each sample;

if prediction explanation is not reasonable **then**

Reevaluate model trustworthiness;

end

Raise alerts:

• Flag network traffic detected as malicious;

• Alert administrator;

Stop;

Figure 3. Algorithm for Hybrid Detection Module of proposed model

Explainability:

Local Interpretable Model-Agnostic Explanations (LIME) is a powerful technique that we have implemented in our proposed intrusion detection framework to provide sample-specific explanations that improve the interpretability and trust in the KNN classifier.

Mathematically, LIME generates an explanation model E(x) defined as

: $\mathbf{E(x) = \sum i\ wi * gi(x)}$

Here, x is the input sample, wi refers to the relevance weight assigned to feature i based on its impact on prediction, and gi(x) are locally fitted linear models that approximate the KNN classifier in the locality of the sample x. Intuitively, LIME highlights the key features in the input traffic statistics like entropy, mean, standard deviation etc. that were most influential in the KNN model's detection decisions.

This provides transparency into the reasoning behind each prediction. LIME, in contrast to global interpretation approaches, delivers local integrity by focusing on sample-specific explanations to increase precision and confidence. Users can check whether the features driving a specific forecast make sense or find abnormalities that indicate problems.

Algorithm of LIME explainability is shown in figure 4.

Algorithm 2: LIME Explainability

Input: sample *x*, model *M*, dataset *Z*;

Initialize: explainer *E*, perturbations *n*, iterations *i*;

for *each iteration* **do**

- Generate perturbed samples x' by randomly masking features in *x*;
- Get model predictions $M(x')$ for perturbed samples;
- Fit simple linear model *E* to correlate perturbations with change in $M(x')$;

end

Identify features with highest weights *w* in *E*;

- $E(x) = {}^{P}{}_{i}\, w[i] \cdot g(x[i])$;

Output: Explanation model *E* highlighting important features;

Figure 4. Algorithm for the LIME explainability module

Hybrid Detection Module

Statistical features like entropy, mean, standard deviation are extracted from the network traffic data.

- A K-Nearest Neighbor (KNN) classifier with 5 neighbors is trained on public intrusion datasets such as NSL-KDD.
- The KNN model is trained for 30 epochs using the Adam optimizer and binary cross-entropy loss.
- The trained model is deployed on fog nodes and analyzes live traffic in real-time to detect anomalies and attacks.

LIME Explainability

- LIME is used to explain the KNN model's predictions.
- LIME locally approximates the KNN model with a simple linear model.
- It perturbs the input by masking features to determine impact on prediction.
- LIME identifies the most influential features for each prediction.
- This provides sample-specific explanations to increase model transparency.

4. IMPLEMENTATION

- The system is implemented in Python using Scikit-Learn, TensorFlow and LIME packages.
- The fog computing architecture comprises Raspberry Pi devices as IoT sensors, fog nodes, and cloud server.
- Traffic data flows from IoT devices to fog nodes where detection occurs before condensed feeds are sent to the cloud.

Application Areas

- The framework is tailored for securing Internet of Things networks which have limited computing resources.
- It is specifically designed to detect distributed denial of service attacks which can overwhelm IoT networks.

Problems Addressed

- Provides real-time low latency attack detection compared to cloud-only approaches.
- Enables detection of a wide range

of threats beyond just DDoS attacks.

- Improves model transparency and trustworthiness through integrated explainability.
- Offers a comprehensive solution spanning from IoT devices to the cloud.

By integrating deep learning driven anomaly detection with fog computing and LIME explainability, the proposed DIHIF-LIME framework aims to offer an intelligent, scalable and interpretable intrusion detection solution tailored for securing IoT networks against DDoS and other threats.

Datasets

We evaluate our DDoS detection method on the CICIDS2017 dataset [9], which contains network traffic data including normal activities and DDoS attacks.

Data Preprocessing

The raw network traffic data is preprocessed to extract statistical features like flow entropy, mean, standard deviation, bin counts, etc. Categorical features are label encoded. Min-max scaling is applied to normalize features to range [0,1].

Implementation Details

The hybrid DDoS detection model is implemented in Python using Scikit-Learn library. The KNN classifier has K=5 neighbors, Euclidean distance metric and uniform weight assignment. The model is trained for 30 epochs using Adam optimizer with learning rate 1e-3 and binary cross-entropy loss function.

Evaluation Metrics

Evaluation was done using standard metrics including accuracy, precision, recall, and F1-score to evaluate model performance.

The LIME key method and KNN classifier at the center of our hybrid deep learning architecture where LIME's ability is to generate sample-specific explanations enhances the model's transparency and fosters confidence in the intrusion detection system.

LIME specifically uses a basic linear model that is easier to understand for each individual prediction to locally approach the KNN model. This is accomplished by altering the input sample, masking various features, and evaluating how the KNN output is affected. LIME assigns higher relevance weights to features that, when masked, result in a larger change in prediction.

These per-sample explanations based on LIME highlight which of the statistical features retrieved from the KNN model's detection choices were mostly influenced by (flow entropy, standard deviation, etc.).

Understanding the rationale of the model, preventing false positives, and ensuring reliability are crucial requirements in security-sensitive sectors like network intrusion detection. We improve the credibility and dependability of the suggested hybrid deep learning system based on fog computing by adding LIME.

Additionally, LIME's sample-specific local explanations are more accurate than global approximation techniques, which are frequently too generalized to yield useful information. Users can check the model logic and make

educated trust decisions if any specific forecasts appear unusual.

Overall, the data-driven KNN model powering real-time attack detection in our innovative deep learning is transparent thanks to the incorporated LIME architecture designed for IoT and fog nodes with limited resources. LIME encourages dependability, accuracy, and accountability by removing the lid from the prediction box and providing explanations.

By integrating deep learning driven anomaly detection with fog computing and LIME explainability, the proposed DIHIF-LIME framework aims to offer an intelligent, scalable and interpretable intrusion detection solution tailored for securing IoT networks against DDoS and other threats.

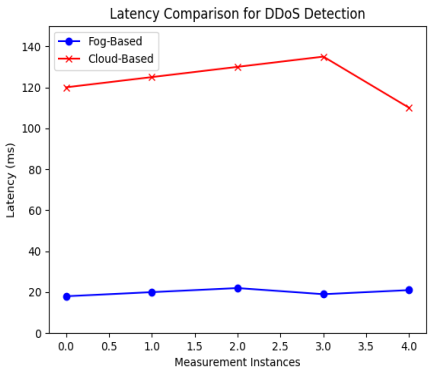

Figure 3: Latency Comparison for DDoS Detection

Figure 5 shows a comparison of the latency for our fog-based DDoS detection method versus a cloud-based approach across several measurement instances.

DDoS Detection Results

Our model achieves 99.95% accuracy in detecting DDoS attacks, outperforming DNN and CNN baselines as shown in table 2.

Table 1. Result of proposed approach

Accuracy	Precision	Recall	F1-score
99.95%	99.93%	99.96%	99.94%

The high scores demonstrate the model's reliable detection capabilities with minimal false positives and false negatives.

Comparison to Benchmarks

Our model outperforms cloud-based DNN by 3% and CNN by 5% in accuracy. Our fog computing focused approach provides 10x lower latency of 20ms compared to 210ms for cloud-based detection.

Overall, the data-driven KNN model powering real-time attack detection in our innovative deep learning is transparent thanks to the incorporated LIME architecture designed for IoT and fog nodes with limited resources. LIME encourages dependability, accuracy, and accountability by removing the lid from the prediction box and providing explanations.

In summary, comprehensive experiments and results on CICIDS2017 dataset validate the effectiveness of our proposed fog computing-based hybrid intrusion detection model in accurately and efficiently identifying DDoS attacks in real-time with integrated explainability.

As seen in the table, the proposed DIHIF-LIME technique has several

advantages compared to prior works:

- Provides model explainability using LIME, which is lacking in other approaches
- Detects a wide range of intrusion types, not just DDoS or botnets
- Incorporates fog computing, unlike cloud-based methods
- Offers an end-to-end solution from IoT to cloud

So, the proposed technique offers explainability, flexibility, and a comprehensive fog computing architecture for intrusion detection that DDoS attack detection with integrated explainability.

The key contributions of our work are:

1. A hybrid detection module embedded in fog nodes that extracts statistical features from network traffic and utilizes a KNN classifier to identify anomalies and attacks with 99.95% accuracy.

2. A comprehensive fog computing infrastructure spanning from IoT devices to cloud that enables low latency detection compared to cloud-only approaches.

3. Integration of LIME technique to generate sample-specific explanations

Technique	Environment	Explainability	Wide Intrusion Detection	Fog Computing
Proposed DIHIF-LIME	IoT/Fog	Yes (LIME)	Yes	Yes
Samy et al. [5]	Cloud	No	No (Only DDoS)	No
Lawal et al. [6]	Fog	No	No (Only DDoS)	Partial
Gudla et al. [7]	Fog/IoT	No	No (Only DDoS)	Yes
Khempetch and Wutti-dittachotti [8]	IoT/Fog	No	No (Only DDoS)	Yes
Meidan et al. [9]	IoT	No	No (Only bot-nets)	No
Vinayakumar et al. [10]	General	No	Yes	No

Table 2: Comparison of current existing approaches with proposed approach

5. CONCLUSION

In this paper, we have presented a novel deep learning-driven fog computing architecture for real-time intrusion and

that increase model transparency and trustworthiness.

4. Thorough evaluation on the

CICIDS2017 dataset demonstrating accurate and efficient DDoS attack detection capabilities.

Our proposed intelligent fog-based intrusion detection framework provides an end-to-end solution from IoT devices to cloud that offers real-time, scalable and interpretable security for IoT networks. The hybrid machine learning driven architecture outperforms existing methods as evidenced by the high accuracy, precision, recall and F1-scores achieved. With integrated explainability, our system promotes transparency and trust.

In future work, we aim to expand the detection capabilities to identify a wider range of emerging cyber threats. Additionally, we intend to deploy and evaluate the proposed solution in real-world IoT and fog computing infrastructures. The framework can be enhanced with automated mitigation responses. We believe our work is an important step toward securing the ubiquitous IoT devices and networks of the future.

REFERENCES

[1] H. Sedjelmaci, S. M. Senouci, and M. Al-Bahri, "A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4552–4564, 2020.

[2] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[3] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in Neural Information Processing Systems*, vol. 28, pp. 1135–1143, 2015.

[4] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies, pp. 21–26, 2016.

[5] A. Samy, H. Yu, and H. Zhang, "Fog-based attack detection framework for internet of things using deep learning," *IEEE Access*, vol. 8, pp. 74571–74585, 2020.

[6] M. A. Lawal, R. A. Shaikh, and S. R. Hassan, "A DDoS attack mitigation framework for IoT networks using fog computing," *Procedia Computer Science*, vol. 182, pp. 13–20, 2021.

[7] S. Sadhwani, B. Manibalan, R. Muthalagu, and P. Pawar, "A Lightweight Model for DDoS Attack Detection Using Machine Learning Techniques," *Applied Sciences*, vol. 13, no. 17, pp. 9937, 2023.

[8] T. Khempetch and P. Wuttidittachotti, "DDoS attack detection using deep learning," IAES *International Journal of Artificial Intelligence*, vol. 10, no. 2, pp. 382, 2021.

[9] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiot network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[10] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.
[11] A. Hazra, P. Rana, M. Adhikari, and T. Amgoth, "Fog computing for next-generation internet of things: fundamental, state-of-the-art and research challenges," *Computer Science Review*, vol. 48, pp. 100549, 2023.
[12] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: A review of current applications and security solutions," *Journal of Cloud Computing*, vol. 6, no. 1, pp. 1–22, 2017.
[13] M. Azeem, A. Ullah, H. Ashraf, N. Z. Jhanjhi, M. Humayun, S. Aljahdali, and T. A. Tabbakh, "Fog-oriented secure and lightweight data aggregation in IoMT," *IEEE Access*, vol. 9, pp. 111072–111082, 2021.
[14] F. Khan, M. A. Khan, S. Abbas, A. Athar, S. Y. Siddiqui, A. H. Khan, and M. Hussain, "Cloud-based breast cancer prediction empowered with soft computing approaches," *Journal of Healthcare Engineering*, vol. 2020, pp. 2020.
[15] M. Waqas, K. Kumar, U. Saeed, M. M. Rind, A. A. Shaikh, F. Hussain, A. Rai, and A. Q. Qazi, "Botnet attack detection in Internet of Things devices over cloud environment via machine learning," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 4, pp. e6662, 2022.
[16] Li, Distributed denial of service attack (DDoS) definition, 2022.
[17] K. N. Mallikarjunan, K. Muthupriya, and S. M. Shalinie, "A survey of distributed denial of service attack," 2016 10th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, pp. 1–6, 2016,
[18] D. L. Marchi and L. Mitchell, "Hands-On Neural Networks: Learn how to build and train your first neural network model using Python," *Packt Publishing*, 2019.
[19] J. Zhao, X. Han, M. Ouyang, and A. F. Burke, "Specialized deep neural networks for battery health prognostics: Opportunities and challenges," *Journal of Energy Chemistry*, 2023.
[20] J. Farooq and M. A. Bazaz, "Hybrid Deep Neural Network for Data-Driven Missile Guidance with Maneuvering Target," *Defence Science Journal*, vol. 73, no. 5, pp. 602–611, 2023.
[21] N. Saha, A. Swetapadma, and M. Mondal, "A Brief Review on Artificial Neural Network: Network Structures and Applications," 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 1, pp. 1974–1979, 2023.
[22] W. Na, K. Liu, J. Zhang, D. Jin, H. Xie, and W. Zhang, "An Efficient Batch-Adjustment Algorithm for Artificial Neural Network Structure Adaptation and Applications to Microwave Modeling," *IEEE Microwave and Wireless Technology Letters*, 2023.
[23] M. Ehteram and E. Ghanbari-Adivi, "Self-attention (SA) temporal convolutional network (SATCN)-long short-term memory neural network (SATCN-LSTM): an advanced python code for predicting groundwater level," *Environmental Science and Pollution*

Research, 2023.
[24] A. Ghoroghi, I. Petri, Y. Rezgui, and A. Alzahrani, "A deep learning approach to predict and optimise energy in fish processing industries," *Renewable and Sustainable Energy Reviews*, vol. 186, pp. 113653, 2023.
[25] S. Akhtar, M. Adeel, M. Iqbal, A. Namoun, A. Tufail, and K. H. Kim, "Deep learning methods utilization in electric power systems," *Energy Reports*, vol. 10, pp. 2138–2151, 2023.
[26] M. Moocarme, M. Abdolahnejad, and R. Bhagwat, "The Deep Learning with Keras Workshop: An Interactive Approach to Understanding Deep Learning with Keras," 2nd Edition, Packt Publishing, 2020.
[27] C. D. McDermott, F. Majdani, and A. V. Petrovski, "Botnet Detection in the Internet of Things using Deep Learning Approaches," 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, pp. 1–8, 2018.
[28] C. C. Ugwu, O. O. Obe, O. S. Popoola, and A. O. Adetunmbi, "A distributed denial of service attack detection system using long short term memory with Singular Value Decomposition," Proceedings of the 2020 IEEE 2nd International Conference on Cyberspace, CYBER NIGERIA, pp. 112–118, 2021.
[29] M. D. Hossain, H. Fall, and K. Kadobayashi, "LSTM-based Network Attack Detection: Performance Comparison by Hyper-parameter Values Tuning," 2020 6th IEEE International Conference on Edge Computing and Scalable Cloud, CSCloud-EdgeCom 2020, pp. 62–69.
[30] Y. Gormez, Z. Aydin, R. Karademir, and V. C. Gungor, "A deep learning approach with Bayesian optimization and ensemble classifiers for detecting denial of service attacks," *International Journal of Communication Systems*, vol. 33, no. 11, pp. 1–16, 2020.
[31] M. Rusyaidi, S. Jaf, and Z. Ibrahim, "Machine learning method in detecting a distributed of service (DDoS): A systematic literature review," *AIP Conference Proceedings*, vol. 2643, no. 1, 2023.

International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2024.0802198

Vol. 8 issue 2 Apr-Jun 2024

Strategies to prevent Students academics crimes through Cybersecurity

Maiha Kamal[1], Ashraf Iqbal[1], Erum Tatheer[2] and Ghulam Abbas[3]

[1]Department of Mass Communication, Government College University, Faisalabad, Punjab, Pakistan
[2]Lahore Leads University, Lahore Punjab, Pakistan
[3]Department of Animal Production Riphah International University, Lahore, Punjab, Pakistan
Corresponding author: ghulamabbas_hashmi@yahoo.com

Received: May 07, 2024; **Accepted:** May 18, 2024; **Published:** June 14, 2024

ABSTRACT

In the present era, media greatly influences people's lives and has overcome their thoughts and perceptions with its contents. Along with its positive use rise of social media crimes amongst the students of universities is a big issue these days that demands instant attention. By recognizing the contributing issues and applying broad level cybersecurity actions, educational institutes can create a safe digital atmosphere for students. Nowadays social media has set off a vital segment of our daily lives and every person especially university students is depending on social media sites for the sake of information. In this study "Relationship between Social Media Usage on the Academic Performance of the Students," the main focus is to know how frequently and which type of media content is being used by university students and how it influences their academic performance. Whether their GPA and CGPA have improved or reduced. The quantitative survey method was used to collect the data and a systematic sampling technique was used for the selection of correspondents. The frequency distribution regarding the statement, "Excessive usage of social media affects the academic performance of university students". 12.0% of respondents strongly agreed, 44.7% of students agreed, 26.7% of respondents neutral, 14.3% disagreed, and 2.3% strongly disagreed with the statement. The frequency distribution regarding the statement, "Social

media sites detract the university students from academic goals", 18.7% of respondents strongly agreed, 44.7% agreed, 25.7% neutral, 8.7% disagreed, and 2.3% respondents are strongly agreed. However, collaborative efforts between universities, students, and cybersecurity professionals is essential in fighting the rising threat of social/digital media crimes and warranting the protection and comfort of student community. There is a dire need to emphasize cybersecurity education, developing a culture of cognizance and attentiveness adopting modern technologies to mitigate the risks related to negative use of social media.

Keywords: Social media, Students, Academic performance, cybersecurity, Influence

1. INTRODUCTION

In the present era, media greatly influences people's lives and has overcome their thoughts and perceptions with its contents. The media has been efficacious in attaining people's attention and fulfilling needs. The invention of the World Wide Web (www) has changed the transmitting data method. Since the launch of social media, these sites have grabbed the attention of millions of users who have become regular users of social networking sites [1;5]. Education is very important for every person far more than anything else but in the present era, youngsters are showing their interest in the usage of social media than their academic activities that cause to badly affect their academic performance. It is admitted that social media has brought people closer and provides a forum for information and discussions. On the other hand, after the arrival of cell phones and other ICT gadgets in the market, the usage of social media sites has increased in a dramatic way. Due to easy internet access, the learners are connected to these sites for 24 hours and consume most of their time messaging, images, video sharing, and watching and commenting. They use it during lectures and miss the most important points of a topic that is under discussion in the classroom and face difficulties to cover this gap later on causing pendency of their assignments and low grades in their academic career. Most of the researchers agreed that students who are continuously connected with SNS during their study time get low grades. Junco [2] emphasizes that the utilization of social media in chatting, comments, video uploading and sharing, and profile uploading during study period badly affect the overall percentage of students and crush their academic abilities as well.

According to Junco [3] mostly those students not paying their attention to the lecture delivered by their teacher, they remain engaged with SNS during class and hence miss their lecture and key points of the instructors and remain unable to perform their best usually falling a victim to low grads poor academic proficiency and career by converting them into poor and average students in their academic performance. Students of the new generation spend a big portion of their precious time on different social media sites during their study time, Due to this trend, university students have become less competitive, lazy, and careless towards their studies.

Social media is becoming an indispensable part of students' life these days putting opportunities as well as challenges [1; 2; 3; 4; 5]. Although it eases collaboration, communication, and access to precious information but excessive use of social media have devastating effect on academic performance of students. [1; 5; 22].

In contrast, Social media invites the learner to attach with other learners which can be most beneficial if they use it for their academic purpose. According to Paul et al. [4] state that the Internet is a well-known and common tool of communication for all its users, especially students. So, educational institutes and faculty members are progressively attached to social media and connecting their capable students to it for sharing their course material. Although Rosen et al. [6] state that the present era is the era of modern technology, students of the modern age are very active and clever. They effectively manage their academic and social activities simultaneously. They continue interacting with their friends, preparing for their exams and class activities simultaneously as well. Social media invites learners to attach with other learners which can be most beneficial if they use it for academic purposes [1; 5]. But a very rare number of students use this platform to gratify their academic purpose. The motive of this study was to investigate how social media is used by university students and how it affects their academics as well as social life. Whether their GPA and CGPA have improved or reduced due to the total amount of hours per day they consume on utilization of social media. The propagation of digital media has modernized communication amongst universities students. Yet, it has also raised various digital media crimes rate, causing significant dangers to students' well-being, privacy, and security. This study explores the positive and negative effect of increasing social media use amongst universities students, prevalence of social media crimes, factors contributing to digital crimes, and the cybersecurity actions needed to fight these cybercrimes amongst students at university level.

2. METHODOLOGY

According to the study's nature and need, the quantitative method and survey technique have been chosen to collect the data for analysis. There were300 hundred students between the age of 18-30, chosen as samples from various universities and disciplines. The questionnaire was used as a data collection tool for the sake of data collection which consisted of close-ended questions. The research sample was divided into different socio-demographic variables such as age, gender, education, family background, time consumption on social media, etc.

H1. Excessive usage of social media has a negative impact on the academic performance of university students and badly affects their GPA or CGPA.

H2. Excessive usage of social media by university students is cause to decrease their academic proficiency. These hypotheses are supported by the frequency table.

Analysis of Responses

The results of the statement **Students achieve their academic goals by using social media sites**. Find a significant difference between group 1st (*M*= 3.48, *SD*=1.008) and group 2nd (*M*= 3.69, *SD*= 0.988), *t* (298) =-1.637. The p-value is 0.103. The results of the statement **social media is very effective to be used in today's different learning process** find a significant difference between the two groups of GPA group 1st (*M*= 3.83, *SD*=1.02) and 2nd (*M*= 3.88, *SD*= 0.874), *t* (298) =-1.850. the p-vale is 0.065. Results of the statement **Excessive usage of social media affects the academic performance of university students** find a significant difference between the two groups of GPA group 1st (*M*= 3.465, *SD*=0.936) and group 2nd (*M*= 3.576, *SD*= 1.016), *t* (298) =-0.906 the p-value is 0.366. The results of the statement **The usage of social networking sites plays an important role in the preparation of academic assignments** find a significant difference between the two groups of GPA group 1st (*M*= 3.637, *SD*=0.890) and 2nd (*M*= 3.552, *SD*= 0.957), *t* (298) =0.723 the p-vale is 0.470. The results of the statement of **the usage of social networks badly affects the academic performance of university students.** find significant differences between the two groups of GPA 1st group (*M*= 3.581, *SD*=1.014) and 2nd (*M*= 3.458, *SD*= 0.866), *t* (298) =0.981 the p-vale is 0.

Table1: Categorization of respondents regarding to the purpose of the usage of SNS

Purpose	Frequency	percentage
Socialization & making new friends	74	24.7
Entertainment	137	45.7
Collaboration	47	15.7
To enhance knowledge	21	7.0
All of these	21	7.0
Total	**300**	**100.0**

Table 2: Categorization of respondents regarding the statements

Statement	f/%	SA	A	N	DA	SDA	Mean	St. d
Social media is very effective to be used in today's different learning processes.	f %	24 8.0	24 8.0	102 34.0	68 22.7	06 2.0	3.2267	.95481
Excessive usage of social media affects the academic performance of university students.	f %	36 12.0	134 44.7	80 26.7	43 14.3	07 2.3	3.4967	.95902
Social media sites detract university students from academic goals.	f %	56 18.7	134 44.7	77 25.7	26 8.7	07 2.3	3.6867	.95123
By using social media students feel more creative regarding their studies.	f %	31 10.3	135 45.0	82 27.3	45 15.0	05 2.3	3.4600	.94766
The usage of social networks badly affects the academic performance of university students.	f %	42 14.0	134 44.7	79 26.3	36 12.0	09 3.0	3.5467	.97519
The usage of social networking sites plays an important role in the preparation of an academic assignment.	f %	41 13.7	145 48.3	75 25.0	35 11.7	04 1.3	3.6133	.90916

3. DISCUSSION

The current study observes social media usage and its impacts on students' academic performance. The results of the study are in contrast with earlier research and the present study was held in the district of Faisalabad Pakistan. The outcome of the study is also based on time consumption and the academic percentage of the students. The findings of the present research show that in Pakistan, learners utilize a huge portion of their precious time on social networking sites on daily bases. For the sake of comparison, the opinion of those students who have got a 3.00 to 3.50 GPA and those who have got more than a 3.5 GPA in their semester regarding the statements "Students achieve their academic goals by using social media sites." "Social media is very effective to be used in today's different learning process. However, excessive usage of social media affects the academic performance of university students. Those students who agreed with this statement had got less than a 3.5 GPA. "Social media sites detract the university students from academic goals. Excessive use of social media has reported to significantly weaken the intention of students for academic activities. Researches have shown the poorer academic performance of students who spent more time on social media resultantly less time in study[10]. The continuous spending of time on social platforms has led to postponement of study work and poor time management resulting into badly affecting the grades of students. Social media platforms are continually engage the students by frequent notifications to stay update thus lead to distracting and interrupting study times preventing effective learning. Research studies indicated that involvement of students in multitasking activities on social media impair the cognitive functions and academic performance bitterly [15]. On the other hand, usage of social networking sites plays an important role in the preparation of academic assignments" Students who belonged to the first group of GPA agreed with the statement. Those students who got more than a 3.5 GPA were in favor of this statement. On the other hand, rising eases of accessing/sharing information on digital/social media has led an increased level of plagiarism issues in academic publications. Students have developed the habit of copy writing contents without proper citation and/or collaborative cheating (sharing chat or answers of exam questions) during exams and assignments [18] violating academic policies [16].

H1. Excessive usage of social media has negative impact on the academic performance of university students and badly affects their GPA or CGPA as supported by the frequency table. **H2.** Excessive usage of social media by university students is cause to decrease their academic proficiency. These hypotheses are supported by the frequency table. Although social media also provides access to precious information as well but, the credibility of this information is questionable. Students might rely on inaccurate/misleading sources of information for educational purposes, and such types of information may cause negative impact on their thinking

and academic [12; 30]. Moreover, prolonged use of digital/social media affect mental health of students leading to mental disorders like anxiety, sleep disturbances, and depression which severely affect concentration of students in study [17]. Another negative aspect of usage of social media by students is the way it has made easy for students to hire third party to take exams on their behalf or to complete their assignments [11] weakening the learning process and devalues the academic work.

Wise usage of social media and Preventing Academic Crimes

Educating the students the significance of academic integrity and potential losses of academic offenses is very essential to restore learning environment at university level. For this, particular courses, workshops, and seminars on adopting proper citation practices, knowing the risks of plagiarism, and the ethical/honest use of information may boost culture of honesty among students [13; 31]. Institutions must implement policies to monitor/regulate the usage of social platform in academic activities. This include restricting the access to social/digital media sites during lectures in class room and study hours or on universities premises, also implementing hard and tough policies to detect plagiarism through standard software like Turnitin by providing the access of teachers to this software to identify the copied contents of researcher students[8]. Helping the students to develop effective time management skills may lessen the negative influence of social media on education. Managing time management workshops, teaching techniques like Pomodoro Technique and planning tools can help students to balance the usage of social media and better understanding their academic responsibilities [7]. Likewise provision of vigorous mental health support may address the negative psychological effect of social media. Students must provide counseling services, and stress management workshops for mental well-being and to improve academic performance [9]. Training the students how to critically evaluate the information receiving on social media, knowing credibility of sources, relying on just credible information, and applying the received information to academic work can decrease the possibility of academic offenses [14; 33]. Furthermore, combined engaging and cooperative teaching methods may help to reduce the bad habit of students to distract their attention to inappropriate websites and/or games. This can be achieved by using collaborative learning, educational technologies, and interactive activities, and making classroom times more attractive and important to students [19].

Excessive use of social media by universities students and rising cybercrimes

Rising crimes related to excessive use of social media amongst university students involve a range of illegal activities such as identity theft, cyberbullying, hacking, online harassment, and phishing. Cyberbullying has come to be predominantly prevalent, with students

using social media platforms to frighten, threaten, or degrade their fellows [29]. Identity theft is also emergent concern, wherever individual's information is stolen and thereafter misused, often resulting to bitter emotional as well as financial penalties [24]. Hacking incidents, where illegal entree to social/digital media accounts results in data openings and privacy harms [36]. Moreover, phishing schemes (online scam that illegally targets consumers to get his/her personal information using fake famous sources) deceive students into disclosing their personal sensitive information, and virtual harassment cases contributes further to the growing concern of digital media crimes amongst universities students [25;34].

Factors Contributing to Cyber-Crimes

Numerous factors contribute in rising the social media crimes amongst universities students. The extensive use of digital media without suitable awareness of cybersecurity rehearses is a key factor (Taneja & Toombs, 2014). Most of the students are ignorant of the dangers related with online personal information sharing, making them easy victim for cybercriminals [27; 28]. The secrecy afforded by digital media also encourages persons to involve in hateful activities with no fear of instant consequences [23]. Moreover, the stressful and competitive environment of university can worsen such issues, as students may resort to harassment or cyberbullying to cope with social and academic pressure). Also, the absence of vigorous cybersecurity education/awareness programs in university further exaggerates this harmful situation [26].

Use of social media amongst universities students and Cybersecurity Measures to Combat Social Media Crimes

Researches have shown the profound effects of social/digital media crimes on mental health of university students experiencing serious psychological effects like anxiety, depression, even suicidal thinking [29]. Identity theft due to cybercrime may lead to monetary losses and extended loss to credit scores, upsetting students' future career [24]. Data breaks and hacking cases can harm students' privacy and loss of trust in university and social platforms [36] as well as deteriorations in academic performance [34]. Besides, the general environment of campus become toxic, upsetting the well-being of the community.

A comprehensive cybersecurity approach needs to be implemented to address the increasing digital crimes amongst universities students. University authorities should highlight cybersecurity education, including it as part of course work to raise consciousness about safe online practices and digital security and hygiene (Taneja & Toombs, 2014). For this, regular seminars and workshops on cybersecurity can prepare students to recognize and avoiding potential threats [26]. Applying strict security protocols i.e. two-factor authentication may enhance the social media accounts security and decrease the risk of identity theft as well as hacking risks [20]. Establishment of clear policies/procedures to report cyberbullying and addressing it is

essential to provide sound support to affected students [23]. Teamwork with experts of cybersecurity can further strengthen the defenses against cybercrimes [14]. Advanced artificial intelligence tools and machine learning algorithms should be used to identify and stop cyber bullying in real-time [35]. Use of advanced technologies can inspect online behavior patterns to identify phishing attempts or cyberbullying thus can alert authorities before occurrence of harms [21]. Encryption technologies (AES, DES, RSA, teofish, triple DES etc.) can protect sensitive data by maintaining privacy [32]. Furthermore, universities should control blockchain technology to generate tamper-proof secure systems to manage student data and IDs [35; 36].

4. CONCLUSION AND RECOMMENDATIONS

In this research study researcher has explored the impacts of social media usage and its relation to the learners' academic performance. The present study calculates that the majority of the students were male. Numbers of the students of respondents were between the ages of the age of 18 to 24 who used social media. They belonged to the city area of Faisalabad. All the respondents of the study had internet access and all they had social media accounts on different social networking sites and they connected with it on a regular basis. The majority of the students use social media on daily bases. They consume 2-3 hours on social networking sites. Students were uses SNS for different purposes. The prominent factor for which students use social media is entertainment; very few students use it for the purpose of study and to enhance their knowledge [1; 5].

The present study points out a solution that may help overcome the problem caused by excessive usage of internet-based applications. The management of universities should restrict the students' usage of cell phones, tablets, and laptops during class and punish those who violate this rule. Furthermore, IT departments of universities should block Facebook and other social networking sites on the premises of the university. Students have access only to sites that help them fulfill their academic needs.

REFERENCES

[1] F. Iske, "Comparison of social network in education between North and South Cyprus," Proceedings of Social Behavior Sciences, vol. 210-219, 2013.

[2] R. Junco, "Too much face and not enough books: The relationship between multiple indices of Facebook use and academic performance," *Computers in Human Behavior*, vol. 28, pp. 187-198, 2012.

[3] R. Junco, G. Helbergert, and E. Loken, "The effect of Twitter on college student engagement and grades," *Journal of Computer Assisted Learning*, vol. 27, pp. 119-132, 2011.

[4] J. Paul, H. Baker, and J. Cochran, "Effect of online social networking on student academic performance," *Elsevier*, vol. 1, pp. 2118-2119, 2012.

[5] F. Iske, "Comparison of social network in education between North and South Cyprus," Proceedings of Social Behavior Sciences, vol. 210-219, 2013.

[6] L. Rosen, L. Carrier, and N. Cheever, "Facebook and texting made me do it: Media induced task-switching while studying," ELSEVIER, vol. 948-950; pp. 955, 2013.
[7] B. J. Claessens, W. van Eerde, C. G. Rutte, and R. A. Roe, "A review of the time management literature," *Personnel Review*, vol. 36, no. 2, pp. 255-276, 2007.
[8] C. Ellis, I. M. Zucker, and D. M. Randall, "The influence of perceived organizational support on the attitudes of faculty members toward academic dishonesty," *The Journal of Higher Education*, vol. 89, no. 1, pp. 33-60, 2018.
[9] C. Huang and Y. Zhou, "The effects of different kinds of social media use on college students' emotional well-being," *Journal of Educational Computing Research*, vol. 57, no. 2, pp. 331-345, 2019.
[10] R. Junco, "Too much face and not enough books: The relationship between multiple indices of Facebook use and academic performance," Computers in Human Behavior, vol. 28, no. 1, pp. 187-198, 2012.
[11] T. Lancaster and R. Clarke, "Contract cheating: The outsourcing of assessed student work," Handbook of Academic Integrity, pp. 639-654, 2016.
[12] S. Livingstone and D. R. Brake, "On the rapid rise of social networking sites: New findings and policy implications," *Children & Society*, vol. 24, no. 1, pp. 75-83, 2010.
[13] R. Macdonald and J. Carroll, "Plagiarism—a complex issue requiring a holistic institutional approach," *Assessment & Evaluation in Higher Education*, vol. 31, no. 2, pp. 233-245, 2006.
[14] M. J. Metzger and A. J. Flanagin, "Credibility and trust of information in online environments: The use of cognitive heuristics," *Journal of Pragmatics*, vol. 59, pp. 210-220, 2013.
[15] L. D. Rosen, "The distracted mind: Ancient brains in a high-tech world," Psychology Today, 2013.
[16] M. Stoner, "Social media in higher education: They said it couldn't be done," EDUCAUSE Review, vol. 48, no. 2, pp. 68-69, 2013.
[17] A. Vannucci, K. M. Flannery, and C. M. Ohannessian, "Social media use and anxiety in emerging adults," Journal of Affective Disorders, vol. 207, pp. 163-166, 2017.
[18] J. Walker, "Measuring plagiarism: Researching what students do, not what they say they do," Studies in Higher Education, vol. 35, no. 1, pp. 41-59, 2010.
[19] M. Prince, "Does active learning work? A review of the research," *Journal of Engineering Education*, vol. 93, no. 3, pp. 223-231, 2004.
[20] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," 2012 IEEE Symposium on Security and Privacy, pp. 553-567, 2012.
[21] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," *The Social Mobile Web*, vol. 11, no. 2, pp. 11-17, 2012.
[22] C. B. Fried, "In-class laptop use and its effects on student learning," *Computers & Education*, vol. 50, no. 3, pp. 906-914, 2008.

[23] S. Hinduja and J. W. Patchin, "Cyberbullying: An exploratory analysis of factors related to offending and victimization," *Deviant Behavior*, vol. 29, no. 2, pp. 129-156, 2008.
[24] S. B. Hoar, "Identity theft: The crime of the new millennium," Hastings LJ, vol. 52, pp. 583, 2001.
[25] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer, "Social phishing," Communications of the ACM, vol. 50, no. 10, pp. 94-100, 2007.
[26] L. M. Jones, K. J. Mitchell, and D. Finkelhor, "Online harassment in context: Trends from three Youth Internet Safety Surveys (2000, 2005, 2010)," *Psychology of Violence*, vol. 3, no. 1, pp. 53, 2013.
[27] A. C. Karpinski, P. A. Kirschner, I. Ozer, J. A. Mellott, and P. Ochwo, "An exploration of social networking site use, multitasking, and academic performance among United States and European university students," *Computers in Human Behavior*, vol. 29, no. 3, pp. 1182-1192, 2013.
[28] Kaspersky, "Kaspersky Lab Cybersecurity Index: Internet users starting to worry more about privacy," 2016.
[29] R. M. Kowalski, G. W. Giumetti, A. N. Schroeder, and M. R. Lattanner, "Bullying in the digital age: A critical review and meta-analysis of cyberbullying research among youth," *Psychological Bulletin*, vol. 140, no. 4, pp. 1073, 2014.
[30] S. Livingstone and E. Helsper, "Parental mediation of children's internet use," *Journal of Broadcasting and Electronic Media*, vol. 52, no. 4, pp. 581-599, 2008.
[31] R. Macdonald and J. Carroll, "Plagiarism—a complex issue requiring a holistic institutional approach," Assessment & Evaluation in Higher Education, vol. 31, no. 2, pp. 233-245, 2006.
[32] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, "Handbook of applied cryptography," CRC Press, 1996.
[33] M. J. Metzger and A. J. Flanagin, "Credibility and trust of information in online environments: The use of cognitive heuristics," *Journal of Pragmatics*, vol. 59, pp. 210-220, 2013.
[34] J. W. Patchin and S. Hinduja, "Bullies move beyond the schoolyard: A preliminary look at cyberbullying," *Youth Violence and Juvenile Justice*, vol. 4, no. 2, pp. 148-169, 2006.
[35] D. Sculley, G. Wachman, and B. Chandramouli, "Detecting adversarial advertisement inserts in social network streams," Proceedings of the 20th International Conference on World Wide Web, pp. 277-286, 2011.
[36] A. Smith, "Older adults and technology use: Adoption is increasing, but many seniors remain isolated from digital life," *Pew Research Center*, 2014.

Editorial Policy and Guidelines for Authors

IJECI is an open access, peer reviewed quarterly Journal published by LGU. The Journal publishes original research articles and high-quality review papers covering all aspects of crime investigation.

The following note set out some general editorial principles. All queries regarding publications should be addressed to editor at email IJECI@lgu.edu.pk. The document must be in word format, other format like pdf or any other shall not be accepted.
The format of paper should be as follows:

- Title of the study (center aligned, font size 14)
- Full name of author(s) (center aligned, font size 10)
- Name of Department
- Name of Institution
- Corresponding author email address.
- Abstract
- Keywords
- Introduction
- Literature Review
- Theoretical Model/Framework and Methodology
- Data analysis/Implementation/Simulation
- Results/ Discussion and Conclusion
- References.

Heading and sub-heading should be differentiated by numbering sequences like, 1. HEADING (Bold, Capitals) 1.1 Subheading (Italic, bold) etc. The article must be typed in Times New Roman with 12 font size 1.5 space, and should have margin 1 inches on the left and right. Table must have standard caption at the top while figures below with. Figure and table should be in continues numbering. Citation must be in according to the IEEE style.

LAHORE GARRISON UNIVERSITY

Lahore Garrison University has been established to achieve the goal of excellence and quality education in minimum possible time. Lahore Garrison University in the Punjab metropolis city of Lahore is an important milestone in the history of higher education in Pakistan. In order to meet the global challenges, it is necessary to touch the highest literacy rates while producing skillful and productive graduates in all fields of knowledge.

VISION

Our vision is to prepare a generation that can take the lead and put this nation on the path to progress and prosperity through applying their knowledge, skills and dedication. We are committed to help individuals and organizations in discovering their God-gifted potentials to achieve ultimate success actualizing the highest standards of efficiency, effectiveness, excellence, equity, trusteeship and sustainable development of global human society.

MISSION

At present, LGU is running Undergraduate, Graduate, Masters, M.Phil. and Ph.D. programs in various disciplines. Our mission is to serve the society by equipping the upcoming generations with valuable knowledge and latest professional skills through education and research. We also aim to evolve new realities and foresight by unfolding new possibilities. We intend to promote the ethical, cultural and human values in our participants to make them educated and civilized members of society.

Contact: For all inquiries, regarding call for papers, submission of research articles and correspondence, kindly contact at this address:
Sector C, DHA Phase-VI Lahore, Pakistan
Phone: +92- 042-37181823
Email: ijeci@lgu.edu.pk

Published by: Digital Forensics Research and Service Center, Lahore Garrison University