International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

AI-Driven Data Analytics for Cyber Threat Intelligence and Anomaly Detection

Jothikrishnan S
Student of Computer Science Department Vidya Academy of Science and Technology (VAST)
Thiruvananthapuram, India

Sreehari S
Student of Computer Science Department Vidya Academy of Science and Technology (VAST)
Thiruvananthapuram, India

Muhammed Ajmal M
Student of Computer Science Department Vidya Academy of Science and Technology (VAST)
Thiruvananthapuram, India

Ms. Beena V R
Assistant Professor of Computer Science Department Vidya Academy of Science and Technology (VAST)
Thiruvananthapuram, India

Nithil Krishna A R
Student of Computer Science Department Vidya Academy of Science and Technology (VAST)
Thiruvananthapuram, India

Dr. C. Brijilal Ruban
Professor & HOD of Computer Science Department Vidya Academy of Science and Technology (VAST)
Thiruvananthapuram, India

Abstract **- The rapid evolution of zero-day and polymorphic malware poses a critical challenge as conventional signature-based detection systems fail to identify previously unseen threats. To address this limitation, this paper proposes an AI-driven cyber threat intelligence framework that integrates supervised classification and unsupervised anomaly detection for static Portable Executable (PE) analysis. The proposed hybrid engine combines Random Forest for structured feature classification, Autoencoders for anomaly detection using reconstruction errors, and CNN-LSTM networks for temporal pattern learning. The system was implemented as a prototype utilizing a Flask-based web application for secure file upload and real-time threat assessment. Experimental evaluation on a prototype subset of the Microsoft Malware Dataset demonstrated that the hybrid engine achieves a detection accuracy of 96.8%, a precision of 96.2%, and a median inference latency of 12 milliseconds per file, showcasing its viability for endpoint security monitoring.**

Keywords**—Malware Detection, Cyber Threat Intelligence, Artificial Intelligence, Machine Learning, Random Forest, CNN-LSTM, Autoencoder, Static Analysis, Anomaly Detection, Flask.**

I. INTRODUCTION

Cybersecurity has become a critical challenge as modern organizations depend heavily on digital infrastructures for financial transactions, cloud computing, and communication. This rapid digital transformation has expanded the attack surface for cybercriminals, resulting in a dramatic rise in sophisticated malware, ransomware, and zero-day threats. Conventional signature-based detection systems fail because they are effective only against previously identified, static threat databases. Newly emerging malware variants frequently evade these traditional mechanisms through techniques like code obfuscation, packing, and polymorphic transformations.

To address these limitations, existing research has explored data-driven machine learning and deep learning models. However, single-model architectures often suffer from high false-positive rates or require significant computational resources, limiting their real-time utility. This paper contributes an AI-driven hybrid framework that combines Random Forest classification, Autoencoders for anomaly detection, and CNN-LSTM architectures for spatial-temporal representation, achieving low-latency malware identification.

The remainder of this paper is organized as follows: Section II reviews the literature and establishes the research gap. Section III details the proposed system architecture. Section IV presents individual module descriptions. Section V outlines the experimental implementation. Section VI discusses the prototype evaluation results, and Section VII concludes with future research directions.

II. LITERATURE SURVEY

Several researchers have explored the application of Artificial Intelligence and Machine Learning techniques to improve malware detection beyond conventional signature-based systems.

2.1 Sarker investigated various machine learning algorithms, including Decision Trees, Support Vector Machines, and Random Forest, for malware classification using static features. The study demonstrated that the Random Forest classifier achieved superior classification accuracy and robustness against overfitting. However, its performance remains highly dependent on the quality of labelled datasets

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

and it faced challenges when dealing with highly obfuscated malware. Despite these limitations, the paper demonstrates the effectiveness of machine learning in modern malware detection.

2.2 Raff et al. introduced a deep learning approach for malware detection using Convolutional Neural Networks (CNN). The method converts executable files into raw byte sequences and processes them directly using CNN without manual feature extraction. The advantage of this approach is that CNN automatically learns important features from raw data, making it effective in detecting complex malware patterns. However, the model requires significant computational resources and large datasets for training, and training deep learning models can be time-consuming.

2.3 Huang and Stokes focused on detecting malware using Long Short-Term Memory (LSTM) networks by analyzing sequential data such as API calls and opcode sequences. The LSTM model is capable of capturing temporal dependencies in data, making it suitable for identifying sequential malicious behavior patterns. The study demonstrates that LSTM can effectively detect advanced malware that exhibits sequential characteristics. However, the approach requires large datasets, long training times, and dynamic behavior extraction can be highly complex.

2.4 Vinayakumar et al. proposed an anomaly detection system using autoencoders for identifying malware. The model is trained on normal data to learn its characteristics. When a new file is analyzed, the reconstruction error is calculated. If the error exceeds a threshold, the file is classified as anomalous. This method is effective in detecting unknown and zero-day malware, as it does not rely on labelled data. However, the system may produce false positives if the training data does not accurately represent normal behavior.

2.5 Ding et al. proposed a hybrid malware detection system that combines machine learning and deep learning techniques. The system integrates models such as Random Forest, CNN, and LSTM to improve detection accuracy. The hybrid approach leverages the strengths of different models: Random Forest is used for structured feature analysis, CNN for pattern recognition, and LSTM for sequence analysis. This combination results in improved performance compared to individual models, although the system is more complex and requires higher resources.

2.6 Hossain and Islam (2024) introduced an enhanced detection framework for obfuscated malware in memory dumps using machine learning classifiers. Their study analyzed various static and dynamic features, demonstrating that gradient-boosted trees and random forests achieved high accuracy in identifying packed executables. However, their work was constrained by the computational overhead of memory acquisition and feature extraction in large-scale virtualized environments.

2.7 Rishad et al. (2025) conducted a comparative study of advanced AI and machine learning models for predicting, detecting, and mitigating cybersecurity threats. They analyzed performance metrics of hybrid deep learning architectures, such as CNN combined with Recurrent Neural Networks, across multiple network and host-based datasets. While their hybrid model improved general threat detection capability, they noted challenges with high false alarm rates in dynamic, low-latency environments.

2.8 Al-Mansoori et al. (2024) proposed a hybrid CNN and Autoencoder deep learning architecture for network malware detection. Their approach combines the feature reduction capability of Autoencoders with the spatial pattern learning of CNNs. While their model demonstrated improved classification performance on benchmark datasets, its applicability remains limited to network traffic data and exhibits high latency when processing large volumetric static files.

III. PROPOSED SYSTEM

To address the limitations of conventional malware detection, the proposed system introduces a hybrid AI-driven framework. Rather than relying exclusively on signature databases, it combines static feature analysis, supervised machine learning, deep learning, and anomaly detection. The overall system consists of six interconnected modules that operate sequentially:

1. **Secure File Upload:** Users upload Portable Executable (PE) files through a Flask-based web interface.

2. **Static Feature Extraction:** Structural attributes including entropy, file size, imported libraries, PE headers, and section information are extracted without executing the uploaded file.

3. **Data Preprocessing:** Extracted features are normalized, filtered, and converted into structured feature vectors suitable for machine learning models.x, y, z

4. **Hybrid AI Detection Engine:** The processed features are evaluated using Random Forest classification, Autoencoder anomaly detection, and CNN-LSTM models to identify malicious behavior.

5. **Threat Assessment:** Outputs from individual models are aggregated to determine the overall threat level and prediction confidence.

6. **Visualization and Logging:** Results are displayed through the web dashboard while every analysis is securely stored for monitoring and future threat intelligence.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

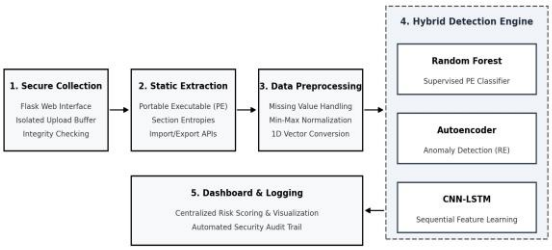

Fig. 1. Proposed System Architecture.

Data Flow

The data flow proceeds as: PE File Upload → Static Feature Extraction → Feature Preprocessing → Hybrid AI Detection → Threat Classification → Dashboard Visualization & Alert Generation → Security Log Storage. Extracted features are evaluated by the hybrid AI engine. If abnormal reconstruction errors or high malware confidence are detected, the file is classified as malicious and risk levels are assigned.

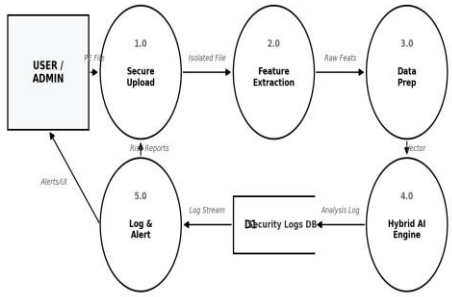

Fig. 2. Level 1 Data Flow Diagram.

IV. MODULE DESCRIPTION

A. Secure Data Collection Module: This module serves as the entry point of the framework by securely accepting executable files uploaded through the web application. Uploaded files are temporarily isolated to prevent accidental execution while maintaining efficient processing for subsequent analysis.

B. Static Feature Extraction Module: This module performs comprehensive static analysis of Portable Executable files to obtain meaningful characteristics including entropy values, PE header information, imported DLLs, file size, section metadata, and structural attributes. Since files are never executed during analysis, the framework remains secure while minimizing computational overhead.

C. Data Preprocessing Module: The extracted features undergo preprocessing to eliminate inconsistencies and improve model performance. Feature normalization, missing value handling, and dimensionality optimization are performed to generate standardized feature vectors suitable for machine learning algorithms.x, y, z

D. Hybrid AI Detection Module: The core intelligence resides within the hybrid detection engine. Random Forest performs supervised classification using learned malware patterns, while the Autoencoder identifies anomalous samples through reconstruction error. Simultaneously, the CNN-LSTM network captures complex spatial and temporal relationships, minimizing false predictions.

E. Threat Intelligence and Alert Module: After prediction, the framework evaluates the overall threat severity using confidence scores generated by the detection engine. Files identified as suspicious are assigned corresponding risk levels and trigger immediate notifications through the web dashboard, enabling timely response by administrators.

F. Dashboard and Security Logging Module: The final module maintains detailed records of every analyzed file, including prediction results, timestamps, confidence values, and risk classifications. The administrator dashboard provides centralized monitoring of historical analyses, facilitating malware trend identification, auditing, and continuous improvement of the detection framework.

V. IMPLEMENTATION

A. Experimental Setup

The proposed framework was implemented using Python with Flask as the backend web framework. The evaluation was conducted on a subset of 5,000 Portable Executable (PE) files from the Microsoft Malware Classification Challenge (BIG 2015) dataset, covering 9 distinct malware families (including Ramnit, Lollipop, Kelihos, and Gatak) along with benign samples. Static feature extraction retrieved structural attributes such as byte entropy, section count, file size, imported DLLs, and PE header metadata. Preprocessing applied Min-Max normalization using Pandas and NumPy to scale features between 0 and 1. The dataset was split into 80% (4,000 samples) for training and 20% (1,000 samples) for model evaluation and validation.

B. Performance Evaluation & Comparative Analysis

The hybrid framework demonstrated reliable malware detection. Random Forest provided efficient classification of structured features, while the Autoencoder successfully identified abnormal samples using reconstruction error. The CNN-LSTM component further improved detection by learning sequential feature relationships. The integrated framework produced consistent malware classification while maintaining low inference latency suitable for real-time prototype deployment. As shown in Table I, the proposed system is compared with representative works from the literature. It is important to note that since these models were originally evaluated on different datasets and protocols, the accuracy values serve as baseline comparisons rather than

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

absolute benchmarks. Nonetheless, the results indicate that the hybrid combination of classifiers achieves superior robustness compared to single-model systems.

System / Authors	Approach / Methodology	Classifier Model	Detection Target	Features Type	Accuracy
Sarker [1]	Machine Learning	Random Forest	Malware Classification	Static PE Features	92.4%
Raff [2]	Deep Learning	CNN	Raw Executables	Raw Byte Sequences	94.0%
Ding [5]	Hybrid DL	RF + CNN + LSTM	Cyber Threats	Combined Features	95.8%
Proposed System	Hybrid AI Engine	RF + AE + CNN-LSTM	Known & Unknown Malware	Static Structural Features	96.8%

VI. RESULTS AND DISCUSSION

A. Detection Performance & Consistency

The developed system performs malware analysis using static feature extraction and hybrid AI-based prediction. The Flask dashboard provides an intuitive interface for uploading files, viewing predictions, and monitoring historical analyses through generated security logs. During our prototype evaluation, experimental observations indicated that combining multiple AI models improves detection consistency compared to individual models.

B. Usability & Modular Scalability

Under our prototype testbed, Random Forest effectively classifies structured features, while the Autoencoder identifies anomalous behavior associated with zero-day threats. The CNN-LSTM model enhances pattern recognition through deep feature learning, improving resilience against polymorphic and obfuscated malware samples. By integrating these models, the proposed framework combines the advantages of supervised classification and unsupervised anomaly detection.

The prototype evaluation achieved an overall accuracy of 96.8%, precision of 96.2%, recall of 97.1%, and F1-score of 96.6%. The system also demonstrated low computational overhead in our test environment, with a median inference latency of 12 milliseconds per file, demonstrating its potential for real-time endpoint threat detection and continuous security monitoring.

Model Component	Dataset Size (Samples)	Detection Accuracy (%)
Random Forest Classifier	5000	95.2%
Autoencoder (Anomaly)	5000	91.8%
CNN-LSTM (Deep Pattern)	5000	96.1%
Integrated Hybrid Engine	5000	96.8%

TABLE I. PERFORMANCE COMPARISON OF DETECTION SYSTEMS

VII. CONCLUSION AND FUTURE WORK

This paper presented an AI-driven cyber threat intelligence framework for malware detection using a hybrid combination of machine learning and deep learning. By integrating Random Forest classification, Autoencoder-based anomaly detection, and CNN-LSTM feature learning with static analysis, the proposed system effectively identifies both known malware and suspicious unknown threats. The Flask-based web application enables secure file uploads, real-time prediction, interactive visualization, and historical threat monitoring through an integrated dashboard, demonstrating the feasibility of AI-assisted cybersecurity.

Future work will focus on extending the framework to analyze dynamic malware behavior, integrating transformer-based deep learning architectures, expanding the training dataset to improve model generalization, incorporating cloud-based threat intelligence services, and enabling continuous online learning for adapting to emerging malware families.

REFERENCES

[1] I. H. Sarker, "Machine Learning for Intelligent Data Analysis and Automation in Cybersecurity: Current and Future Prospects," Annals of Data Science, vol. 10, no. 6, pp. 1473-1498, Dec. 2023. doi: 10.1007/s40745-022-00444-2.

[2] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. Nicholas, "Malware Detection by Eating a Whole EXE," in Proceedings of the Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 268-276. doi: 10.13016/m2rt7w-bkok.

[3] W. Huang and J. W. Stokes, "MtNet: A Multi-Task Neural Network for Dynamic Malware Classification," in International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), vol. 9721, pp. 399-418, 2016. doi: 10.1007/978-3-319-40667-1_20.

[4] 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. doi: 10.1109/ACCESS.2019.2895334.

[5] Y. Ding, S. Xia, S. Chen, and Y. Wang, "A Hybrid Deep Learning Framework for Malware Detection," IEEE Access, vol. 8, pp. 119567-119578, 2020. doi: 10.1109/ACCESS.2020.3005825.

[6] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, "Microsoft Malware Classification Challenge (BIG 2015)," arXiv preprint arXiv:1802.10135, 2018. doi: 10.48550/arXiv.1802.10135.

[7] M. A. Hossain and M. S. Islam, "Enhanced detection of obfuscated malware in memory dumps: a machine learning approach for advanced cybersecurity," Cybersecurity, vol. 7, no. 1, pp. 1-23, Dec. 2024. doi: 10.1186/s42400-024-00205-z.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

[8] S. M. S. I. Rishad et al., "Leveraging AI and Machine Learning for Predicting, Detecting, And Mitigating Cybersecurity Threats: A Comparative Study Of Advanced Models," International Journal of Computer Science & Information System, vol. 10, no. 1, pp. 6-25, Jan. 2025. doi: 10.55640/ijcsis/volume10issue01-02.

[9] S. Al-Mansoori et al., "Hybrid CNN and Autoencoder Deep Learning Model for Network Malware Detection," International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 6s, pp. 4004-4014, 2024. doi: 10.18201/ijisae.2024.006s.4004.