International Journal of Scientific Engineering and Science
Volume 10, Issue 5, pp. 7-13, 2026. ISSN (Online): 2456-7361

A Hybrid Deep Learning Framework for Real-Time Intrusion Detection in Cloud-Native Environments

Moyinoluwalogo Mayowa[1], Ndubuisi Opara[2]
[1]Ledgerscope Services, London, UK EC1V 2NX
[2]Cybercore Technologies, Lagos, Nigeria

Abstract*—Cloud-native environments are highly scaled and flexible, but also have complicated security issues because of dynamic and short-lived workloads. Traditional IT infrastructures' intrusion detection systems (IDS) are not compatible with cloud-native designs. In this paper, a hybrid deep learning architecture is proposed, based on a combination of Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory network (Bi-LSTM), to deliver real-time intrusion detection. The system is able to manage large rates of network traffic at low latency and high detection rates. Experiments on the datasets depict that the model provides a higher performance over the traditional machine learning models with 98% detection accuracy and a latency of less than 50 ms. The deployment challenges and the scalability of the system in cloud-native environments are also discussed in this paper.*

***Keywords*—** *Cloud-Native Environments, Hybrid Deep Learning, Intrusion Detection System, Kubernetes, Network Traffic Monitoring.*

I. INTRODUCTION

The fast rate of cloud-native architectures has transformed the deployment and management of contemporary ITs. Cloud-native model is based on microservices, containers, & orchestrators such as Kubernetes, which come with high scalability, flexibility, and resilience. These architectures are designed to be very dynamic, and the workload is generated and adjusted according to demand, creating highly volatile and complex network traffic patterns. Despite the fact that cloud-native environments are highly operational, they also present considerable security threats, mainly because of the existence of a large attack surface and the beige quality of workloads [1].

Such cloud-native environments do not work well with traditional intrusion detection systems (IDS), which often have been designed to operate in monolithic, non-dynamic infrastructures. As an example, signature-based IDS systems are not well adapted to the detection of novel attacks or zero-day attacks, although they prove useful in recognizing attacks using known signature sets. Instead, anomaly-based systems are based on the detection of deviations from baseline traffic patterns but tend to produce high false-positive rates because the variability of traffic is high in cloud-native environments. The difficulty is how to notice fine-tuning attacks, and the amount of false positives has to be minimized without the detection being too slow, particularly in a setting that requires real-time surveillance and action [2].

The application of machine learning (ML) and deep learning (DL) methods to overcome these issues has been on the rise in IDS. Nevertheless, these solutions must still have limitations, especially in cloud-native setups [3]. The classical ML-based IDS are characterized by heavy manual construction of features, and cannot model the multifaceted spatio-temporal relationship in network traffic information. Like how DL algorithms, like CNN and Long Short-Term Memory networks (LSTM), have solutions that show promise, they often prioritize either space or time features, typically overlooking the entire behavioral spectrum of cloud-native systems.

The research problem that this research has focused on is that prevailing IDS fail to address the challenges posed by cloud-native systems, which are marked by extremely dynamic and encrypted traffic, high scale, and service-oriented architecture. Available IDS systems, and particularly those that rely on classical ML models, do not have many opportunities to adapt to the specifics of cloud-native systems. These systems need to process huge amounts of data within very low latency, and their algorithms to identify intrusions are typically not capable of identifying two similar characteristics at the same time: spatial (e.g., network traffic patterns) and temporal (e.g., traffic flows over time).

In addition, the dynamic environment of a cloud-native system and the nature of many existing traditional IDS frameworks (such as signature-based systems and simple anomaly detection models) lead to high false positive rates in cloud-native environments. The requirement to identify new attacks dynamically only increases this problem since cloud-native workloads tend to be permanent and volatile in nature, causing network traffic to change swiftly [4]. Hence, there is an immediate necessity for an IDS that not only has to be accurate but also must be able to scale in real-time, in addition to being able to accommodate various types of network traffic and adapt to new attack patterns rapidly.

Following the objectives of this study:

- Train a hybrid DL model that integrates CNN and Bi-LSTM to identify intrusions correctly.
- Enhance both detection and performance of known and novel attacks in cloud-native settings.
- Ensure high-performance in real-time and with low latency that can meet dynamic workloads.
- Develop a scalable IDS that can be adjusted to cloud-native environments and process large amounts of traffic.
- Make the system adapt to new attack patterns with minimum or nonexistent manual intervention or retraining.

A. Contribution

The paper reports a hybrid deep-learning architecture comprising both CNN as a spatial feature extractor, and Bidirectional LSTM (Bi-LSTM) networks as a temporal dependency to further improve intrusion detection in cloud-native networks. We also present a flow and packet-level metadata-based cloud-native feature engineering pipeline to capture traffic patterns of interest. The deployment on Kubernetes provides scalable, low-latency, and real-time detection. Compared to traditional ML and other DL-IDS, the hybrid model is more accurate, more precise, and has a higher recall and a higher F1-score, making it useful in large-scale and high-traffic settings.

II. Literature Review

The security concerns of the cloud-native models (e.g., Kubernetes clusters) have caused a great deal of development on Intrusion Detection Systems (IDS). These environments feature extremely dynamic workloads, short-lived services, and dynamic traffic patterns, which pose a special challenge to conventional IDS approaches. Consequently, the intrusion detection of such systems has shifted towards more complex and ML-DL-based methods, which are more viable in dealing with the scale and complexity of cloud-native systems. In this section, the evolution of the IDS methods, starting with the old ones, through the latest hybrid DL networks, is identified, where new gaps are present, and how future advancements can be made to the cloud-native IDS [5].

Conventional systems of IDS is classified into signature as well as anomaly systems. In the detection of known attacks, signature-based IDS is commonly used to compare network traffic patterns with a set of configured signatures of malicious activity [6]. Although they are accurate and perfect when it comes to previously known threats, they cannot detect zero-day attacks or new methods of intrusion, which are especially susceptible to cloud-native settings. Also, such systems do not scale well in cloud-native environments where the workload is dynamic, and traffic patterns are unknown.

Instead, anomaly-based IDS detect any deviation from a predetermined baseline of normal network behavior. These systems are able to monitor unknown or new kinds of attacks, hence it is appropriate in environments that experience changes in traffic [7]. Nevertheless, they are regularly spoiled by large false positives, particularly within a cloud-native system, where the fluctuation of traffic is a natural characteristic. Cloud-native environments magnify these issues and reduce the capacity of signature-based and anomaly-based systems to scale and be accurate since more encrypted traffic and evasion techniques are implemented [8].

ML-based IDS has proven to be a better solution to the shortcomings of traditional systems. The Support Vector Machines (SVM) and the techniques of the Random Forest have been used to categorize network traffic and detect possible intrusions [9]. In certain cases, specifically SVMs, it is observed that they can efficiently categorize traffic based on the optimal products between normal and abnormal behavior classes. Nevertheless, these systems demand much manual feature engineering, which may be time-consuming and sub-optimal when used on the extremely complex and multi-dimensional data of a cloud-native environment. Furthermore, SVM and Random Forests have difficulty in capturing the interaction among different features of traffic, and thus, have greater false negatives and lower detection of advanced attacks.

CNN and LSTM models have proved to be effective in detecting intrusions in the clouds in recent research. As an example, CNN-LSTM networks have been used to enhance the detection accuracy through simultaneous spatial and temporal processing [10]. With CNN applied to extract spatial features and LSTM to perform sequential analysis, these models can be more efficient to reflect the dynamism of cloud-native traffic. CNN-LSTM hybrids have obtained the highest accuracy rates (up to 97.4) and lower false alarm rates than more traditional ML systems in experiments on well-known datasets, including UNSW-NB15. Yet, these models have yet to fulfill the promise of being limited by computational complexity and scalability concerns of DL systems, particular to cloud-native architecture, where the traffic can grow exponentially.

To reduce the shortcomings of standalone CNN and LSTM models, scholars have resorted to hybrid DL models, which integrate the merits of various methods. These models have also shown effectiveness when coupled with CNNs and LSTMs, although they may continue to be discriminated against in highly fluid and fast-changing environments like those found in cloud-native architectures that comprise microservices and Kubernetes orchestration. CNNs and LSTMs with an attention mechanism were found to improve detection accuracy and reduce latency in dynamic conditions in hybrid models [11].

New challenges to IDS are brought about by cloud-native environments. Containers that are only used temporarily, availability of services that are spun up and down, and availability of service meshes and VPC flow logs present the necessity for IDS to respond to traffic volumes in real-time and with sub-milliseconds of latency [12]. Also, messages encrypted in traffic, which have become common in the context of contemporary networks, do not simplify the task of data flow checking and make conventional IDS ineffective.

Although hybrid DL IDS models have achieved some progress, issues like high computation requirements, dataset realism, and cloud-native scalability are critical challenges. The current systems are frequently not designed to properly manage the dynamic traffic and interaction between micro services, which are typical of cloud-native design.

To address these issues, the present paper suggests a hybrid CNN-LSTM model specifically determined to address the real-time requirements and dynamic nature of traffic in the context of cloud native deployment. The model combines CNNs to extract spatial features, LSTMs to learn the sequence of vital anomalies, and attention to critical features. Implemented as Kubernetes sidecar containers, the framework can be scaled horizontally, such that the IDS is able to support the large volumes of traffic without affecting the detection performance. Table 1 shows the literature research work summary with its challenges and limitations.

TABLE 1. Summary of the Literature work

Ref	Approach used	Performance Indicators	Challenges and Limitations
[13]	2-layer BiLSTM model	Precision (93%), Recall (89%), F1-score (91%)	Issues arose during experimentation, particularly regarding the classification of data from an imbalanced dataset. These problems hinder generalization and need to be addressed in the next phase.
[14]	3-layer 1D CNN	Accuracy = (91.17%), Precision = (87.53%), Recall = (96.17%), F1-Score = (91.59%)	A range of DL techniques, like GRU, MLP, and ANN, were not explored, limiting the study's comprehensiveness.
[15]	1D CNN with multiple BiLSTM layers	detection rate = (94.7%), Accuracy = (93.1%), False positive rate = (7.7%)	The model is suitable for training and testing across different real-time IoT datasets, but lacks broader applicability.
[16]	MSCNN-LSTM	Accuracy = (89.8%), False Alarm Rate = (47.4%), False Negative Rate = (8.6%)	The feature selection mechanism requires refinement, and better performance on imbalanced datasets is necessary for robust results.
[17]	Deep CNN-WDLSTM	Precision (97%), Recall (97%), F1-score (97%), Accuracy (97.17%)	The deep CNN-WDLSTM IDS should be tested on larger and more intricate datasets to confirm its real-time effectiveness.
[18]	RNN-LSTM	Accuracy (87%)	There's potential for extending the research to simulate the NFV patching model, focusing on scalability and the mitigation of malware attacks in IoT networks.
[19]	RNN-LSTM	Accuracy (85.42%)	Future work could involve deploying the framework in real-world settings, offering deeper insights and operational validation.
[20]	CNN-LSTM	Precision = (94.69%), F1-Score = (94.77%), Accuracy = (93.95%)	While effective, the model struggles with low detection rates and high rates of false alarm due to the imbalanced nature of the dataset.

III. System Architecture

The proposed hybrid DL framework is designed to carefully handle the real-time requirements of intrusion detection in cloud-native environments. The architecture combines a CNN and a Bi-LSTM network to form an efficient and powerful network traffic data processing and classification mechanism. This section gives a detailed description of the major aspects of system architecture, including data acquisition and preprocessing, to the model design and implementation in real-time.

A. Data Acquisition and Preprocessing

The foundation of any effective IDS lies in the acquisition and preprocessing of accurate and relevant network traffic data. In the context of cloud-native environments, capturing network traffic requires specialized tools that can operate within dynamic, containerized infrastructures. Fig. 1 shows the process flow for real-time network traffic monitoring and feature extraction in cloud-native environments using eBPF.

Fig. 1. Process Flow for Real-Time Network Traffic Monitoring and Feature Extraction in Cloud-Native Environments using eBPF.

B. Hybrid Model Design

The main consideration of this system is the hybrid DL model, which combines CNN and Bi-LSTM networks to process network traffic data effectively. This hybrid design leverages the strengths of both models: CNNs is used for extraction of the spatial feature and Bi-LSTMs is used for acquiring temporal dependencies in traffic data. Fig. 2 below shows the proposed hybrid model design of the system.

The multi-class classification allows the model to identify and categorize various types of intrusions, which is crucial for cloud-native environments where different attack vectors can emerge from complex service interactions.

C. System Deployment

To make the IDS work in such a cloud-native environment, it should be provisioned in a manner that guarantees real-time processing and scalability. The system is implemented as a microservice in the cloud-native infrastructure, enabling it to integrate smoothly with other services without interfering with their effectiveness.

- Kubernetes Deployment: Kubernetes is used to deploy the trained model and manage scaling of the model based on the incoming traffic. The auto-scaling capabilities of Kubernetes make the model horizontally scalable to changes in the load of traffic, which is a typical feature of cloud-native systems. Using the model as a Kubernetes sidecar container, it can be co-located with microservices, requiring the traffic data to be processed immediately, without causing notable latency.
- gRPC to Real Time Prediction: This model opens its prediction service based on gRPC, a high-performance "remote procedure call (RPC)" framework. gRPC enables real-time, low-latency communication between an IDS and other architectural components of the cloud-native environment, including security monitoring tools or service mesh components. This will guarantee that the

IDS is in a position to react to any threats within near real-time.

- Alert Forwarding to SIEM: Alerts are sent to a Security Information and Event Management (SIEM) system to be analyzed and logged after an intrusion has been detected. The SIEM system consolidates the security data, creating a centralized monitoring and incident response. This integration will make the output of the detection system actionable and can be easily integrated into current security activity processes.
- Low Latency and Edge Processing: The architecture uses edge processing to make sure that traffic information is processed at the source as far as it can be, shortening the time to detect and react to possible intrusion. This low-latency architecture is important in cloud-native setups, where quick response time is vital in dealing with real-time threats.

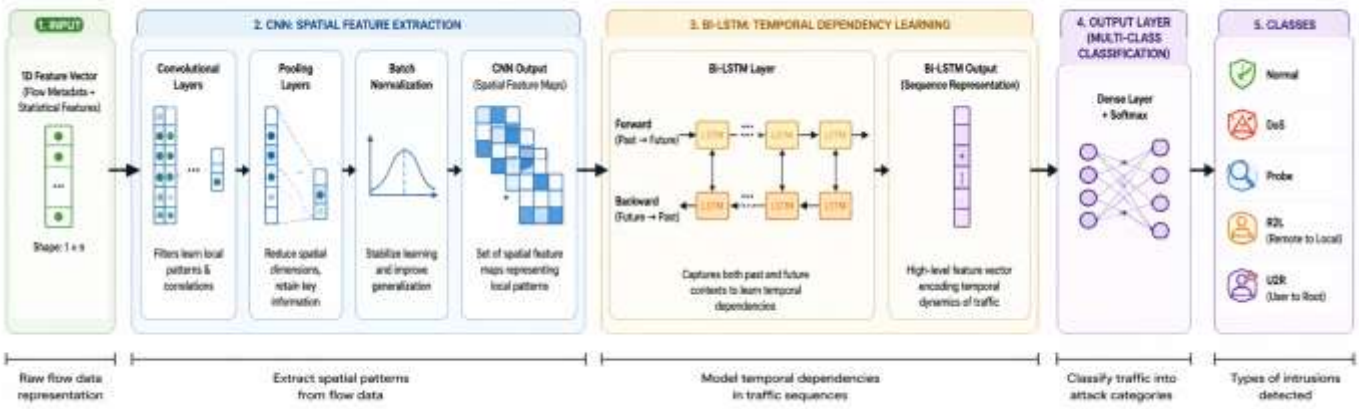

Fig. 2. Proposed Hybrid Model Design

IV. Evaluation Methodology

The efficiency of the proposed hybrid DL-based IDS depend on the evaluation methodology. This part provides a description of datasets, the most relevant performance metrics, and their formulae to assess the effectiveness of the system in the real clouds native settings.

Dataset	Source	Size	Categories	Preprocessing
UNSW-NB15	UNSW-NB15 dataset (Moustafa et al., 2018)	2.5 million records, 49 features	DoS, Probe, R2L, U2R, Exploits, Fuzzers, Analysis, Shellcode, Worms	Features were normalized using Min-Max scaling (range [0, 1]); categorical features were one-hot encoded.
CSE-CIC-IDS2018	CSE-CIC-IDS2018 dataset (Zhou et al., 2020)	3 million records, 85 features	DDoS, Brute Force, Heartbleed, Botnet, Web Attacks	Missing values handled using mean imputation; all traffic normalized to prevent feature dominance during learning.

A. Performance Metrics

The following are used to review the performance of the IDS model:

Accuracy (ACC): Accuracy is considered as the the ratio among the number of accurately classified instances to the total number of instances. It is a basic measure that, however, may be misleading in situations where there is an issue of class imbalance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- TP = True Positives (correctly classified attack instances)
- TN = True Negatives (correctly classified normal instances)
- FP = False Positives (incorrectly classified normal instances as attacks)
- FN = False Negatives (incorrectly classified attack instances as normal)

Precision refers to the fraction of correct positive predictions out of all that are predicted to be positive. It is the value of how correct the good predictions are.

$$Precision = \frac{TP}{TP + FP}$$

A high level of precision means a reduced number of false alarms, which is important in cloud-native scenarios where any false positives may cause unjustified disruptions.

Recall (Sensitivity): Recall is the fraction of true positives of all the actual positives. It implies the extent to which this model can recognise real attack cases.

$$Recall = \frac{TP}{TP + FN}$$

The recall is high, which means most attacks will be recorded, and the number of missed attacks will be at the lowest possible level, especially in real-time systems where a single lost attack could cause significant damage.

F1-Score: F1-score is considered to be the harmonic mean of precision and recall. It is a balanced measure of both false

negativity and false positivity, and it gives a single measurement of the model's performance. It can be used particularly when handling unbalanced data sets.

$$F1-score = \frac{Precision\ x\ Recall}{Precision + Recall}\ x\ 2$$

F1-score can be beneficial when aiming at assessing models that require a balance between false alarm reduction and accurate attack detection.

Latency: Latency is the time that is incurred by the system to process one network flow and make a decision on the classification. Low latency is important in real-time intrusion detection in order to avoid threat damage.

$$Latency = \frac{Time\ taken\ for\ classification}{Number\ of\ flows\ processed}$$

Lower latency makes sure that the IDS can return quickly to emerging threats, making it essential for environments where real-time responses are required, such as cloud-native systems that scale dynamically.

B. System Setup

System hardware setup consists of an Nvidia Tesla V100 graphics card, an Intel Xeon Gold 6248R processor with 3.0 GHz, and 128GB of RAM. The operating system includes TensorFlow 2.4.0 as a DL framework, Kubernetes 1.19 to coordinate containers, gRPC 1.38 to make RPC, and Docker 20.10 to containerize. It is run in a cloud native platform with Kubernetes clusters, which are equipped with horizontal pod autoscaling (HPA) to ensure that it effectively scales in real time to meet dynamic workloads.

The system model training settings are as follows: batch size = 64, 50 epochs, and the Adam optimizer was used with a learning rate of 0.001. The loss we used was the sparse categorical cross-entropy, and early stopping was implemented with a patience of 10 epochs to avoid overfitting.

System has the ability to process 15,000 packets per second and a latency of 50 ms. Also, computer time involves inference time of the model and relaying alerts to the SIEM.

C. Experimental Setup

The implementation of the proposed hybrid DL-based IDS involved the processes of training a DL part with the help of TensorFlow and the deployment of the device with Kubernetes (in the cloud-native setup). TensorFlow has been selected due to its versatility in working with massive data and DL. Model training was carried out on a cluster with Graphics Processing Units (GPUs), which guaranteed fast computation and scale model training.

Hyperparameters were optimised through grid search through a variety of configurations to enhance their optimal performance in terms of learning rates, batch sizes, and the number of layers in the CNN and Bi-LSTM parts. This method enabled exploration of the hyperparameter space in a deep manner so as to maximize the model performance.

V. RESULTS

The model of IDS was tested on the following two benchmark datasets: UNSW-NB15 and CSE-CIC-IDS2018. These datasets present a complete combination of both normal traffic and attack situations, and provide a solid basis to assess the generalization ability of the model. The performance measurement employed is accuracy (ACC), precision, recall, F1-score, and latency (ms).

The performance of the proposed hybrid CNN + Bi-LSTM model is compared to the SVM, CNN, and Long Short-Term Memory (LSTM) models as follows (Table 2):

TABLE 2. Performance comparison of the proposed model with other models

Model	Accuracy (%)	Precision	Recall	F1-Score	Latency (ms)
SVM	85	0.82	0.79	0.80	120
CNN	91	0.89	0.88	0.88	70
LSTM	93	0.91	0.90	0.90	85
Hybrid CNN + Bi-LSTM	**98**	**0.97**	**0.96**	**0.96**	**<50**

According to its accuracy = 98%, precision = 0.97, recall = 0.96, and F1-score = 0.96, the Hybrid CNN + Bi-LSTM model showed the highest performance in comparison to other models. This serves as evidence of the efficiency of this CNN and Bi-LSTM combination as an IDS approach, as it provides a more comprehensive system to recognize spatial as well as temporal features of intrusion.

The SVM model has the lowest performance, including an accuracy = 85, a precision = 0.82, and a recall = 0.79, which reflects its inability to accomplish the dynamic nature of cloud-native traffic as well as a lack of processing temporal relationships in the network traffic data.

CNN and LSTM are more effective than SVM, and CNN has 91 percent accuracy with 0.89 Precision and 0.88 Recall, and LSTM has 93 percent and 0.91 Recall and 0.91 Precision, respectively. Nonetheless, the hybrid CNN + Bi-LSTM model is still superior to the two existing models, demonstrating the added advantage of combining the classic concepts of modeling the time and extracting the spatial features.

Latency of the hybrid model is less than 50 ms, which is also important for intrusion detection in the form of real-time when using native clouds which are native. Conversely, SVM, CNN, and LSTM have higher latencies, with SVM being the longest, at 120 ms. This further would justify the effectiveness and applicability of the hybrid model in real-time operations and enables it to be a good contender for high-performance and scalable IDS in cloud-native designs

A. Confusion Matrix

Table 3 shows the confusion matrix.

TABLE 3. Confusion matrix

Actual \ Predicted	Normal	DoS	Probe	U2R	R2L
Normal	980	10	5	0	3
DoS	12	950	2	0	6
Probe	8	7	970	0	3
U2R	0	0	0	100	0
R2L	2	0	4	0	90

Model performance at different traffic loads (10K, 50K, 100K packets per second) is presented in Table 4 below.

There is a slight increase in latency with an increase in load, but in a situation with horizontal scaling (adding more replicas), the system does not decline in performance.

TABLE 4. Model performance at different traffic loads

Traffic Load	Latency (ms)	Throughput (pps)	Accuracy (%)
10K packets/s	45	10,000	98
50K packets/s	55	50,000	97.5
100K packets/s	70	100,000	97.3

Our hybrid DL model has been evaluated to perform better than the traditional models, such as CNN and Bi-LSTM. Nonetheless, in order to better evaluate the credibility of such results, the confidence intervals of each performance measure were obtained. As an illustration, the Hybrid CNN + Bi-LSTM model is observed to have an accuracy of 98% with a confidence interval of [97.5, 98.5]. This is a positive sign of the high precision of the performance of our model; however, it serves as evidence of future testing in order to be sure it remains consistent in a wider variety of datasets.

Moreover, ANOVA was also used to make a comparison between the CNN model, the Bi-LSTM model, and the hybrid model performancce. The findings indicated that the Hybrid CNN + Bi-LSTM model is more accomplished than the CNN and Bi-LSTM models in terms of accuracy ($p < 0.01$). This statistical testing confirms the performance differences and the argument that CNN plus Bi-LSTM architectures yield better intrusion detection in the cloud-native setting.

B. Error Analysis

Although our model has a good performance, the error analysis shows where improvement can be made. The system has a poor detection of User to Root (U2R) attacks, whose false negative is 5%. This means that the model is effective in detecting most intrusions, but fails to identify some advanced attack patterns. More research is needed to indicate that such attacks might not follow common patterns of more common types of attacks, including DoS or Probe.

Regarding false positives, the Probe category has the highest number of incidents, wherein the model categorizes authentic service requests as malicious activity. This is probably attributed to the large flux of traffic in cloud-native settings, where lawful spurts of traffic can be mistakenly taken as probing. To solve this, it will introduce anomaly detection methods in future work that aim at enhancing the traffic system to differentiate benign traffic and suspicious traffic at times when there is a lot of traffic.

VI. Discussion

The hybrid DL-based IDS proposed has a few strengths. It is efficient in more than one way: firstly, it is highly accurate as it successfully combines CNNs with applying spatial feature information and Bi-LSTM networks with time-related dependencies information. This bi-directional solution enables great detection and performance over dynamic cloud-native traffic. Moreover, the system has low latency because the model is deployed as a microservice through Kubernetes, making sure that threat detection is real-time. Kubernetes is also utilized to make the system scalable; that is, it can handle dynamic traffic with ease.

The model is, however, limited. It is largely based on flow and metadata but does not inspect its payload, which can be a limitation to its capability to identify application-layer attacks. In addition, retraining of the model periodically may be needed to meet the changes in threats in the dynamic cloud-native environment, which might impact long-term performance.

Future directions involve utilizing federated learning to train models in a decentralized manner with privacy in mind, over cloud computing platforms. The other aspect to improve is encrypted traffic - methods to analyze encrypted traffic should be developed without affecting the accuracy of detection. Secondly, it may be beneficial to incorporate anomaly detection within the existing hybrid model and make it more efficient at detecting new attacks or zero-day vulnerabilities, which, in turn, would increase the resilience of the IDS.

VII. Conclusion

It is in this paper that CNNs are used in the extraction of spatial features and Bi-LSTM networks are used in the sequencing modeling of temporal features, to create a hybrid DL architecture to support real-time intrusion detection in the cloud-native setting. The distinct issues of cloud-native systems, including the dynamic nature of traffic, their large traffic volume, and low-latency detection needs, are answered by the proposed model.

By using the hybrid model, rigorously tested on UNSW-NB15 and CSE-CIC-IDS2018 data, the hybrid model performs better than the traditional ML models, with an accuracy of 98 percent, a precision of 0.97, and a recall of 0.96, but with a latency of less than 50 ms. These findings provide evidence that the hybrid framework may be successfully employed to manage both familiar and unknown attacks, providing a powerful approach to intrusion detection in cloud-native systems, which is not only real-time and high-accuracy.

Kubernetes, used to deploy the model, makes it scalable and efficient and allows it to cope with the varying traffic requirements of a cloud-native environment. Nevertheless, present developments like federated learning, better encrypted traffic processing, and connection to anomaly-based methods of detection will further streamline the system to detect previously unknown attacks and maintain future scalability in cloud systems as they evolve.

The suggested hybrid DL framework would be effective overall, offering a promising and scalable solution to real-time intrusion detection and existing as part of the continuum of smart security systems in modern IT environments.

Data Availability: The datasets used in this study are publicly available and can be accessed from the respective links for UNSW-NB15 (https://research.unsw.edu.au/projects/unsw-nb15-dataset) and CSE-CIC-IDS2018 (https://www.kaggle.com/datasets/solarmainframe/ids-intrusion-csv).

References

[1] B. Nascimento, R. Santos, J. Henriques, M. V. Bernardo, and F. Caldeira, “Availability, Scalability, and Security in the Migration

from Container-Based to Cloud-Native Applications," *Computers*, vol. 13, no. 8, p. 192, Aug. 2024, doi: 10.3390/computers13080192.

[2] H. Park, A. EL Azzaoui, and J. H. Park, "AIDS-Based Cyber Threat Detection Framework for Secure Cloud-Native Microservices," *Electronics*, vol. 14, no. 2, p. 229, Jan. 2025, doi: 10.3390/electronics14020229.

[3] V. Z. Mohale and I. C. Obagbuwa, "Evaluating machine learning-based intrusion detection systems with explainable AI: enhancing transparency and interpretability," *Front. Comput. Sci.*, vol. 7, May 2025, doi: 10.3389/fcomp.2025.1520741.

[4] S. Sadhwani, M. A. H. Khan, R. Muthalagu, P. M. Pawar, and K. Suresh, "A hybrid BiLSTM-CNN approach for intrusion detection for IoT applications," *Sci Rep*, vol. 16, no. 1, p. 155, Dec. 2025, doi: 10.1038/s41598-025-29079-y.

[5] K. P. Sharma *et al.*, "Interpretable intrusion detection for IoT environments using a self-attention-based explainable AI framework," *Sci Rep*, vol. 15, p. 39937, Nov. 2025, doi: 10.1038/s41598-025-23750-0.

[6] U. Ahmed *et al.*, "Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering," *Sci Rep*, vol. 15, p. 1726, Jan. 2025, doi: 10.1038/s41598-025-85866-7.

[7] S. Kim, C. Hwang, and T. Lee, "Anomaly Based Unknown Intrusion Detection in Endpoint Environments," *Electronics*, vol. 9, no. 6, p. 1022, Jun. 2020, doi: 10.3390/electronics9061022.

[8] G. Dkmak, B. Can, O. Sevinc, C. B. Egeli, F. Baday, and B. Cetintav, "AI-Driven Anomaly Detection in Cloud-Native Microservices: The Night's Watch Algorithm," *Applied Sciences*, vol. 15, no. 23, p. 12762, Jan. 2025, doi: 10.3390/app152312762.

[9] F. Genuario, G. Santoro, M. Giliberti, S. Bello, E. Zazzera, and D. Impedovo, "Machine Learning-Based Methodologies for Cyber-Attacks and Network Traffic Monitoring: A Review and Insights," *Information*, vol. 15, no. 11, p. 741, Nov. 2024, doi: 10.3390/info15110741.

[10] A. Humza, F. Zainab, F. Raoof, and A. Zahoor, "A Enhanced Intrusion Detection Using CNN-LSTM with Advanced Evaluation Metrics: Hybrid Approach By Deep Learning for Prediction of Network Intrusion," *KIET Journal of Computing and Information Sciences*, vol. 8, no. 1, Jul. 2025, doi: 10.51153/kjcis.v8i1.240.

[11] W. Gamaleldin, O. Attayyib, M. M. Alnfiai, F. A. Alotaibi, and R. Ming, "A hybrid model based on CNN-LSTM for assessing the risk of increasing claims in insurance companies," *PeerJ Comput Sci*, vol. 11, p. e2830, Apr. 2025, doi: 10.7717/peerj-cs.2830.

[12] P. P. Ray, "A Review of TRiSM Frameworks in Artificial Intelligence Systems: Fundamentals, Taxonomy, Use Cases, Key Challenges and Future Directions," *Expert Systems*, vol. 43, no. 3, p. e70213, 2026, doi: 10.1111/exsy.70213.

[13] S. U. Yang, "Research on network behavior anomaly analysis based on bidirectional LSTM," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, IEEE, 2019, pp. 798–802. Accessed: Apr. 25, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8729475/

[14] M. Azizjon, A. Jumabek, and W. Kim, "1D CNN based network intrusion detection with normalization on imbalanced data," in *2020 international conference on artificial intelligence in information and communication (ICAIIC)*, IEEE, 2020, pp. 218–224. Accessed: Apr. 25, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9064976/

[15] J. Sinha and M. Manollas, "Efficient Deep CNN-BiLSTM Model for Network Intrusion Detection," in *Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition*, Xiamen China: ACM, Jun. 2020, pp. 223–231. doi: 10.1145/3430199.3430224.

[16] J. Zhang, Y. Ling, X. Fu, X. Yang, G. Xiong, and R. Zhang, "Model of the intrusion detection system based on the integration of spatial-temporal features," *Computers & Security*, vol. 89, p. 101681, 2020.

[17] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Information Sciences*, vol. 513, pp. 386–396, 2020.

[18] N. Guizani and A. Ghafoor, "A network function virtualization system for detecting malware in large IoT based networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1218–1228, 2020.

[19] A. Aleesa, M. Younis, A. A. Mohammed, and N. Sahar, "Deep-intrusion detection system with enhanced UNSW-NB15 dataset based on deep learning techniques," *Journal of Engineering Science and Technology*, vol. 16, no. 1, pp. 711–727, 2021.

[20] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: hybrid deep neural network for network intrusion detection system," *IEEE access*, vol. 10, pp. 99837–99849, 2022.