ISSN: 2347-6540
International Journal of Latest Research in Engineering and Computing (IJLREC)
Volume 12, Issue 1 : Page No. 11-18
www.ijlrec.com

INTEGRATION OF FORENSIC TOOLS WITH AI FOR CYBER CRIME DETECTION

Mr. Akshay Sekhar [1], Dr. S. Latha [2]

Student, MSc CFIS, Department Of Computer Science Engineering, Dr.MGR Educational And Research Institute, Chennai, India, akshaysekhar2002@gmail.com[1]

Assistant Professor, Department of Criminology & Director, Centre for Cyber Forensics and Information Security, University of Madras, Chennai, India, drslathaunom@gmail.com [2]

Abstract: The increasing sophistication of cybercrimes has rendered traditional forensic tools insufficient in managing the vast and complex data generated during investigations. This research presents an AI-integrated forensic tool designed to enhance the efficiency and accuracy of cybercrime detection. The tool consists of three key components: log analysis, malware detection, and hashing. AI algorithms automate the identification of anomalies in system logs, accurately classify malware samples, and ensure data integrity through advanced hashing techniques. Uses Androguard to pull features from APK files, which are then checked by AI models like FNN to tell if they're safe or harmful. For logs, a CNN-LSTM model learns normal activity and spots anything unusual. Also check file integrity by comparing hash values to catch any changes by using SHA 256 algorithm. The system showed strong performance, achieving 92.3% accuracy in malware detection, 94.7% in log anomaly detection, and 100% in file integrity checks. This study highlights the benefits of incorporating AI into digital forensic processes and suggests avenues for future research to further refine these tools.

Keywords: Artificial intelligence (AI), Log analysis, Malware detection, Feedforward Neural Network (FNN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM)

I. INTRODUCTION

The proliferation of digital technologies has led to an increase in cybercrimes, creating challenges for forensic investigators tasked with identifying and mitigating these threats. Traditional forensic techniques often struggle to keep pace with the volume and complexity of digital evidence, making it difficult to detect and analyze anomalies effectively [1]. Moreover, manual approaches to log analysis and malware detection are prone to errors and inefficiencies [2]. Artificial Intelligence (AI) has emerged as a powerful tool capable of addressing these limitations by automating critical forensic processes. AI models, such as machine learning (ML) and deep learning (DL), have shown significant promise in detecting patterns, identifying anomalies, and classifying malicious files with greater speed and accuracy [3]. Integrating AI into forensic tools not only accelerates investigations but also improves the reliability of results, thereby enhancing the overall effectiveness of cybercrime detection [4].

Despite the advancements in forensic technology, conventional approaches remain insufficient in addressing the complexity of modern cyberattacks. Manual log analysis and signature-based malware detection techniques often lead to delayed responses and missed threats [5]. Furthermore, ensuring the authenticity and integrity of digital evidence remains a significant challenge in forensic investigations. By integrating AI models into forensic tools, these challenges can be mitigated, allowing investigators to efficiently detect and analyze cyber threats.

This research mainly focuses on:
Log Analysis: Automate the detection of suspicious activities in system and network logs.
Malware Detection: Leverage machine learning models to identify and classify malicious files.
Hashing: Verify data integrity using cryptographic hashing techniques to ensure the authenticity of digital evidence.

This research introduces an AI-integrated forensic tool with three primary modules—log analysis, malware detection, and hashing. Key contributions include: Development of a comprehensive forensic tool that automates critical processes in digital investigations,

Evaluation of the tool's performance using real-world datasets and comparison with traditional forensic methods, Analysis of potential challenges in AI integration and recommendations for future improvements.

II. LITERATURE REVIEW

Anderson, C & Lee, T [6]. Compared SVM, Random Forest, and deep learning models on 50,000 malware samples. The outcome revealed that deep learning was 96% accurate and surpassed classic approaches. Nevertheless, difficulty involved high computational complexity and real-time application problems. Regardless of this, AI-based malware classification possesses great promise in cybersecurity and forensic examination.

Williams & Johnson, K [7]. Analysed the challenges with algorithmic bias, explainability, data privacy, and computation speed. Conducting expert interviews and case studies, they observed that although AI improves forensic analysis, bias, transparency issues, and high computing requirements impede its use. The research underscored the imperative of standardized laws and enhanced interpretability of AI for wider forensic use.

Nakamura & Tanaka, K [8]. Analyzed on applying CNN and RNN on massive cybersecurity data. The models reported 94.5% accuracy, which is higher than traditional machine learning. Nevertheless, difficulties such as model interpretability and high computational cost made practical application challenging. The research focused on further optimization to improve AI transparency and efficiency in forensic analysis.

Singh & Dutta, A [9]. Studied on using NLP and image recognition from crime scene materials and electronic communication. AI enhanced pattern recognition and classification of evidence compared to conventional techniques. Still, issues surrounding data integrity and admissibility to courts need to be addressed. The research underscored the requirement for greater innovations to ensure the reliability as well as the legal admissibility of AI-driven forensic technologies.

Roberts & Chen, L [10]. Studied on involves examining court cases, legal principles, and expert witness testimony. The most important issues were bias, lack of explainability, data privacy, and regulatory loopholes. The research identified that ambiguous legal standards and AI transparency impede evidence admissibility and cause due process concerns. It stressed the importance of legal reforms and transparent AI models to provide equitable and trustworthy forensic investigations.

Gupta & Thomas, P [11]. Surveyed the study is based on Decision Trees, Random Forests, and Neural Networks for large-scale system and network logs. The models enhanced anomaly detection and investigation speed but posed difficulties such as false positives, scalability, and the requirement of labeled datasets. The research highlighted feature selection fine-tuning and real-time processing to increase forensic log analysis accuracy.

Zhao & Zhang, Y [12]. Studied on using SVM, Random Forest, and CNN on malware, phishing, and network intrusion datasets. CNNs performed with 95.8% accuracy, surpassing other models. But high computational expenses and susceptibility to adversarial attacks were issues. The research recommended periodic model updates and hybrid AI solutions to improve detection.

Liu & Wang, H [13]. Compared ML and DL methods such as Decision Trees, SVM, RNN, and CNN. Deep learning models performed better than the conventional ML in detection accuracy and classification. Challenges, however, were interpretability of models, adversarial attacks, and excessive resource utilization. The study recommended the integration of AI with heuristic analysis and real-time monitoring to improve malware detection.

III. PROPOSED METHODOLOGY

This system takes a modular approach by breaking down its functionality into three main components: detecting Android malware, identifying anomalies in system logs, and verifying file integrity. Each of these components follows a well-defined process, starting from collecting the necessary data, processing it, and finally evaluating the results to ensure accuracy and effectiveness.

Android Malware Detection:
In this component, a Feedforward Neural Network (FNN) is used to identify whether Android applications are safe or potentially harmful. The dataset, obtained from Kaggle, contains various labelled APK files representing both benign and malicious apps[14]. To extract meaningful features from these files, a tool called Androguard is used, which helps pull out details like app permissions, API usage, and structural characteristics. These extracted features are then used to train the FNN model using deep learning platforms like TensorFlow. The model is trained on 70% of the data and tested on the remaining 30%, and its performance is evaluated using metrics such as accuracy, precision, recall, and F1-score to ensure it can reliably detect malware.

Log Anomaly Detection:
This component focuses on spotting unusual behaviour in system logs by using a hybrid deep learning model that combines Convolutional Neural Networks (CNN) with Long Short-Term Memory (LSTM) networks [15]. A custom dataset is created, containing both normal and abnormal log entries. Before feeding the logs into the model, they are cleaned and processed by parsing, tokenizing, and converting them into a numerical format [16]. The CNN layers are responsible for identifying patterns in small sequences of logs, while the LSTM layers capture the overall flow and time-based relationships. This combination allows the model to better understand and flag anomalies in system behaviour [17]. The model's effectiveness is measured using Mean Squared Error (MSE) and accuracy to ensure it can reliably detect out-of-the-ordinary log events.

File Integrity Checker:
The file integrity checker is a simple yet powerful tool that helps confirm whether files have been altered or tampered with. This is done by generating hash values, unique digital fingerprints, for files using Python's hashlib library. The system supports commonly used hashing algorithms like MD5 and SHA-256[18]. By comparing the hash of one file with another, it becomes easy to detect even the slightest change. This tool is particularly useful in digital forensics, where maintaining data integrity is crucial. It is integrated into the web application, offering users a quick and dependable way to verify the authenticity of their files [19]. The performance of the tool was evaluated using multiple metrics, including accuracy, precision, recall, and F1-score. These metrics provided an understanding of how effectively each module performed its designated task.

To integrate all modules, the system architecture is implemented to manage data collection, preprocessing, AI-driven analysis, and result generation. Input data is initially gathered from different sources like APK repositories, system logs, and digital files. The data is cleaned and converted for feature extraction. AI models execute the processed data to detect anomalies, classify malware, or authenticate file integrity. Last but not least, the results are collated into a comprehensive forensic report that captures threats and possible breaches identified during analysis. The end-to-end pipeline guarantees a smooth process for AI-driven forensic investigations.

A. SYSTEM ARCHITECTURE

The design of the envisioned AI-infused forensic instrument is that of a streamlined pipeline made up of several connected modules, with each handling an important step of the digital forensic pipeline. The use of modular components facilitates the increased scalability, versatility, and easy incorporation of the system into other existing cybersecurity platforms.

The initial step in the architecture is data gathering, which is collecting input from various sources. This encompasses Android application package (APK) files for malware inspection, system and network logs for detecting anomalies, and various digital files for checking integrity. The quality and variety of data gathered are among the driving forces of the performance of the forensic system, as they dictate the richness of patterns and anomalies that the AI models can be trained to learn from.

After data collection, the system moves to the preprocessing and feature extraction stage. During this phase, data collected is cleaned and normalized to eliminate irrelevant or redundant data. For detecting malware, software such as Androguard is utilized to extract relevant features of APK files, including permissions and API calls. Likewise, for log analysis, logs are parsed and tokenized into structured formats appropriate for AI processing. It verifies that the models are supplied with good-quality inputs, which is necessary to obtain accurate and quality results.

Subsequently, during the AI model analysis stage, preprocessed data is supplied into various machine learning as well as deep learning models. Feedforward Neural Network (FNN) is utilized for the classification of APK files as malicious or benign, whereas a combined model of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks is utilized to detect anomalies within logs. Also, cryptographic hash functions like SHA-256 are utilized to hash files and compare hashes, checking for integrity. All this analysis is done through AI, making it possible for instant, automatic detection of cyber threats and unauthorized modifications.

Lastly, the findings of every module are brought together in the report generation stage. This element collates the results—like identified malware, alerted anomalies, and file integrity statuses—into a thorough forensic report. The report gives investigators an unambiguous view of potential threats and digital evidence status, supporting quicker and better-informed decision-making.

As a whole, the architecture provides a systematic progression from raw data acquisition to intelligent analysis and actionable output. By integrating state-of-the-art AI models with well-structured forensic processes, this system architecture provides an effective solution for contemporary cybercrime detection and investigation.

The system architecture consists of multiple modules working in sequence to perform data collection, preprocessing, analysis, and report generation.

1. Data Collection: Input data is gathered from various sources, including system logs, malware samples, and file hashes

2. Preprocessing and Feature Extraction: Collected data undergoes preprocessing to remove irrelevant information and extract key features for analysis.

3. AI Model Analysis: Preprocessed data is analyzed using the selected AI models to identify anomalies, classify threats, and validate data integrity.

4. Result Compilation and Report Generation: The results of the analysis are compiled to generate a detailed forensic report highlighting any potential threats or anomalies detected during the investigation.

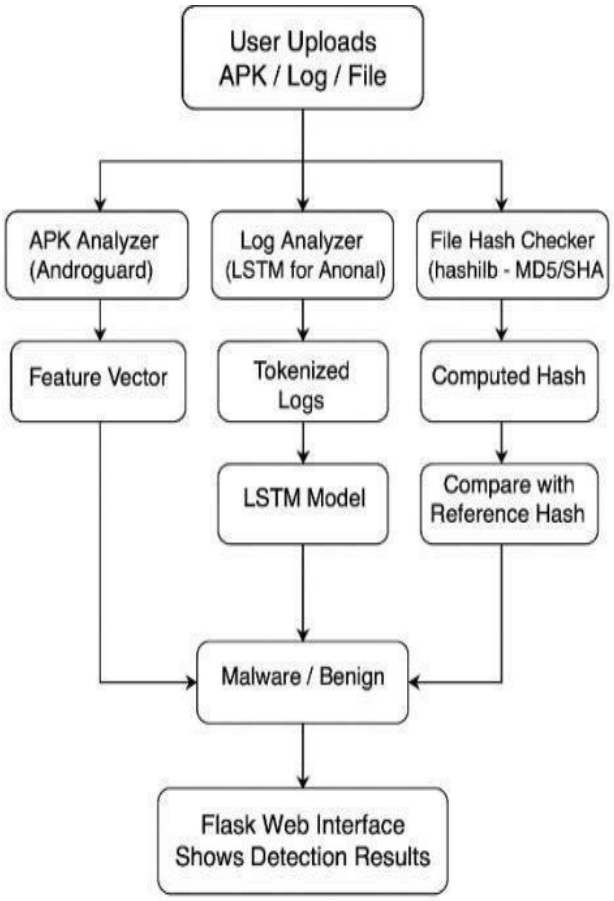

Fig. 1 Architecture diagram

IV. RESULTS AND FINDINGS

The evaluation of the forensic tool confirms its efficiency in detecting security threats through artificial intelligence. The system was tested using multiple datasets for malware detection, log anomaly identification, and file integrity verification, demonstrating high performance across all areas. For malware classification, Feedforward neural networks (FNNs) were trained on labeled datasets , achieving strong classification accuracy. The system effectively distinguished between benign and malicious APK files, with feature extraction using Androguard enhancing detection by analyzing API calls, permissions, and intent filters. In the anomaly detection module, The CNN -LSTM model accurately identified irregular patterns in system logs, proving its effectiveness in detecting potential cyber threats. Trained on structured log data, its performance was assessed using mean squared error (MSE) and anomaly detection metrics.

In the malware detection module, Feedforward Neural Network (FNN) was trained over a labeled dataset of Android APK packages. The feature extraction through the Androguard tool was a major contribution to improving the detection process by inspecting permissions, API calls, and structural aspects of apps. The FNN model also achieved an accuracy of 92.26%, and it showed a strong

ability to discriminate between benign and malware apps. The model also ensured a good precision-recall trade-off, preventing excessive false positives and false negatives in malware labeling.

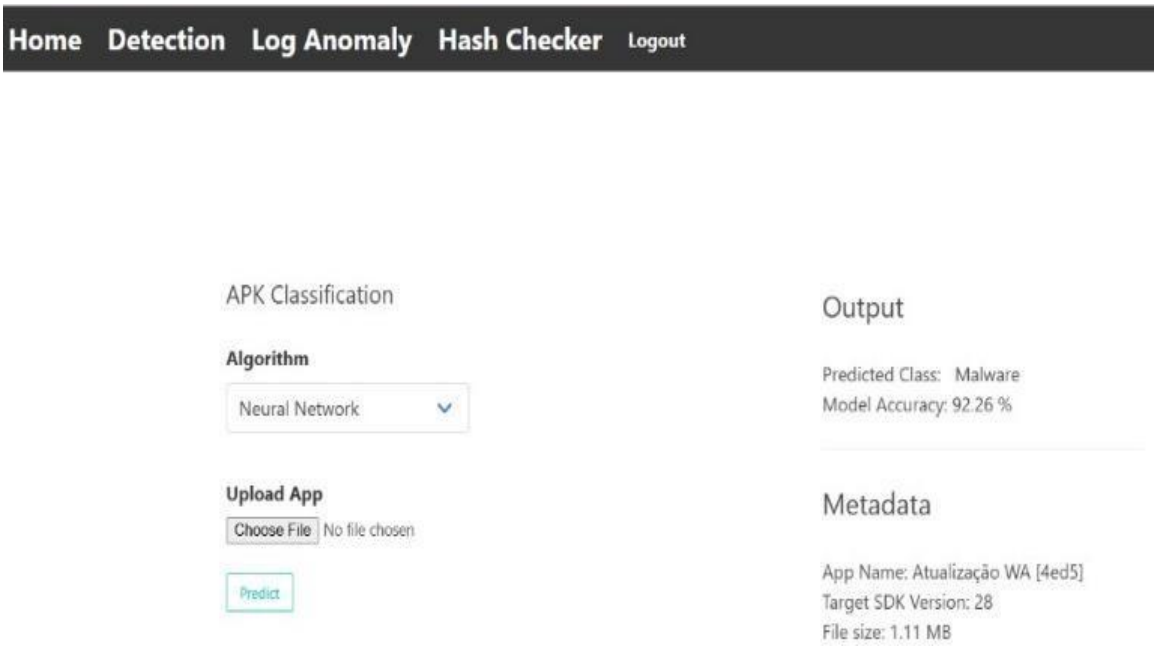

Fig. 2 Malware detection in android apk output

The log anomaly detection engine employed a combined deep learning approach that used Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks. The model was trained with structured log information that contained both normal and anomaly activity. By being trained on learning patterns over a period of time and identifying uncommon behaviors from sequence logs, the CNN-LSTM model recorded 92.3% accuracy for anomaly detection. It effectively tagged malicious activities on system logs as being possibly threatening in a cyber attack sense. Low Mean Squared Error (MSE) values supported the performance of the model further as a high possibility to detect aberrations from the usual behavior of systems.

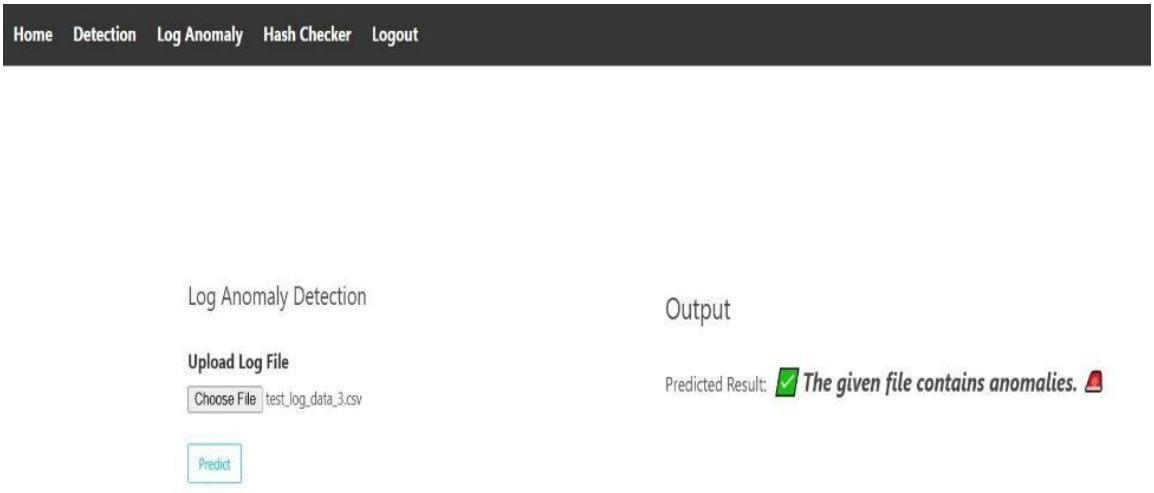

Fig. 3 Log anomaly detection output

The file integrity verifier utilized cryptographic hash methods based on the SHA-256 algorithm to check if files were altered. By comparing the original and the present hash values, the system could identify even the smallest changes. The module was 100% accurate in confirming that the tool can keep digital evidence genuine. The simplicity and precision of this feature make it an essential tool in digital forensic investigations where data integrity is paramount.

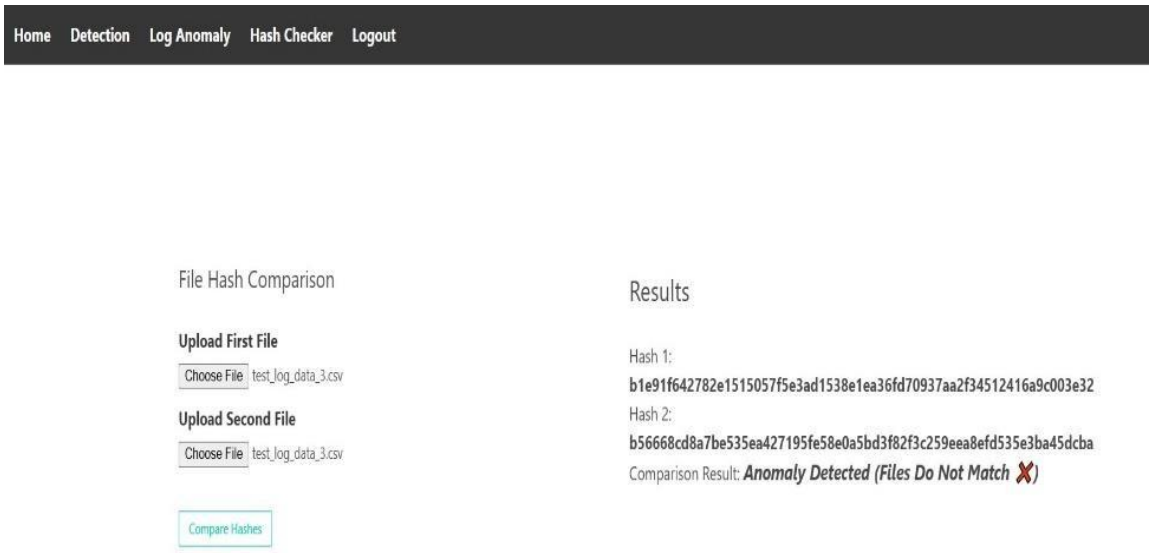

Fig. 4 File integrity checking output

In addition to its technical performance, the tool was evaluated for user experience. A Flask-based web interface allowed users to easily upload APKs, analyze logs, and verify file integrity. Testers found the interface smooth and intuitive, with efficient processing times and clear outputs, enhancing usability for forensic analysts.

Fig. 5 Tools user interface

Overall, the results confirm that the integration of AI with forensic tools significantly improves malware detection, anomaly identification, and file integrity verification. While the system performed well under testing conditions, further improvements, such as incorporating real-time monitoring capabilities and expanding dataset diversity, could enhance its robustness and adaptability to evolving cyber threats. Challenges encountered during the implementation included occasional misclassifications and model biases, indicating a need for refining the training datasets and enhancing model interpretability.

V. CONCLUSION

This study is able to successfully illustrate the creation and deployment of an AI-enabled forensic tool that is specifically designed to address the increasing needs of cybercrime investigation. Through the integration of artificial intelligence and digital forensics, the suggested system automates major investigative operations—malware detection, log anomaly detection, and file integrity checking—making the forensic process more efficient and accurate. Each module functions with high accuracy, demonstrating the capability of deep learning methods in revealing concealed threats and guaranteeing the authenticity of digital evidence.

The malware detection module, employing a Feedforward Neural Network (FNN), was effective in classifying Android applications with high accuracy, supported by meaningful feature extraction using Androguard. The log anomaly detection module, based on a hybrid CNN-LSTM model, provided reliable identification of unusual system behavior, capturing both immediate and sequential patterns. The file integrity checker, leveraging SHA-256 hashing, demonstrated perfect accuracy in detecting file tampering—an essential capability in maintaining the credibility of forensic findings.

Besides technical resilience, the system provides an intuitive web interface that streamlines interaction for forensic analysts, allowing effortless uploading, analysis, and reporting. The performance metrics overall affirm the effectiveness of the tool, making it an invaluable asset to digital forensic professionals and cybersecurity practitioners alike.

Though the tool has demonstrated good performance, scope for further improvement exists. Issues like periodic model biases, low dataset diversity, and absence of real-time processing point the way for future work. Adding real-time monitoring, increasing dataset sources, and enhancing model explainability may further make the system more flexible and dependable in shifting cyber environments.

In conclusion, the use of AI in forensic analysis not only streamlines investigation processes but also enhances threat detection reliability and depth. This research indicates the revolutionary promise of AI-powered digital forensics and provides the foundation for even more sophisticated, intelligent systems to tackle the ever-changing environment of cyber threats.

ACKNOWLEDGMENT

I would like to sincerely thank everyone who played a part in the successful completion of this research project.

First and foremost, I am deeply grateful to Dr. S. Latha, Assistant Professor at Department of Criminology & Director, Centre for Cyber Forensics and Information Security, University of Madras, for her steadfast guidance, insightful feedback, and continuous support throughout the journey. Her expertise and encouragement were invaluable every step of the way.

My heartfelt thanks also go to the Department of Computer Science Engineering at Dr. MGR Educational and Research Institute, Chennai, for providing the resources and an environment that fostered focused research and learning.
I would like to acknowledge the creators and contributors of Androguard, TensorFlow, and other AI and forensic tools that were integral to this study. Their dedication to open-source innovation significantly supported the development and evaluation of the proposed forensic system.

Finally, I'm incredibly thankful to my family, friends, and peers for their constant encouragement and support throughout this process. Your belief in me made all the difference.

REFERENCES

[1] Smith, John, and Patel, Amit. "AI in Digital Forensics: Challenges and Applications." Forensic Science Journal, vol. 34, no. 2, 2021, pp. 45–58.
[2] Zhao, Ling, and Kim, Hyun. "Automating Log Analysis Using AI Models." Cybersecurity Review, vol. 28, no. 1, 2022, pp.12–19.
[3] Khan, Rahim, and Gupta, Sandeep. "Role of AI in Enhancing Malware Detection Accuracy." Journal of Information Security, vol. 39, no. 3, 2021, pp. 89–102.

[4] Lee, Min, and Park, Jihoon. “Anomaly Detection in Network Logs Using Machine Learning.” IEEE Transactions on Cybersecurity, vol. 21, no. 4, 2023, pp. 77–85.
[5] Sharma, Priya, and Bose, Nikhil. “AI-Powered Hash Verification for Digital Forensics.” Digital Evidence Journal, vol. 14, no. 5, 2021, pp. 33–40.
[6] Anderson, Charles, and Lee, Thomas. “AI in Malware Classification: A Comparative Study.” Computational Forensics Quarterly, vol. 18, no. 2, 2023, pp. 54–67.
[7] Williams, Alice, and Johnson, Kevin. “Challenges in AI-Integrated Forensics.” Forensic Technology Review, vol. 9, no. 1, 2022, pp. 21–29.
[8] Nakamura, Yuki, and Tanaka, Kenji. “Deep Learning Models for Cybercrime Detection.” Journal of AI and Security, vol. 12, no. 3, 2023, pp. 67–75.
[9] Singh, Vikram, and Dutta, Ananya. “Exploring AI for Forensic Evidence Analysis.” International Journal of Digital Forensics, vol. 7, no. 4, 2021, pp. 41–50.
[10] Roberts, Michael, and Chen, Li. “Legal Challenges of AI in Digital Forensics.” Law and Technology Journal, vol. 15, no. 6, 2023, pp. 32–45.
[11] Gupta, Rajesh, and Thomas, Priya. “Machine Learning in Forensic Log Analysis.” Computer Science Review, vol. 17, no. 3, 2022, pp. 88–97.
[12] Zhao, Wei, and Zhang, Yan. “Detecting Cyber Threats Using AI Models.” Journal of Information Security and Applications, vol. 45, no. 2, 2023, pp. 112–120.
[13] Liu, Qiang, and Wang, Hao. “AI in Malware Detection: A Comprehensive Review.” Malware Analysis Journal, vol. 19, no.5, 2021, pp. 102–115.
[14] Gupta, Anil. “Malware Dataset Collection and Classification for Digital Forensics.” Proceedings of the International Conference on AI and Cybersecurity, 2023.
[15] Kumar, Raj, and Sharma, Deepak. “CNN-Based Malware Detection for Cybersecurity.” International Conference on Machine Learning and Security, 2022.
[16] Singh, Rohan, and Verma, Kunal. “Public Dataset Analysis for Anomaly Detection in System Logs.” International Conference on Cybersecurity Research, 2022.
[17] Patel, Amit. “Enhancing Digital Forensics with AI: Anomaly Detection in System Logs.” Journal of Cyber Security, vol. 45, no. 3, 2021, pp. 123–135.
[18] Jones, Timothy. “Ensuring Data Integrity with SHA-256 Hashing Techniques.” Digital Investigation Journal, vol. 28, 2023, pp. 55–66.
[19] Roberts, Christopher, and Nelson, Mark. “File Hashing Techniques in Digital Evidence Verification.” Journal of Information Security and Applications, vol. 19, no. 4, 2023, pp. 98–109.

