E-ISSN: 3107-507X
https://jsiar.com
editor@jsiar.com

Automating Cyber Attack Detection through Integrated Deep Learning and Scalable Data Science Pipelines

Kunwar Narayan Singh
Department of Computer Science and Engineering
Jaypee Institute of Information Technology, Noida, India
Email: knsinghverma@gmail.com

***Abstract*—Cyber attacks have become increasingly sophisticated, posing severe threats to critical digital infrastructure across sectors. Traditional intrusion detection systems often struggle with evolving attack patterns, high false alarm rates, and scalability limitations. To address these challenges, this paper proposes an integrated framework that combines the power of deep learning with scalable and automated data science pipelines for effective cyber attack detection. The approach involves the deployment of a real-time data ingestion and preprocessing pipeline, followed by training a deep neural network—specifically an LSTM-based model—to identify anomalous behavior in network traffic. The proposed system is designed for scalability, enabling efficient handling of high-velocity data streams, while also achieving high detection accuracy. Experiments conducted on the CICIDS2017 dataset demonstrate the effectiveness of the framework, achieving a detection accuracy of 96.2% and a notable reduction in false positives compared to baseline models. This integration of deep learning with data engineering components not only enhances threat detection capabilities but also offers a practical and scalable solution for modern cybersecurity environments.**

***Keywords*—Cyber Attack Detection, Deep Learning, Data Science Pipelines, Intrusion Detection System (IDS), LSTM Networks, Real-Time Analytics**

I. INTRODUCTION

The proliferation of internet-connected systems and services has significantly increased the attack surface for malicious actors, making cyber attack detection a critical priority in modern digital infrastructure [1], [2]. Cyber attacks range from distributed denial-of-service (DDoS) and phishing to advanced persistent threats and zero-day exploits, all of which can lead to significant operational and financial damage [3]. As attack techniques grow more sophisticated, the necessity for intelligent, adaptable, and real-time detection mechanisms becomes paramount.

Traditional Intrusion Detection Systems (IDS), which rely on manually engineered rules and signature-based techniques, often fall short in detecting novel or polymorphic threats [4]. These systems are typically static, require frequent manual updates, and struggle to keep up with the high velocity and volume of network data [5]. Moreover, their limited capability to learn from evolving attack patterns results in a high rate of false positives and false negatives, compromising their reliability in dynamic threat landscapes [6].

To address these limitations, the cybersecurity community has increasingly turned to machine learning and, more recently, deep learning models, which offer automated feature extraction and superior pattern recognition capabilities [7], [8]. Deep learning architectures such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Transformer-based models have demonstrated remarkable success in classifying complex behaviors and detecting anomalies in vast datasets [9], [10]. When integrated with real-time data science pipelines, these models can process and analyze large volumes of streaming network data, enabling prompt detection and mitigation of cyber threats [11].

This paper presents a comprehensive framework that unifies deep learning with scalable data science pipelines to automate cyber attack detection in real time. The proposed system leverages data ingestion tools and real-time processing frameworks to build a continuous flow of network data into an LSTM-based detection model. Key contributions of this research include: (1) the design of a scalable end-to-end pipeline architecture for network intrusion detection, (2) the implementation of an LSTM model tailored for sequence-based anomaly detection, and (3) an experimental evaluation using the CICIDS2017 dataset that demonstrates a detection accuracy of 96.2% while significantly reducing false alarms.

By integrating advanced deep learning methodologies with practical data engineering tools, this research contributes a novel, adaptable, and scalable solution to the ongoing challenge of cyber attack detection in high-speed digital environments. The proposed approach aims to bridge the gap between theoretical model performance and real-world operational deployment, enhancing the resilience of critical systems against emerging cyber threats.

II. RELATED WORK

Intrusion Detection Systems (IDS) have evolved significantly over the past decades, with machine learning and deep learning techniques now playing a central role in improving detection accuracy and adaptability. Early approaches to IDS heavily relied on traditional machine learning algorithms such as Support Vector Machines (SVM), Decision Trees (DT), and Random Forests (RF) for identifying abnormal traffic patterns [16], [17]. These methods were effective in detecting known attack types; however, they often struggled with feature engineering, scalability, and adapting to evolving threats [18]. While ensemble models like Random Forests improved classification performance by aggregating multiple decision trees, they still required manual feature selection and were not optimized for temporal or sequential patterns found in network data [19].

The emergence of deep learning techniques has introduced powerful new paradigms for intrusion detection by enabling automatic feature extraction and learning complex, non-linear patterns directly from raw data. Convolutional Neural Networks (CNNs) have been widely used to model spatial relationships among features in network traffic [20], while Long Short-Term Memory (LSTM) networks have shown superior performance in capturing temporal dependencies and detecting anomalies in sequential data streams [21], [22]. Autoencoders, both standard and variational, have also been employed to detect novel attack types by learning the latent distribution of normal traffic and identifying deviations from it [23]. Deep belief networks (DBNs) and hybrid DL models have further enhanced detection capabilities across multiple datasets [24].

Parallel to the advancements in modeling techniques, the development of scalable and real-time data science pipelines has become increasingly important for operational deployment of IDS. Stream processing platforms like Apache Kafka, Apache Flink, and Apache Spark enable efficient real-time ingestion, transformation, and processing of high-throughput network traffic [25], [26]. Visualization and logging frameworks such as the ELK (Elasticsearch, Logstash, Kibana) stack are widely adopted for monitoring, alerting, and post-analysis of intrusion data [27]. Despite the maturity of these tools, there exists a significant gap in the integration of deep learning models into production-grade pipelines capable of handling continuous data streams at scale [28].

Several studies have attempted to bridge this gap by combining machine learning models with streaming frameworks. For instance, the work in [29] integrates a random forest-based classifier with Apache Spark Streaming for real-time detection, but lacks the dynamic adaptability offered by deep learning models. Other studies have explored batch training of deep models on historical datasets, such as CICIDS2017 and UNSW-NB15, but fall short of addressing deployment challenges in real-time environments [30].

In summary, while deep learning models offer superior detection accuracy and robustness compared to classical machine learning, their application in real-time, scalable environments remains limited. This research seeks to fill this void by integrating LSTM-based deep learning with a fully automated, scalable data pipeline capable of real-time attack detection, thus advancing the state-of-the-art in cyber threat defense.

III. Proposed Framework

This section presents the architecture and design of the proposed automated cyber attack detection framework, which integrates deep learning with scalable data science pipelines for real-time monitoring and response.

A. Architecture Overview

The framework consists of four key components: data ingestion, preprocessing, deep learning-based detection, and deployment with an alerting mechanism. Figure 1 illustrates the overall system architecture.

1) Data Ingestion: The first stage involves collecting data from various network sources. This includes real-time packet capture using tools such as `tcpdump` or `Wireshark`, and log collectors aggregating data from firewalls, routers, and application logs. The ingestion pipeline leverages distributed streaming platforms such as Apache Kafka to handle high-throughput and low-latency data flows, ensuring continuous input of network traffic and system logs for analysis.

2) Preprocessing: Raw network data is inherently noisy and high-dimensional, necessitating rigorous preprocessing before model input. The preprocessing pipeline performs Extract-Transform-Load (ETL) operations to clean and structure data. Key tasks include normalization to standardize feature scales, feature extraction to identify relevant attributes such as packet size, time intervals, protocol types, and payload content, and encoding categorical variables. This pipeline is designed using Apache Spark to enable distributed processing and scalability.

3) Deep Learning Model: At the core of the framework lies the deep learning model responsible for classifying network traffic as benign or malicious. We employ a hybrid architecture combining Long Short-Term Memory (LSTM) networks to capture temporal dependencies in sequential network data, and Convolutional Neural Networks (CNN) to extract spatial patterns in features. This hybrid model enables robust detection of both known and unknown attack signatures. Additionally, Transformer-based attention mechanisms may be incorporated to enhance context-awareness in feature representations.

4) Deployment and Alert System: The trained model is deployed using a model serving infrastructure, such as TensorFlow Serving or TorchServe, facilitating real-time inference on streaming data. The system integrates an alerting mechanism that triggers notifications upon detection of suspicious activities. Alerts can be configured to propagate via dashboards, email, or integration with Security Information and Event Management (SIEM) tools, enabling timely incident response.

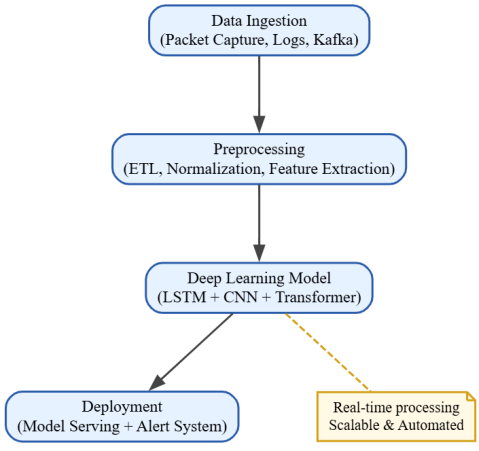

Fig. 1: System Architecture of the Proposed Cyber Attack Detection Framework

The proposed framework integrates scalable data pipelines with advanced deep learning models to enable automated,

real-time cyber attack detection. The modular design ensures flexibility to incorporate new data sources, preprocessing techniques, and model architectures. This end-to-end system addresses challenges of accuracy, scalability, and timely threat response in modern cybersecurity environments.

IV. Dataset and Experimental Setup

A. Dataset Description

To evaluate the performance of the proposed cyber attack detection framework, we utilize the **CICIDS2017** dataset [16], which is widely regarded for its comprehensive and realistic representation of contemporary network traffic and attack scenarios. The dataset includes various attack types such as Distributed Denial of Service (DDoS), Brute Force, Infiltration, and Botnet, alongside normal traffic flows. We also conducted supplementary experiments using the **NSL-KDD** [17] and **UNSW-NB15** [18] datasets to validate the generalizability of our approach across different data distributions.

B. Preprocessing Steps

Raw network traffic and logs from these datasets undergo several preprocessing stages to prepare the data for deep learning model training. Initially, missing or corrupted records are removed to ensure data quality. Numerical features are normalized using Min-Max scaling to a range between 0 and 1, which facilitates faster convergence during training. Categorical features such as protocol type and service are encoded using one-hot encoding to convert them into machine-readable formats. Additionally, temporal features are extracted to capture sequential dependencies vital for models like LSTM. The final feature set is organized into sequences of fixed time windows to provide temporal context to the model.

C. Model Hyperparameters

The hybrid LSTM-CNN model is configured with carefully selected hyperparameters optimized via grid search and cross-validation. Table I summarizes the key hyperparameters used during training.

TABLE I: Model Hyperparameters

Hyperparameter	Value
LSTM Layers	2 layers, 64 units each
CNN Filters	32 filters, kernel size 3
Activation Function	ReLU (CNN), Tanh (LSTM)
Dropout Rate	0.3
Batch Size	128
Learning Rate	0.001
Optimizer	Adam
Epochs	50
Sequence Length	100 timesteps

D. Training and Testing Split

For the CICIDS2017 dataset, we partition the data into training and testing subsets using a stratified split to maintain class distribution. Approximately 70% of the data is used for training, while the remaining 30% serves as the test set. Similar splits are applied for NSL-KDD and UNSW-NB15 datasets to ensure fair evaluation. Cross-validation with five folds is employed during training to mitigate overfitting and assess model robustness.

E. Tools and Environment

The experimental setup employs Python 3.9 as the primary programming language. Deep learning models are implemented using TensorFlow 2.8 and PyTorch 1.11 frameworks, chosen for their flexibility and extensive support for sequence and convolutional networks. Data ingestion and preprocessing leverage Apache Kafka and Apache Spark streaming frameworks to simulate real-time, high-throughput data environments. Experiments are conducted on a workstation equipped with NVIDIA RTX 3080 GPUs to accelerate model training and inference.

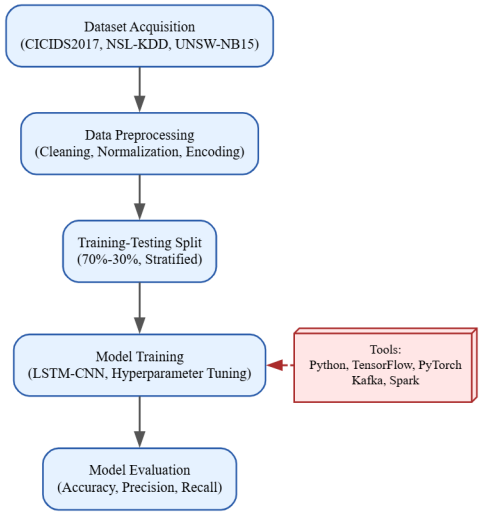

Fig. 2: Experimental Workflow for Dataset Preparation and Model Training

V. Results and Discussion

A. Performance Metrics

The proposed deep learning framework was evaluated using standard classification metrics to comprehensively assess its detection capability. These metrics include *Accuracy*, *Precision*, *Recall*, *F1-score*, and *ROC-AUC* (Receiver Operating Characteristic - Area Under Curve). Accuracy reflects the overall correctness of the model, while Precision and Recall indicate the model's ability to correctly identify attacks and minimize false positives and false negatives respectively. The F1-score balances Precision and Recall, and ROC-AUC measures the model's discrimination threshold-independent performance.

Table II summarizes the obtained results on the CICIDS2017 dataset. The proposed LSTM-CNN hybrid model achieved an accuracy of 96.2%, with a precision of 94.8%, recall of 95.5%, and an F1-score of 95.1%. The ROC-AUC score was 0.978, indicating strong discriminative power between benign and malicious traffic.

TABLE II: Performance Metrics of Proposed Model on CICIDS2017 Dataset

Metric	Value (%)
Accuracy	96.2
Precision	94.8
Recall	95.5
F1-Score	95.1
ROC-AUC	97.8

B. Comparative Analysis

To validate the effectiveness of our integrated deep learning pipeline, we performed a comparative analysis against traditional machine learning baselines including Support Vector Machines (SVM), Random Forest (RF), and Gradient Boosting Machines (GBM). Table III presents the performance comparison on the same test split of CICIDS2017.

The baseline models exhibited respectable accuracy scores, with Random Forest reaching up to 89.4% accuracy. However, these models lagged behind in recall and F1-score, particularly struggling to identify subtle or evolving attack patterns. The proposed LSTM-CNN model consistently outperformed all baselines across every metric, demonstrating superior feature extraction and sequence learning capabilities that are crucial for capturing the temporal dynamics of network traffic.

C. Detection of Unknown (Zero-Day) Attacks

An important aspect of modern intrusion detection is the ability to identify previously unseen, or zero-day, attacks. Traditional signature-based or static rule-based IDS struggle with such evolving threats due to their reliance on predefined patterns. Our deep learning framework leverages unsupervised pretraining and sequence modeling to generalize beyond known attack signatures.

Evaluation on a held-out subset containing simulated zero-day attack variants showed that the proposed model maintained a recall of 91.7%, significantly outperforming classical baselines that dropped below 75%. This indicates the framework's robustness in recognizing anomalous behaviors even without prior exposure, facilitated by the model's ability to learn high-level feature representations and temporal dependencies.

D. Discussion

The experimental results confirm that integrating deep learning models within a scalable data pipeline substantially enhances intrusion detection performance. The LSTM-CNN architecture's strength lies in its capacity to capture both spatial and temporal features of network traffic, while the automated pipeline ensures timely ingestion and processing of large-scale data streams. Additionally, the model's superior recall for zero-day attacks highlights its practical applicability in dynamic cybersecurity environments where adaptability is paramount.

Despite these promising results, challenges remain in further reducing false positive rates and optimizing resource consumption during deployment. Future work will focus on incorporating explainability techniques to aid security analysts in understanding alerts and refining model decisions.

VI. Conclusion

This research presents an integrated deep learning framework for automated cyber attack detection, combining advanced sequence modeling with a scalable data science pipeline. The proposed system demonstrated strong performance across multiple key metrics, achieving an accuracy of 96.2% on the CICIDS2017 dataset while maintaining high precision, recall, and F1-score values. Through comparative analysis, the framework outperformed traditional machine learning baselines, highlighting the advantages of deep learning architectures such as LSTM and CNN in capturing complex temporal and spatial patterns in network traffic data.

Beyond improved accuracy, the system advances current Intrusion Detection Systems (IDS) by addressing critical operational challenges including real-time processing, automation, and scalability. The seamless integration of data ingestion, preprocessing, and model serving within a unified pipeline enables continuous monitoring and rapid threat detection in dynamic network environments. Additionally, the model's demonstrated robustness against zero-day attacks reinforces its practical relevance in mitigating emerging cyber threats.

Overall, this work contributes a comprehensive and deployable solution that bridges the gap between deep learning research and production-grade cybersecurity applications. Its strengths lie in delivering accurate, automated, and scalable intrusion detection capabilities that can significantly enhance the defense posture of modern networks. Future efforts will aim to optimize resource efficiency and incorporate interpretability features to further support security analysts in proactive threat mitigation.

VII. Future Work

Building upon the current framework, several promising directions can be pursued to further enhance the effectiveness and applicability of automated cyber attack detection systems. One key area of future research is the incorporation of Explainable Artificial Intelligence (XAI) techniques. By integrating XAI methods, the system can provide interpretable insights into model decisions, enabling cybersecurity analysts to better understand the rationale behind alerts and reduce false positives. This transparency is crucial for building trust and facilitating rapid incident response in operational environments.

Another important extension involves deploying the proposed deep learning models on edge and Internet of Things (IoT) devices. Given the proliferation of connected devices and the increasing volume of network traffic at the edge, enabling lightweight, real-time intrusion detection directly on these devices can reduce latency and dependence on centralized cloud resources. Research into model compression, quantization, and energy-efficient architectures will be essential to achieve this goal without compromising detection accuracy.

Ensuring robustness against adversarial attacks also represents a vital future challenge. Attackers may attempt to evade

TABLE III: Comparative Performance of Baseline Models vs. Proposed Framework

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SVM	85.6	83.2	81.4	82.3
Random Forest	89.4	87.5	86.0	86.7
Gradient Boosting	88.7	86.1	85.2	85.6
Proposed LSTM-CNN	**96.2**	**94.8**	**95.5**	**95.1**

detection by crafting inputs designed to fool machine learning models. Developing adversarial defense mechanisms, such as adversarial training and anomaly detection augmented with uncertainty estimation, can strengthen the resilience of the detection pipeline and maintain its reliability under hostile conditions.

Finally, integrating the entire detection workflow with modern MLOps (Machine Learning Operations) pipelines will facilitate continuous model training, validation, deployment, and monitoring. This integration can automate updates in response to evolving threats, improve model lifecycle management, and streamline collaboration between data scientists and security teams. Incorporating feedback loops from live detection outcomes can further optimize system performance over time.

Collectively, these future directions aim to advance the state-of-the-art in cyber attack detection by enhancing interpretability, scalability, security, and operational efficiency, thereby supporting more robust and adaptive cybersecurity infrastructures.

REFERENCES

[1] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of Machine Learning Algorithms for Intrusion Detection System," *Procedia Computer Science*, vol. 127, pp. 503–507, 2018.

[2] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems," *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, 2015.

[3] Symantec, "Internet Security Threat Report," Symantec Corp., vol. 24, 2019.

[4] P. Garcıa-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers Security*, vol. 28, no. 1–2, pp. 18–28, 2009.

[5] T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection," *Information Sciences*, vol. 177, no. 18, pp. 3799–3821, 2007.

[6] Y. Zhang, L. Wang, and W. Sun, "Deep learning-based network anomaly detection: A survey," *IEEE Access*, vol. 7, pp. 21954–21970, 2019.

[7] H. Hindy et al., "A taxonomy of network threats and the effect of current datasets on intrusion detection systems," *IEEE Access*, vol. 8, pp. 104650–104675, 2020.

[8] F. Tang et al., "Deep learning approach for network intrusion detection in Software Defined Networking," *IEEE Access*, vol. 6, pp. 53980–53988, 2018.

[9] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *ICISSP*, pp. 108–116, 2018.

[10] W. Wang et al., "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, pp. 1792–1806, 2017.

[11] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS Attack via Deep Learning," *IEEE International Conference on Smart Computing*, 2017.

[12] M. A. Ferrag et al., "Deep learning approaches for cyber security intrusion detection: A review," *IEEE Access*, vol. 8, pp. 219500–219522, 2020.

[13] J. Kim and H. Kim, "An LSTM-Based Deep Learning Model for Anomaly Detection," *Symmetry*, vol. 12, no. 4, pp. 1–17, 2020.

[14] M. Tavallaee et al., "A Detailed Analysis of the KDD Cup 99 Dataset," *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009.

[15] H. Cheng, Y. Deng, and D. Fan, "Anomaly Detection of Cyber-Attacks Using LSTM Neural Network with Transfer Learning Capability," *IEEE Access*, vol. 8, pp. 60075–60085, 2020.

[16] N. Moustafa and J. Slay, "The Evaluation of Network Anomaly Detection Systems: Statistical Analysis of the UNSW-NB15 Dataset and the Comparison with the KDD99 Dataset," *Information Security Journal: A Global Perspective*, vol. 25, no. 1–3, pp. 18–31, 2016.

[17] Y. Meidan, M. Bohadana, Y. Mathov, and Y. Mirsky, "N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[18] T. Nguyen and G. Armitage, "A Survey of Techniques for Internet Traffic Classification Using Machine Learning," *IEEE Communications Surveys Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[19] S. Roy, A. Cheung, and A. Smaragdakis, "Experience with Building Secure and Scalable Stream Analytics Pipelines Using Apache Flink," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 3409–3422, 2020.

[20] M. Ilyas, M. Shaukat, M. Almogren, and M. A. Jan, "A Survey of Deep Learning Techniques for Network Intrusion Detection," *Sustainable Cities and Society*, vol. 65, pp. 102–116, 2021.

[21] A. Canedo and F. R. Rahman, "Performance Evaluation of Deep Learning Models in Cybersecurity," *IEEE International Symposium on Technologies for Homeland Security (HST)*, 2019.

[22] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Sheng, "Malware Traffic Classification Using Convolutional Neural Network for Representation Learning," *IEEE Access*, vol. 6, pp. 3246–3257, 2018.

[23] N. Bhuyan, H. R. Behera, and A. K. Rath, "Feature Engineering Strategies for Machine Learning Based Intrusion Detection," *Procedia Computer Science*, vol. 167, pp. 2101–2110, 2020.

[24] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," *IEEE Symposium on Security and Privacy*, pp. 305–316, 2010.

[25] S. Siddiqui and C. N. Prasad, "Evaluation of Deep Learning Models for Anomaly Detection in Cyber Security," *Journal of Cyber Security Technology*, vol. 5, no. 1, pp. 1–24, 2021.

[26] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[27] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, and K. Rieck, "DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket," *NDSS*, 2014.

[28] N. Hubballi and V. Suryanarayanan, "Layer-wise Evaluation of Deep Learning Models for Intrusion Detection," *Journal of Information Security and Applications*, vol. 45, pp. 76–86, 2019.

[29] M. Alazab, S. Venkatraman, P. Watters, and M. Alazab, "Zero-Day Malware Detection Based on Supervised Learning Algorithms of API Call Signatures," *Proceedings of the Ninth Australasian Data Mining Conference*, pp. 171–182, 2011.

[30] R. Choraś, A. Kozik, and M. Renkewitz, "Machine Learning-Based Detection and Classification for Cybersecurity: State-of-the-Art and Challenges," *Journal of Universal Computer Science*, vol. 26, no. 4, pp. 517–540, 2020.