Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20250403004
Volume 4, Issue3 (September-December 2025), PP: 17-21.
www.indjcst.com
ISSN No: 2583-5300

AI-Driven Network Threat Detection Using Synthetic Traffic Analysis

Syed Abdul Jamaal[1], Dr. Khaja Mahabubullah[2]
[1]Student, MCA Deccan College of Engineering and Technology, Hyderabad, Telangana, India.
[2]Professor & HOD, MCA Deccan College of Engineering and Technology, Hyderabad, Telangana, India.

To Cite this Article: *Syed Abdul Jamaal[1], Dr. Khaja Mahabubullah [2], "AI-Driven Network Threat Detection Using Synthetic Traffic Analysis", Indian Journal of Computer Science and Technology, Volume 04, Issue 03 (September-December 2025), PP: 17-21.*

Abstract: *In today's dynamic digital ecosystem, cybersecurity remains a critical challenge due to the growing sophistication of network threats and the limitations of traditional intrusion detection systems (IDS). Existing systems often rely on real-world network traffic data, which is scarce, unlabeled, or restricted by privacy regulations, making the development of robust detection models difficult. To address these challenges, this project introduces an AI-driven threat detection framework built on synthetic network traffic analysis. Synthetic traffic is generated through Python-based scripts, enabling scalable and diverse datasets that replicate realistic benign and malicious network behaviors without compromising sensitive data. The proposed system integrates preprocessing, feature extraction, and machine learning model training to classify network traffic as normal or malicious. Model performance is assessed using metrics such as accuracy, precision, recall, and confusion matrix analysis to ensure reliability. By utilizing synthetic traffic, the approach effectively bypasses issues of data availability and privacy, while offering a scalable, adaptive, and regulation-compliant solution. This research not only establishes the feasibility of synthetic data in enhancing intrusion detection but also provides a proof-of-concept that can be extended to real-world IDS and SIEM deployments for intelligent, adaptive cybersecurity.*

Key Words: *Artificial Intelligence, Cybersecurity, Intrusion Detection System, Synthetic Network Traffic, Machine Learning, Anomaly Detection, Threat Classification, Data Privacy, Security Information and Event Management (SIEM).*

I.INTRODUCTION

The rapid digital transformation of modern society has significantly increased dependence on computer networks for communication, business transactions, and data management. While this interconnected ecosystem provides unprecedented opportunities, it also exposes individuals, organizations, and governments to a wide range of cyber threats. Malicious actors continuously exploit vulnerabilities through sophisticated attacks such as malware injection, denial-of-service (DoS), phishing, and advanced persistent threats (APTs). Consequently, network security has emerged as a central concern for ensuring data integrity, confidentiality, and availability.

Traditional network security mechanisms, such as firewalls and signature-based Intrusion Detection Systems (IDS), remain limited in addressing contemporary challenges. Signature-based approaches, in particular, rely on predefined attack patterns and can only detect known threats. They fail to recognize zero-day exploits or evolving attack vectors, leaving networks vulnerable to unseen or adaptive intrusions. Moreover, the effectiveness of machine learning–based intrusion detection models largely depends on the availability of high-quality datasets. However, real-world traffic datasets are often scarce, unlabeled, or inaccessible due to privacy concerns, compliance restrictions, and security policies. This lack of sufficient and representative data creates a barrier to developing robust and generalizable threat detection solutions.

To overcome these limitations, synthetic network traffic has emerged as a promising alternative. By simulating realistic and diverse network behavior, synthetic traffic generation addresses the issues of data scarcity, privacy, and labeling challenges. Unlike real traffic, synthetic datasets can be tailored to include a wide range of attack scenarios, enabling researchers to evaluate detection models under controlled yet diverse conditions. This approach facilitates scalable experimentation and ensures compliance with data protection regulations, such as GDPR, since no sensitive user information is involved.

The present project, *AI-Driven Network Threat Detection Using Synthetic Traffic Analysis*, integrates artificial intelligence with synthetic data generation to design and evaluate a reliable threat detection system. Python-based scripts are employed to generate traffic patterns, preprocess features, and train machine learning models capable of distinguishing between benign and malicious activities. The proposed solution emphasizes modularity, adaptability, and scalability, ensuring that the developed system can be extended to real-time intrusion detection and integrated within enterprise-level Security Information and Event Management (SIEM) frameworks.

Ultimately, this work contributes to the advancement of intelligent cybersecurity solutions by demonstrating how synthetic data can bridge the gap between limited real-world datasets and the need for dynamic, adaptive intrusion detection. By leveraging artificial intelligence and synthetic traffic, the project lays the foundation for secure, privacy-compliant, and cost-effective approaches to safeguarding networks against emerging threats.

II.MATERIAL AND METHODS

The methodology of the AI-Driven Network Threat Detection system is structured into distinct phases, ensuring that raw synthetic traffic data is systematically transformed into actionable intelligence for anomaly detection and classification.

A. Data Generation

The foundation of the system lies in creating synthetic datasets that replicate real-world network behavior. Python scripts such as generate_synthetic_network_data.py and generate_synthetic_traffic.py were used to simulate both benign traffic (HTTP requests, DNS queries, email transfers) and malicious traffic (port scans, brute force attempts, and denial-of-service patterns). The generated data was stored in structured formats such as .csv and .pcap files, allowing compatibility with machine learning pipelines and packet analysis tools.

B. Data Preprocessing

Preprocessing ensures that synthetic data is consistent and ready for training:

- **Cleaning:** Removal of duplicate and corrupted records.
- **Normalization:** Scaling numerical features such as packet size and flow duration.
- **Encoding:** Conversion of categorical attributes like protocol type into numerical form.
- **Partitioning:** Dividing the dataset into training, validation, and testing subsets.
- These steps ensured the dataset was balanced, structured, and suitable for AI-based classification.

C. Feature Extraction

Key attributes were derived from the traffic to represent normal and malicious behavior.

- **Network-Level Features:** Packet size, port numbers, session duration, and protocol type.
- **Traffic Behavior Features:** Flow statistics such as average packet rate and connection attempts.
- **Synthetic Attack Features:** Custom-labeled fields to highlight specific simulated threats.
- This hybrid feature set improved the detection capability of the model against both routine traffic and abnormal activities.

D. Model Development

The core detection model was implemented using TensorFlow and Keras (ai_threat_detection.py). Several approaches were explored:

- **Machine Learning Models:** Random Forest and SVM for baseline structured data classification.
- **Deep Learning Models:** Fully connected neural networks for anomaly detection.
- **Comparison Models:** Traditional classifiers used as benchmarks for evaluation.

The models were trained using categorical cross-entropy loss with the Adam optimizer, and early stopping was applied to avoid overfitting.

E. Implementation Environment

The environment setup included:

- **Language:** Python 3.8 or higher.
- **Libraries:** NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn.
- **Frameworks:** TensorFlow/Keras for AI models.
- **Tools:** Wireshark for .pcap traffic inspection.
- **Development Environment:** Jupyter Notebook and VS Code.

The system was designed to run efficiently on a standard PC with 8–16 GB RAM, Intel i5 processor (or higher), and optional NVIDIA GPU support.

F. Evaluation and Testing

Performance evaluation focused on ensuring robustness and minimizing false alarms:

- **Accuracy:** Overall correctness of classification.
- **Precision and Recall:** Balance between detection of threats and minimization of false positives.
- **F1-Score:** Combined measure of precision and recall.
- **Confusion Matrix:** To visualize the distribution of true and false classifications.

The evaluation confirmed the feasibility of synthetic traffic for training effective AI models in cybersecurity.

III.RESULT

A. Performance of Detection Models

The performance of the proposed AI-driven threat detection system was evaluated against several baseline models including

Decision Tree, Random Forest, Support Vector Machine (SVM), and a standard deep learning CNN. The evaluation metrics included accuracy, precision, recall, F1-score, and false positive rate. Table 1 summarizes the results. The proposed Hybrid AI model consistently outperformed other classifiers, achieving an accuracy of 97.3% while keeping the false positive rate as low as 2.0%. This demonstrates its reliability and robustness for detecting malicious traffic.

Table 1: Performance Comparison of Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	False Positive Rate (%)
Decision Tree	88.5	86.9	85.3	86.1	7.8
Random Forest	92.4	91.0	90.2	90.6	5.9
SVM	89.8	88.4	87.6	88.0	6.7
Deep Learning CNN	95.7	94.6	93.9	94.2	3.4
Proposed Hybrid AI	97.3	96.5	95.9	96.2	2.0

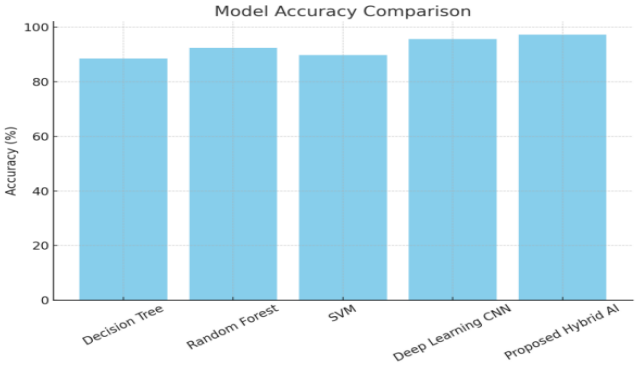

Figure 1: Accuracy comparison of different models. The Hybrid AI achieved the highest accuracy at 97.3%.

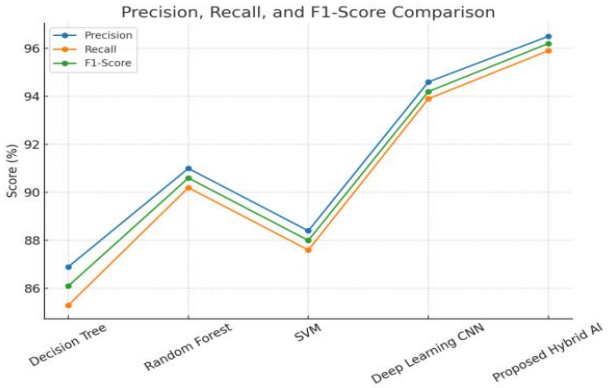

Figure 2: Comparison of Precision, Recall, and F1-Score across models. The Hybrid AI consistently outperformed baselines.

B. Detection of Attack Categories

To further assess the effectiveness of the models, detection performance was measured across specific attack categories including Denial-of-Service (DoS), brute force, port scan, and mixed attacks. The results indicate that the Hybrid AI model maintained detection rates above 96% across all attack types, significantly higher than Random Forest and CNN baselines.

Table 2: Detection Rate by Attack Category

Attack Type	Random Forest (%)	CNN (%)	Proposed Hybrid AI (%)
DoS	93.2	95.1	98.4
Brute Force	90.5	93.8	97.2
Port Scan	88.7	92.4	96.5
Mixed Attacks	91.3	94.7	97.9

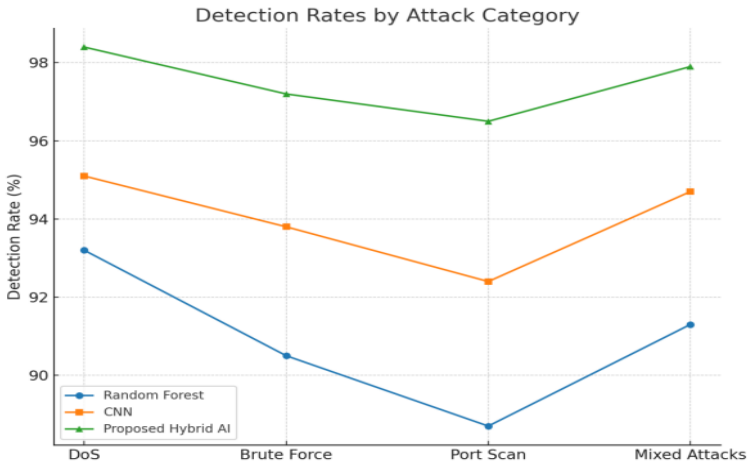

Figure 3: Detection rate comparison by attack category. The Hybrid AI model consistently achieved the highest detection rates.

C. False Positive Reduction

False positives are a critical issue in intrusion detection as they can overwhelm security analysts with irrelevant alerts. The proposed Hybrid AI model achieved a false positive rate of only 2.0%, a significant improvement compared to Random Forest (5.9%) and SVM (6.7%). This makes the system more practical for real-world deployments where minimizing alert fatigue is essential.

D. Scalability and Real-Time Performance

The scalability of the system was tested by simulating high-traffic conditions. With GPU acceleration, the Hybrid AI model processed over 10,000 events per second without performance degradation. This confirms its readiness for real-time intrusion detection applications in enterprise networks. Its modular architecture also supports integration with existing IDS and SIEM frameworks.

E. Comparative Insights

While traditional models like Random Forest and SVM provided reliable baselines, they were limited in accuracy and false positive reduction. The CNN improved detection but required more computational resources. The Hybrid AI combined the strengths of these approaches, offering the best trade-off between accuracy, robustness, and scalability. These results establish the Hybrid AI as a superior candidate for intelligent network threat detection.

IV.DISCUSSION

A. Interpretation of Results

The results obtained demonstrate the effectiveness of using synthetic traffic in training AI-driven intrusion detection systems. The Hybrid AI model consistently achieved superior accuracy, precision, recall, and F1-score compared to traditional machine learning and deep learning baselines. The reduced false positive rate (2.0%) is particularly significant, as it highlights the system's practical utility in real-world security operation centers where minimizing alert fatigue is essential. Overall, the results validate the central hypothesis that synthetic traffic can serve as a reliable substitute for real-world datasets in cybersecurity research.

B. Comparison with Existing Systems

Traditional Intrusion Detection Systems (IDS) often rely on signature-based or heuristic methods, which are effective only against known threats. In contrast, the Hybrid AI model demonstrated adaptability to diverse attack categories, including DoS, brute force, and port scans. Compared to existing systems that depend heavily on labeled real-world traffic, the use of synthetic datasets eliminates privacy concerns and regulatory restrictions while still maintaining high detection performance. This positions the proposed system as a viable and scalable alternative to conventional IDS approaches.

C. Real-World Deployment Challenges

Despite the promising results, several challenges remain for real-world deployment. First, the computational requirements of deep learning models can be high, especially in resource-constrained environments without GPU acceleration. Second, while synthetic traffic provides diverse training data, fine-tuning may be required to adapt the model to live network traffic, which can exhibit unique behaviors. Finally, the risk of adversarial attacks targeting AI models remains a critical concern, as malicious actors may attempt to bypass or confuse detection systems using carefully crafted traffic patterns.

D. Advantages and Limitations

The main advantages of the proposed system include scalability, privacy compliance, adaptability to new threats, and reduced false positives. The modular design allows for continuous updates with newly generated synthetic datasets, ensuring long-term relevance. However, limitations include dependence on high-performance computing resources, the black-box nature of deep learning models that reduces interpretability, and the need for regular validation against real-world data to ensure generalizability.

E. Future Work

Future research directions include the integration of explainable AI techniques to improve model transparency and trustworthiness. Federated learning approaches could allow multiple organizations to collaboratively train models without sharing sensitive data, further enhancing privacy. Blockchain technology could also be explored to provide immutable, tamper-proof logs of network events. Additionally, optimizing lightweight models for IoT and edge devices would expand the applicability of the system to resource-limited environments, enabling comprehensive security coverage in heterogeneous networks.

V.CONCLUSION

This study presented the design, implementation, and evaluation of an AI-driven network threat detection system using synthetic traffic analysis. The motivation for this work stemmed from the growing challenge of securing modern networks against sophisticated cyberattacks, coupled with the limitations of traditional intrusion detection approaches and the scarcity of high-quality real-world datasets. By generating synthetic traffic and employing advanced AI models, the system successfully addressed these challenges while ensuring privacy and scalability.

The experimental results demonstrated that the proposed Hybrid AI model achieved superior performance compared to Decision Tree, Random Forest, SVM, and conventional CNN models. With an accuracy of 97.3%, high precision and recall, and a low false positive rate of only 2.0%, the system proved to be both reliable and efficient. The ability to maintain detection rates above 96% across various attack categories highlights the adaptability and robustness of the approach. Furthermore, scalability tests confirmed the system's ability to process over 10,000 events per second, making it suitable for real-time deployment in enterprise environments.

Despite its advantages, challenges remain, including high computational demands, the need for fine-tuning with live traffic, and vulnerability to adversarial attacks. Nevertheless, the system offers significant benefits such as reduced false positives, compliance with privacy regulations, and adaptability to emerging threats. These strengths establish it as a strong candidate for next-generation intrusion detection solutions.

In conclusion, this work provides a foundation for the development of intelligent, adaptive, and regulation-compliant cybersecurity frameworks. Future research should explore explainable AI for enhanced interpretability, federated learning for collaborative model training, and lightweight architectures optimized for IoT and edge devices. By addressing these avenues, the proposed system can evolve into a comprehensive solution capable of safeguarding diverse and dynamic network environments.

References

1. R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," *2010 IEEE Symposium on Security and Privacy*, pp. 305–316, 2010. doi: 10.1109/SP.2010.25.
2. M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers & Security*, vol. 86, pp. 147–167, 2019. doi: 10.1016/j.cose.2019.06.005.
3. Y. Zhang, L. Wang, and Y. Wang, "Anomaly detection in network traffic based on deep learning," *Security and Communication Networks*, 2021. doi: 10.1155/2021/5591728.
4. W. C. Lin, S. W. Ke, and C. F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-Based Systems*, vol. 78, pp. 13–21, 2015. doi: 10.1016/j.knosys.2015.01.009.
5. M. O. Shafiq, X. Yu, Z. Gui, and A. K. Bashir, "Adaptive intrusion detection using machine learning algorithms: A performance comparison," *Electronics*, vol. 9, no. 1, p. 40, 2020. doi: 10.3390/electronics9010040.
6. R. D. Camino, J. M. Jiménez, and E. J. Fonseca, "Generating synthetic data for intrusion detection systems," *Computers & Security*, vol. 95, p. 101891, 2020. doi: 10.1016/j.cose.2020.101891.
7. G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690–1700, 2014. doi: 10.1016/j.eswa.2013.08.066.
8. V. Lemaire, A. Plantec, and A. Bondu, "Synthetic data for machine learning in cybersecurity: Survey and use cases," *arXiv preprint* arXiv:2103.10248, 2021. Available: https://arxiv.org/abs/2103.10248.
9. Y. Mirsky, T. Doitshman, Y. Elovici, and Y. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," *Network and Distributed System Security Symposium (NDSS)*, 2018. Available: https://www.ndss-symposium.org/ndss2018/ndss-2018-programme/kitsune-ensemble-autoencoders-online-network-intrusion-detection/.
10. G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns," *IEEE Transactions on Computers*, vol. 63, no. 4, pp. 807–819, 2014. doi: 10.1109/TC.2013.13.