Enhancing Network Intrusion Detection for TLS Traffic Using Deep Learning

Hidayatul Muttaqien
Department of Electrical Engineering, Universitas Hasanuddin, Gowa, Indonesia
hidayatul.muttaqien@unmul.ac.id (corresponding author)

Muhammad Niswar
Department of Informatics, Universitas Hasanuddin, Gowa, Indonesia
niswar@unhas.ac.id

Syafruddin Syarif
Department of Electrical Engineering, Universitas Hasanuddin, Gowa, Indonesia
syafruddin.s@eng.unhas.ac.id

Zahir Zainuddin
Department of Informatics, Universitas Hasanuddin, Gowa, Indonesia
zahir@unhas.ac.id

Received: 9 July 2025 | Revised: 13 August 2025 and 25 August 2025 | Accepted: 30 August 2025

 DOI: https://doi.org/10.48084/etasr.13267

ABSTRACT

The increased utilization of Transport Layer Security (TLS) encryption in contemporary network communication introduces new obstacles for Network Intrusion Detection Systems (NIDS), since encrypted traffic constrains the efficacy of traditional signature-based techniques. This study presents a real-time intrusion detection method for TLS traffic utilizing a combination of Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory (BiLSTM) networks. CNNs are employed to derive spatial representations of TLS information from Suricata logs, including JA3 fingerprints, cipher suites, and connection statistics, and BiLSTM is utilized to capture bidirectional temporal dependencies of encrypted traffic to identify intricate anomaly patterns. This model was evaluated utilizing an extensive TLS dataset comprising both valid and malicious traffic, including Command-and-Control (C2) connections, malware communication, and data exfiltration. The experimental findings indicate that the CNN–BiLSTM model attained a detection accuracy of 98.7%, a False Positive Rate (FPR) of 1.4%, and an average processing time of 12.9 ms per session, rendering it appropriate for real-time application in corporate network security systems. This methodology enhances the capability of hybrid Deep Learning (DL) models to identify concealed dangers in TLS communication without requiring data decryption.

Keywords-Transport Layer Security (TLS); Network Intrusion Detection Systems (NIDS); Deep Learning (DL); Convolutional Neural Network (CNN); Bidirectional Long Short-Term Memory (BiLSTM); encrypted traffic; Suricata

I. INTRODUCTION

Transport Layer Security (TLS) is an essential protocol for safeguarding the integrity and confidentiality of digital communications via end-to-end encryption. This encryption inherently safeguards against malicious traffic, including Command-and-Control (C2) communication, malware dissemination, and data exfiltration, thereby presenting considerable challenges for traditional Network Intrusion Detection Systems (NIDS) that depend on content inspection [1, 2]. Machine Learning (ML) and Deep Learning (DL) methodologies provide solutions by utilizing TLS metadata, including JA3 fingerprints, cipher suites, and handshake lengths, without necessitating traffic decryption [3]. Lightweight DL architectures have been created to improve intrusion detection efficacy [4], and the hybrid Convolutional Neural Network–Bidirectional Long Short-Term Memory (CNN–BiLSTM) has demonstrated usefulness for encrypted communication within the Internet of Things (IoT) context [5].

The hybrid CNN–BiLSTM model has garnered interest due to its ability to integrate the spatial feature extraction capabilities of CNNs with the temporal dependency comprehension of BiLSTM in network traffic analysis [6]. This model attains elevated precision in identifying TLS anomalies

inside real-world datasets [7]. The design is augmented by the incorporation of attention [8, 9], enabling the model to concentrate processing on essential qualities. The CNN–BiLSTM enhanced with attention has demonstrated efficacy in augmenting accuracy and diminishing false positives in real-time contexts [10].

The efficacy of DL architectures is significantly influenced by the choice of pertinent features. TLS metadata from Suricata can produce several features per session, rendering feature selection strategies crucial to prevent overfitting and diminish computing complexity. Random Forest (RF) is frequently employed to identify significant features, including handshake duration, byte entropy, and cipher suite [11, 12]. The embedded feature selection method is utilized in CNN-based NIDS, enabling the network to internally learn significant weights [13].

In addition to accuracy and computational efficiency, two primary problems in the deployment of contemporary NIDS are real-time processing capabilities and data privacy protection. Authors in [14] introduced a lightweight Gated Recurrent Unit (GRU) model designed for low-latency edge situations. Authors in [10] demonstrated that the hybrid CNN–BiLSTM architecture regularly surpassed conventional models in real-time applications while preserving high accuracy. This methodology is essential for enterprises that consistently oversee TLS traffic on high-velocity networks.

To safeguard the confidentiality of important TLS metadata, Federated Learning (FL) has emerged as a potential methodology. FL enables the distribution of model training without the transmission of raw data among nodes. This ensures data confidentiality without sacrificing detection efficacy, illustrating that the amalgamation of FL with CNN–BiLSTM can mitigate the danger of data leaking while preserving high accuracy in encrypted TLS contexts [15]. This research develops a TLS intrusion detection system utilizing Suricata logs, integrating CNN–BiLSTM with an attention mechanism to facilitate privacy-conscious real-time detection.

Early research has concentrated on the enhancement of NIDS through the use of optimal feature selection techniques and DL architectures [16]. Employing RF for feature selection can markedly enhance classification accuracy [17]. Utilizing embedded feature selection on imbalanced datasets has effectively diminished false positives [18]. Additionally, a reinforcement learning methodology (ID-RDRL) has been proposed for dynamic feature selection [19]. The BiLSTM model for identifying temporal patterns was effectively utilized [10], whereas authors in [20] provided a comprehensive evaluation of DL architectures, including CNN, Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM), in relation to NIDS. Authors in [21] and authors in [22] incorporated a semi-supervised variational autoencoder to improve flexibility. To mitigate the problem of imbalanced data, authors in [23] employed RCSMOTE, which markedly enhanced the model's sensitivity to infrequent yet crucial threats.

The identification of anomalies in encrypted TLS traffic without decryption has emerged as a significant area of research. Authors in [3] were the first to utilize JA3 fingerprinting and TLS metadata for the categorization of malware. Authors in [24] elaborated on this concept by utilizing the ET-BERT paradigm, analogous to Natural Language Processing (NLP), to analyze TLS information as structured input. Authors in [6] enhanced accuracy with a BiLSTM model integrated with multi-head attention, whereas authors in [25] presented a scalable CNN–LSTM hybrid system utilizing MapReduce. In high-traffic contexts, authors in [26] discovered that the hybrid DL model sustains elevated accuracy. The original CNN architecture for NIDS was created by authors in [27], and authors in [28] asserted that RNN can facilitate real-time intrusion detection with high throughput.

Today, execution necessitates integration with Software-Defined Networking (SDN) and a privacy-preserving methodology. Authors in [29] introduced a methodology to improve traffic integrity in SDN environments. Authors in [30] introduced a taxonomy of TLS risks within SDN. Authors in [31] and authors in [32] employed blockchain-based technologies, including smart contracts, to safeguard metadata. Authors in [33] and authors in [34] employed blockchain to enhance fault tolerance in SDN control. Authors in [35] and authors in [36] investigated scalability and real-time security in distributed SDN controllers. Authors in [37] provide an extensive survey on traffic classification utilizing ML, including encrypted TLS as well.

Survey literature offers significant context for the development of encrypted traffic management. Authors in [38] encapsulate the most sophisticated DL methodologies in comprehensive traffic analysis. Authors in [39] investigate graph mining for cybersecurity metadata, including TLS. Authors in [37] and authors in [40] provide a preliminary overview on the utilization of ML and DL in encrypted communication, emphasizing the necessity of extensive TLS attack datasets. Authors in [41] concentrate on the security of wireless communication via DL, whereas authors in [42] highlight the significance of TLS metadata privacy. Authors in [43] assess host-based intrusion detection systems, offering a comparison with conventional methods in encrypted networks.

The literature review indicates that DL, specifically the integration of CNNs and Bi-LSTM architectures, demonstrates significant potential in improving the efficacy of NIDS in the context of encrypted TLS traffic. Hybrid models such as CNN–BiLSTM effectively combine spatial feature extraction and temporal pattern recognition, rendering them exceptionally adept at detecting intricate anomalies in TLS sessions without requiring content decryption. Prior research has indicated that this methodology enhances both accuracy and sensitivity while markedly decreasing false positives. This architecture, bolstered by approaches like RF-based feature selection and attention mechanisms, has demonstrated reliability in addressing difficulties related to real-time processing, data privacy, and fluctuations in attack patterns. This research centers on the development and assessment of the CNN–BiLSTM model to improve intrusion detection in TLS traffic, aligning with contemporary trends and requirements in network security technology.

II. MATERIALS AND METHODS

This study aims to develop an efficient, scalable, and real-time NIDS for encrypted TLS traffic, utilizing a combination of CNN–BiLSTM architecture and RF. The main stages include data gathering, feature extraction and preprocessing, and DL model construction. The dataset was acquired via Suricata, which archives TLS metadata in JSON format (eve.json), encompassing critical details such as JA3 fingerprint, cipher suites, handshake duration, and flow attributes including packet count and bytes exchanged with the server. This approach parallels the studies conducted by authors in [44] and authors in [45]. The features were subsequently processed via CNN feature selection approaches and embedded methods, in alignment with the methodologies of authors in [46] and authors in [14], who emphasized the significance of filtering to mitigate noise. The CNN–BiLSTM model was enhanced by employing CNN as a spatial extractor and BiLSTM to capture temporal dependencies, supplemented by an attention mechanism to emphasize significant aspects. The model training employed the Adam optimizer and regularization techniques (dropout, batch normalization), alongside evaluation criteria including accuracy, precision, recall, False Positive Rate (FPR), and real-time latency (under 30 ms), following the guidelines established by authors in [47]. Comparative studies were performed against baselines including RF, CNN–LSTM, and BiLSTM [48-50]. A lightweight FL prototype was constructed to protect privacy and facilitate distributed deployment, adhering to the methodology proposed by authors in [51]. This integration is anticipated to yield an encrypted TLS NIDS proficient in effectively identifying diverse contemporary attacks in real-world scenarios.

Figure 1 illustrates the workflow of the proposed TLS intrusion detection system, which incorporates a CNN, a BiLSTM, a CNN feature selection mechanism, and CNN attention weights. The procedure commences when researchers activate TLS Data Capture via Suricata. Suricata diligently captures TLS metadata without decrypting the transmission, hence maintaining data privacy while producing logs in the eve.json format. This file includes critical attributes such as JA3 fingerprints, cipher suites, handshake duration, packet statistics, source and destination IP addresses, ports, and TLS version. Following Suricata's data collection, researchers execute the Preprocessing and Feature Extraction phase. At this level, the system purges the data to eliminate missing or inconsistent entries and implements label encoding to transform category features. The researchers subsequently conducted feature selection employing two distinct methodologies. Initially, RF assesses the significance of each attribute according to its impact on model accuracy. Secondly, the attention weights of CNN discern the most pertinent aspects during the training phase, enabling the model to selectively concentrate on information crucial for classification. The chosen features are subsequently processed by two concurrent modeling pathways. The CNN Branch derives geographic patterns from TLS metadata, including packet size distribution and the frequency of cipher suite utilization. Simultaneously, the BiLSTM Branch acquires temporal patterns and event sequences, such as fluctuations in handshake duration or repetitive communication patterns that signify anomalies. Each pathway produces a preliminary prediction at the Classification Layer, which ascertains if the TLS session is benign or malicious. The researchers subsequently integrated the outputs from the two pathways at the Fusion Layer. This layer integrates the spatial data from CNN and the temporal representation from BiLSTM into a cohesive feature vector, thereby augmenting the model's capacity to identify intricate assault patterns. The aggregated outcome is thereafter analyzed by the final Classification Layer to render a definitive judgment on the state of the TLS traffic. The concluding phase is Deployment, during which researchers incorporate the trained model into the Security Information and Event Management (SIEM) system. This connection facilitates real-time intrusion detection in high-velocity networking settings. This procedure enables the system to sustain high accuracy, reduce false positives, and uphold computing economy. Consequently, Figure 1 illustrates an architectural design that is both technically proficient and prepared for large-scale deployment.

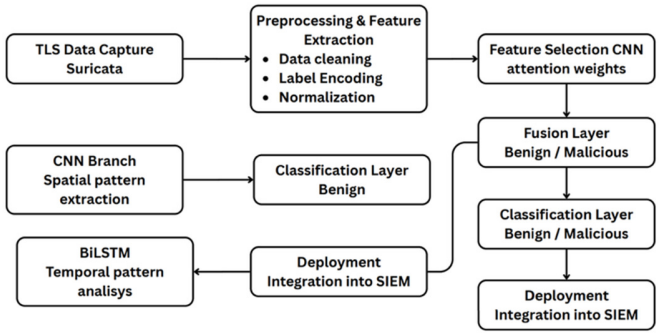

Fig. 1. Workflow of the proposed TLS intrusion detection system, illustrating stages from Suricata-based metadata capture through CNN–BiLSTM classification and SIEM deployment.

A. Dataset Collection Using Suricata

The dataset for this study was generated by running Suricata for one year, from January 2024 to December 2024, and storing the output in Suricata Logs in JSON format within the eve.json file [52]. The dataset was obtained utilizing Suricata, an open-source network threat detection engine proficient in real-time traffic analysis and intrusion detection. Suricata records TLS info in JSON format. This system collects and analyzes TLS packets without performing packet decryption. Suricata logs provide essential information such as cipher suites, handshake duration, TLS session status, packet statistics, JA3 fingerprints, source and destination IP addresses, port numbers, and TLS versions. These logs function as instruments for anomaly detection without necessitating decryption, thereby safeguarding data privacy. TLS packets are aggregated based on Flow ID and classified using Suricata Rules, producing a dataset labeled as benign or malicious traffic. This study analyzed 30,000,000 TLS sessions, including malicious flows identified through anomalies in JA3 fingerprints, repeated unsuccessful handshakes, and connections to recognized malicious domains. The annotated dataset is employed for feature extraction and model training in the proposed TLS intrusion detection system framework.

Figure 2 illustrates the dataset generation flow, from packet capture through preprocessing and labeling.

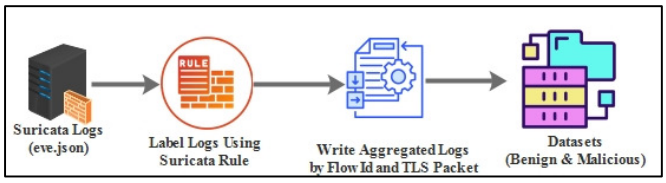

Fig. 2. Dataset generation flow using Suricata.

B. Feature Extraction and Preprocessing

This phase entails the extraction of statistical features and TLS metadata from Suricata logs, encompassing attributes such as packet_count, flow.bytes_to/from_server, flow.duration, src_port, JA3 fingerprint, and cipher suite—analogous to the feature engineering techniques employed by authors in [46], authors in [2], and authors in [29]. Then, the data are subjected to cleansing: entries lacking JA3 or cipher suite are eliminated, in accordance with the methodology established by authors in [53]. Categorical features such as cipher suite and JA3 are encoded via label encoding, whereas numerical features undergo normalization using Min-Max scaling (range 0–1) to enhance model stability during training [54]. To mitigate superfluous features, two methodologies are employed: RF feature importance [46] and embedded CNN attention, as delineated by authors in [45]. The finished dataset comprises approximately 30 to 50 informative attributes, prepared for input into the DL model.

C. Machine Learning Model Development

This study's TLS intrusion detection model was constructed utilizing a hybrid architecture that combines CNN and BiLSTM, augmented by an attention mechanism to emphasize significant aspects in encrypted TLS sessions. CNN is employed to derive spatial patterns from feature representations such as packet count and byte distribution per direction, whereas BiLSTM adeptly captures temporal dependencies in session sequences, including recurrent handshake patterns or anomalous inter-packet intervals. The attention layer is subsequently integrated atop the BiLSTM to prioritize significant aspects, such as JA3 fingerprints and cipher suites commonly associated with malicious traffic, as elucidated in the studies by authors in [8] and authors in [10].

The model was trained using the Adam optimizer with an initial learning rate of 0.001, a batch size of 128, and for 50 epochs. Regularization was implemented via dropout (0.5) and batch normalization to mitigate overfitting, as advised by authors in [44] and authors in [45]. The model was assessed using standard metrics including accuracy, precision, recall, F1-score, FPR, and latency to guarantee real-time performance, in accordance with the methodology established by authors in [47].

The model was compared with baselines including RF, CNN–LSTM, and BiLSTM to verify that the hybrid CNN–BiLSTM offers a substantial benefit. The FL approach was ultimately employed during the training phase to evaluate the efficacy of collaboration among nodes without necessitating the exchange of raw data, as suggested by authors in [51], which is vital for maintaining TLS metadata privacy. This methodology is intended to function effectively in both edge and cloud security contexts utilizing Suricata logs.

III. RESULTS AND DISCUSSION

The hybrid CNN–BiLSTM model outperformed the single CNN model (94.3%) and the single BiLSTM model (95.1%) in experimental testing, achieving a detection accuracy of 98.7%. The measured precision was 97.9%, the recall was 98.5%, and the FPR was at 1.4%. The model's average latency was merely 12.9 ms per TLS session, rendering it appropriate for real-time detection in extensive network environments. The enhancement in performance results from the CNN's ability to extract spatial features from the packets, while the BiLSTM discerns sequential temporal patterns, such as handshake timeouts, obsolete cipher suites, and C2 data patterns. Figure 3 depicts the distribution of performance metrics, demonstrating a positive link between architectural complexity and detection accuracy. The hybrid model reliably identifies malicious TLS as effectively as or more effectively than the baseline models, demonstrating that the application of CNN–BiLSTM is highly efficacious for TLS anomaly detection without requiring content decryption.

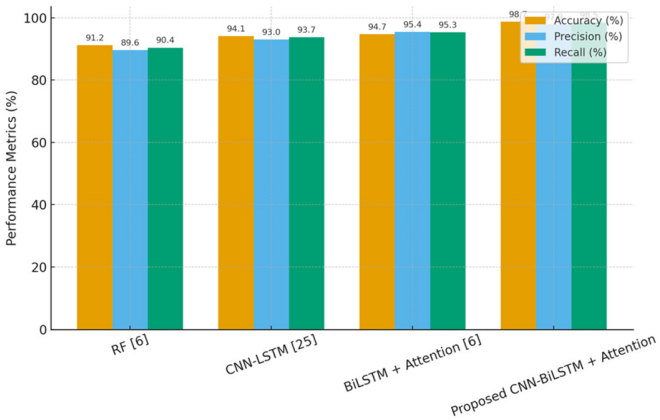

Fig. 3. Distribution of Performance Metrics across models.

Table I summarizes the performance comparison between the hybrid model and the baselines (RF, CNN–LSTM, BiLSTM). The hybrid model demonstrates a 4–9% enhancement in accuracy relative to the individual CNN and BiLSTM and a decrease in FPR by as much as 90%. The statistical t-test results indicate $p < 0.01$ when comparing hybrid to baselines, affirming the significance of the performance improvement.

The CNN model employed in this study demonstrates superior detection accuracy and offers interpretability via activation mapping on the convolutional layers. Within the TLS framework, CNN can extract spatial patterns from features including packet size distribution, JA3 fingerprint vectors, and cipher suite encoding, which are represented as input matrices. By examining the activations in the initial convolutional layers, one can discern local features that substantially aid in anomaly categorization.

TABLE I. PERFORMANCE COMPARISON OF THE HYBRID CNN–BILSTM MODEL WITH BASELINE MODELS

Model (Ref.)	Accuracy (%)	Precision (%)	Recall (%)	FPR (%)	Latency (ms)
RF [6]	91.2	89.6	90.4	5.8	12.5
CNN–LSTM [25]	94.1	93.0	93.7	3.9	13.2
BiLSTM + attention [6]	95.5	94.7	95.3	2.8	14.1
CNN–BiLSTM + attention (proposed)	98.7	97.9	98.5	1.4	12.9

A statistical study of feature importance from CNN was performed utilizing the Gradient-weighted Class Activation Mapping (Grad-CAM) technique, revealing that the most significant activation regions consistently emerged in the visual representations of flow length and JA3 entropy. This indicates that CNN can proficiently discern critical components indicative of TLS attacks, such as botnet communication or the utilization of obsolete ciphers. This CNN methodology allows the system to autonomously identify anomalous patterns in TLS information without the need for manual feature engineering, rendering it a robust method for interpretive DL-based NIDS.

Table II presents the ten primary features identified as determinants in anomaly detection on TLS traffic, based on their importance to classification performance. This study established that Packet Count, Flow Bytes to Server, and Flow Packets to Server are critically significant, as these features directly indicate the intensity and patterns of data transmission that frequently diverge in attack scenarios, including C2 and data exfiltration. Moderately important features, such as Flow Bytes to Client, Flow Packets to Client, Source Port, and Flow Duration, affect detection by indicating atypical traffic patterns, including uncommon ports or disproportionate connection lengths. Flow ID, Flow Age, and Flow Reason are classified as features of low significance because they have minimal impact on categorization. Nonetheless, they retain contextual relevance within the DL model. Utilizing the outcomes of this feature selection enables the system to concentrate the training process on the most representative attributes, hence enhancing the efficiency and accuracy of the CNN–BiLSTM model in identifying anomalies in encrypted TLS traffic.

TABLE II. FEATURE IMPORTANCE FOR ANOMALY DETECTION IN TLS TRAFFIC

Feature	Description	Importance
Packet Count	Total number of packets in a flow	High
Flow Bytes to Server	Total data sent to the server	High
Flow Packets to Server	Number of packets sent to the server	High
Flow Bytes to Client	Total data received from the server	Medium
Flow Packets to Client	Number of packets received from the server	Medium
Source Port	Source-assigned port number	Medium
Flow Duration	Total duration of the flow	Medium
Flow ID	Distinct flow identifier	Low
Flow Age	Time elapsed since flow initiation	Low
Flow Reason	Reason for flow termination (e.g., timeout, reset)	Low

IV. CONCLUSION

This study successfully developed and evaluated an efficient Network Intrusion Detection System (NIDS) for encrypted Transport Layer Security (TLS) traffic using a hybrid architecture of Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM). The model utilizes Suricata logs as its data source, enabling automatic feature selection via CNN and the capture of sequential patterns with BiLSTM, thus achieving enhanced performance relative to benchmark models, including Random Forest (RF), CNN–Long Short-Term Memory (LSTM), and individual BiLSTM.

The hybrid CNN–BiLSTM model demonstrates a detection accuracy of 98.7%, precision of 97.9%, recall of 98.5%, and a low False Positive Rate (FPR) of 1.4%, with an average detection latency of 12.9 ms, making it suitable for real-time deployment. Furthermore, features such as packet count, bytes transmitted to the server, and packets sent to the server contributed most significantly to classification performance.

The novelty of this study lies in the integration of CNN–BiLSTM with an attention mechanism specifically tailored for large-scale TLS metadata, the use of a one-year real-world TLS dataset comprising 30 million sessions, rarely available in prior research, and the application of dual feature selection methods—RF and CNN attention weights—to optimize both detection performance and computational efficiency.

The main contributions of this work include the development of a hybrid detection architecture that significantly outperforms traditional Machine Learning (ML) and single Deep Learning (DL) models, the provision of a publicly accessible large-scale TLS dataset to advance research in encrypted traffic analysis, and the demonstration of the model's capability for real-time deployment in high-speed corporate networks while maintaining high accuracy, low FPRs, and operational scalability.

REFERENCES

[1] R. Mohammad, F. Saeed, A. A. Almazroi, F. S. Alsubaei, and A. A. Almazroi, "Enhancing Intrusion Detection Systems Using a Deep Learning and Data Augmentation Approach," *Systems*, vol. 12, no. 3, Mar. 2024, Art. no. 79, https://doi.org/10.3390/systems12030079.

[2] A. A. Ghani and S. A. Alasadi, "A Deep Learning Algorithm to Cybersecurity: Enhancing Intrusion Detection with a Hybrid GRU and BiLSTM Model," *Engineering, Technology & Applied Science Research*, vol. 15, no. 3, pp. 23605–23612, Jun. 2025, https://doi.org/10.48084/etasr.10666.

[3] B. Anderson, S. Paul, and D. McGrew, "Deciphering malware's use of TLS (without decryption)," *Journal of Computer Virology and Hacking Techniques*, vol. 14, no. 3, pp. 195–211, Aug. 2018, https://doi.org/10.1007/s11416-017-0306-6.

[4] R. H. Altaie and H. K. Hoomod, "An Intrusion Detection System using a Hybrid Lightweight Deep Learning Algorithm," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16740–16743, Oct. 2024, https://doi.org/10.48084/etasr.7657.

[5] M. Jouhari and M. Guizani, "Lightweight CNN-BiLSTM based Intrusion Detection Systems for Resource-Constrained IoT Devices," in *2024 International Wireless Communications and Mobile Computing*, Ayia Napa, Cyprus, 2024, pp. 1558–1563, https://doi.org/10.1109/IWCMC61514.2024.10592352.

[6] J. Zhang, X. Zhang, Z. Liu, F. Fu, Y. Jiao, and F. Xu, "A Network Intrusion Detection Model Based on BiLSTM with Multi-Head

Attention Mechanism," *Electronics*, vol. 12, no. 19, Oct. 2023, Art. no. 4170, https://doi.org/10.3390/electronics12194170.

[7] H. Benaddi, M. Jouhari, and O. Elharrouss, "A lightweight hybrid approach for intrusion detection systems using a chi-square feature selection approach in IoT," *Internet of Things*, vol. 32, Jul. 2025, Art. no. 101624, https://doi.org/10.1016/j.iot.2025.101624.

[8] X. Qiu, G. Yan, and L. Yin, "CLSTM-MT (a Combination of 2-Conv CNN and BiLSTM Under the Mean Teacher Collaborative Learning Framework): Encryption Traffic Classification Based on CLSTM (a Combination of 2-Conv CNN and BiLSTM) and Mean Teacher Collaborative Learning," *Applied Sciences*, vol. 15, no. 9, May 2025, Art. no. 5089, https://doi.org/10.3390/app15095089.

[9] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Systems with Applications*, vol. 185, Dec. 2021, Art. no. 115524, https://doi.org/10.1016/j.eswa.2021.115524.

[10] R. Ben Said, Z. Sabir, and I. Askerzade, "CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software-Defined Networking With Hybrid Feature Selection," *IEEE Access*, vol. 11, pp. 138732–138747, 2023, https://doi.org/10.1109/ACCESS.2023.3340142.

[11] A. N. Abdullah, "Development of an Intrusion Detection System using an Ensemble Voting Machine Learning Technique," *Engineering, Technology & Applied Science Research*, vol. 15, no. 3, pp. 23917–23922, Jun. 2025, https://doi.org/10.48084/etasr.10764.

[12] F. Amiri, M. Rezaei Yousefi, C. Lucas, A. Shakery, and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1184–1199, Jul. 2011, https://doi.org/10.1016/j.jnca.2011.01.002.

[13] M. S. Akhtar and T. Feng, "Deep Learning-Based Framework for the Detection of Cyberattack Using Feature Engineering," *Security and Communication Networks*, vol. 2021, no. 1, Dec. 2021, Art. no. 6129210, https://doi.org/10.1155/2021/6129210.

[14] K. Mala and H. S. Annapurna, "The Zoneout Regularized Gated Recurrent Unit Algorithm for Network Intrusion Detection with Class Imbalance Mitigation," *Engineering, Technology & Applied Science Research*, vol. 15, no. 4, pp. 24758–24763, Aug. 2025, https://doi.org/10.48084/etasr.11823.

[15] A.-A. Maiga, E. Ataro, and S. Githinji, "Balancing Data Privacy and 5G VNFs Security Monitoring: Federated Learning with CNN + BiLSTM + LSTM Model," *Journal of Electrical and Computer Engineering*, vol. 2024, no. 1, Mar. 2024, Art. no. 5134326, https://doi.org/10.1155/2024/5134326.

[16] N. O. Aljehane, H. A. Mengash, S. B. H. Hassine, F. A. Alotaibi, A. S. Salama, and S. Abdelbagi, "Optimizing intrusion detection using intelligent feature selection with machine learning model," *Alexandria Engineering Journal*, vol. 91, pp. 39–49, Mar. 2024, https://doi.org/10.1016/j.aej.2024.01.073.

[17] S. Z. Majidian, S. TaghipourEivazi, B. Arasteh, and A. Ghaffari, "Optimizing random forests to detect intrusion in the Internet of Things," *Computers and Electrical Engineering*, vol. 120, Dec. 2024, Art. no. 109860, https://doi.org/10.1016/j.compeleceng.2024.109860.

[18] H. Liu, M. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 703–715, May 2019, https://doi.org/10.1109/JAS.2019.1911447.

[19] K. Ren, Y. Zeng, Z. Cao, and Y. Zhang, "ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model," *Scientific Reports*, vol. 12, no. 1, Sep. 2022, Art. no. 15370, https://doi.org/10.1038/s41598-022-19366-3.

[20] J. Lansky *et al.*, "Deep Learning-Based Intrusion Detection Systems: A Systematic Review," *IEEE Access*, vol. 9, pp. 101574–101599, 2021, https://doi.org/10.1109/ACCESS.2021.3097247.

[21] S. Elsayed, K. Mohamed, and M. A. Madkour, "A Comparative Study of Using Deep Learning Algorithms in Network Intrusion Detection," *IEEE Access*, vol. 12, pp. 58851–58870, 2024, https://doi.org/10.1109/ACCESS.2024.3389096.

[22] T.-P. Nguyen, J. Cho, and D. Kim, "Semi-supervised intrusion detection system for in-vehicle networks based on variational autoencoder and adversarial reinforcement learning," *Knowledge-Based Systems*, vol. 304, Nov. 2024, Art. no. 112563, https://doi.org/10.1016/j.knosys.2024.112563.

[23] P. Soltanzadeh and M. Hashemzadeh, "RCSMOTE: Range-Controlled synthetic minority over-sampling technique for handling the class imbalance problem," *Information Sciences*, vol. 542, pp. 92–111, Jan. 2021, https://doi.org/10.1016/j.ins.2020.07.014.

[24] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, and J. Yu, "ET-BERT: A Contextualized Datagram Representation with Pre-training Transformers for Encrypted Traffic Classification," in *Proceedings of the ACM Web Conference 2022*, Virtual Event, Lyon, France, 2022, pp. 633–642, https://doi.org/10.1145/3485447.3512217.

[25] P. R. Kanna and P. Santhi, "Hybrid Intrusion Detection using MapReduce based Black Widow Optimized Convolutional Long Short-Term Memory Neural Networks," *Expert Systems with Applications*, vol. 194, May 2022, Art. no. 116545, https://doi.org/10.1016/j.eswa.2022.116545.

[26] Kirubavathi. G and A. R. Nair, "Hybrid Deep Learning framework-based intrusion detection system for the Internet of Things," in *2024 International Conference on Intelligent Systems for Cybersecurity*, Gurugram, India, 2024, pp. 1–6, https://doi.org/10.1109/ISCS61804.2024.10581228.

[27] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *2017 International Conference on Advances in Computing, Communications and Informatics*, Udupi, India, 2017, pp. 1222–1228, https://doi.org/10.1109/ICACCI.2017.8126009.

[28] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," *Computer Communications*, vol. 199, pp. 113–125, Feb. 2023, https://doi.org/10.1016/j.comcom.2022.12.010.

[29] A. Bhardwaj, R. Tyagi, N. Sharma, A. Khare, M. S. Punia, and V. K. Garg, "Network intrusion detection in software defined networking with self-organized constraint-based intelligent learning framework," *Measurement: Sensors*, vol. 24, Dec. 2022, Art. no. 100580, https://doi.org/10.1016/j.measen.2022.100580.

[30] M. Rahouti, K. Xiong, Y. Xin, S. K. Jagatheesaperumal, M. Ayyash, and M. Shaheed, "SDN Security Review: Threat Taxonomy, Implications, and Open Challenges," *IEEE Access*, vol. 10, pp. 45820–45854, 2022, https://doi.org/10.1109/ACCESS.2022.3168972.

[31] I. Aliyu, M. C. Feliciano, S. Van Engelenburg, D. O. Kim, and C. G. Lim, "A Blockchain-Based Federated Forest for SDN-Enabled In-Vehicle Network Intrusion Detection System," *IEEE Access*, vol. 9, pp. 102593–102608, 2021, https://doi.org/10.1109/ACCESS.2021.3094365.

[32] A. Chiras, A. Peratikou, and S. Stavrou, "Increasing Security of Containerized Blockchain using SDN," in *2024 Panhellenic Conference on Electronics & Telecommunications*, Thessaloniki, Greece, 2024, pp. 1–5, https://doi.org/10.1109/PACET60398.2024.10497057.

[33] S. Faizullah, M. A. Khan, A. Alzahrani, and I. Khan, "Permissioned Blockchain-Based Security for SDN in IoT Cloud Networks," in *2019 International Conference on Advances in the Emerging Computing Technologies*, Al Madinah Al Munawwarah, Saudi Arabia, 2020, pp. 1–6, https://doi.org/10.1109/AECT47998.2020.9194181.

[34] M. H. Rifat, A. Islam Ananna, T. Intesir Ahmed, S. Akter, and N. Mansoor, "Blockchain-Based Controller Recovery and SDN Packet Filtering Scheme for Softwarized UAVs," in *2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems*, Dhaka, Bangladesh, 2024, pp. 1–5, https://doi.org/10.1109/iCACCESS61735.2024.10499453.

[35] C. Kaushik, D. VarunTeja, M. S. Krishna, and S. Jaavali, "DDoS Attack Detection and Mitigation Using Mininet and RYU Controller in SDN Environment," in *2024 15th International Conference on Computing Communication and Networking Technologies*, Kamand, India, 2024, pp. 1–7, https://doi.org/10.1109/ICCCNT61001.2024.10724700.

[36] A. Jain, D. Kumar Khatri, A. Ayyagiri, C. Mokkapati, V. B. R. Bhimanapati, and L. H. Alzubaidi, "Secure and Scalable IoT Networks: Optimizing Blockchain and SDN for Smart Environments," in *2024 4th*

International Conference on Blockchain Technology and Information Security, Wuhan, China, 2024, pp. 338–344, https://doi.org/10.1109/ICBCTIS64495.2024.00060.

[37] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the Deployment of Machine Learning Solutions in Network Traffic Classification: A Systematic Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1988–2014, Apr. 2019, https://doi.org/10.1109/COMST.2018.2883147.

[38] M. Shen *et al.*, "Machine Learning-Powered Encrypted Network Traffic Analysis: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 791–824, Jan. 2023, https://doi.org/10.1109/COMST.2022.3208196.

[39] B. Yan *et al.*, "Graph Mining for Cybersecurity: A Survey," *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 2, Nov. 2023, Art. no. 47, https://doi.org/10.1145/3610228.

[40] S. Rezaei and X. Liu, "Deep Learning for Encrypted Traffic Classification: An Overview," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 76–81, May 2019, https://doi.org/10.1109/MCOM.2019.1800819.

[41] B. Ji *et al.*, "Survey of Secure Communications of Internet of Things with Artificial Intelligence," *IEEE Internet of Things Magazine*, vol. 5, no. 3, pp. 92–99, Sep. 2022, https://doi.org/10.1109/IOTM.001.2100178.

[42] Ö. Aslan, S. S. Aktuğ, M. Ozkan-Okay, A. A. Yilmaz, and E. Akin, "A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions," *Electronics*, vol. 12, no. 6, Mar. 2023, Art. no. 1333, https://doi.org/10.3390/electronics12061333.

[43] A. Thakkar and R. Lohiya, "A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions," *Artificial Intelligence Review*, vol. 55, no. 1, pp. 453–563, Jan. 2022, https://doi.org/10.1007/s10462-021-10037-9.

[44] M. S. K. K, T. Sree, S. V. D, Y. S. S. Harsha, and N. Rajagopalan, "Suricata-Based Intrusion Detection and Isolation System for Local Area Networks," in *2024 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication*, Karaikal, India, 2024, pp. 1–5, https://doi.org/10.1109/IConSCEPT61884.2024.10627890.

[45] B. Omarov, O. Auelbekov, A. Suliman, and A. Zhaxanova, "CNN-BiLSTM Hybrid Model for Network Anomaly Detection in Internet of Things," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 3, pp. 436–444, Mar. 2023, https://doi.org/10.14569/IJACSA.2023.0140349.

[46] Z. Liu *et al.*, "Using Embedded Feature Selection and CNN for Classification on CCD-INID-V1—A New IoT Dataset," *Sensors*, vol. 21, no. 14, Jul. 2021, Art. no. 4834, https://doi.org/10.3390/s21144834.

[47] M. S. Alshehri, J. Ahmad, S. Almakdi, M. A. Qathrady, Y. Y. Ghadi, and W. J. Buchanan, "SkipGateNet: A Lightweight CNN-LSTM Hybrid Model With Learnable Skip Connections for Efficient Botnet Attack Detection in IoT," *IEEE Access*, vol. 12, pp. 35521–35538, 2024, https://doi.org/10.1109/ACCESS.2024.3371992.

[48] S. Ebadinezhad, N. N. Nia, N. Shirzad, and N. K. Osemeha, "Enhancing Intrusion Detection Systems Using RNN, LSTM, and Hybrid RNN-LSTM Models," in *2025 International Conference on Machine Learning and Autonomous Systems*, Prawet, Thailand, 2025, pp. 1108–1115, https://doi.org/10.1109/ICMLAS64557.2025.10968214.

[49] A. A. A. Mohammed, "Improving Intrusion Detection Systems by Using Deep Learning Methods on Time Series Data," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 19267–19272, Feb. 2025, https://doi.org/10.48084/etasr.9417.

[50] H. Y. I. Khalid and N. B. I. Aldabagh, "A Survey on the Latest Intrusion Detection Datasets for Software Defined Networking Environments," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13190–13200, Apr. 2024, https://doi.org/10.48084/etasr.6756.

[51] M. H. Bhavsar, Y. B. Bekele, K. Roy, J. C. Kelly, and D. Limbrick, "FL-IDS: Federated Learning-Based Intrusion Detection System Using Edge Devices for Transportation IoT," *IEEE Access*, vol. 12, pp. 52215–52226, 2024, https://doi.org/10.1109/ACCESS.2024.3386631.

[52] H. Muttaqien, "Suricata Logs." Kaggle. [Online]. Available: www.kaggle.com/datasets/muttaqien19/dataset-suricata-logs.

[53] A. Ferriyan, A. H. Thamrin, K. Takeda, and J. Murai, "Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic," *Applied Sciences*, vol. 11, no. 17, Sep. 2021, Art. no. 7868, https://doi.org/10.3390/app11177868.

[54] S. N. and A. Haldorai, "Efficient Intrusion Detection System Data Preprocessing Using Deep Sparse Autoencoder with Differential Evolution," *IET Information Security*, vol. 2024, no. 1, Aug. 2024, Art. no. 9937803, https://doi.org/10.1049/2024/9937803.