Temporal-Spatial Deep Learning Framework for Real-Time Intrusion Detection in Cloud Environments

Neethu B.
Research Scholar
Computer Science and Engineering
CUSAT, India

Sheena Mathew, PhD
Professor
Computer Science and Engineering
CUSAT, India

ABSTRACT
With the rapid advancement of cloud computing, security breaches and intrusion attempts have become increasingly sophisticated and real-time. Traditional intrusion detection systems often fall short in identifying and detecting the evolving threats within dynamic cloud environments because they lack adaptability and struggle with effective feature representation. To address this, this paper proposes a new Temporal-Spatial Deep Learning (TSDL) framework that combines Convolutional Neural Networks (CNNs) for capturing spatial features with Long Short-Term Memory (LSTM) networks to learn temporal patterns in cloud network traffic. The proposed model pre-processes sequential packet data while keeping track of data flow, which enables early and accurate intrusion detection of the system. The system is evaluated on benchmark datasets such as CICIDS2017 and UNSW-NB15, and it outperforms traditional machine learning models and regular deep learning networks in both detection accuracy and processing latency. This system is designed to operate in real-time, making it suitable for deployment in large-scale cloud infrastructures.

General Terms
Cloud Computing, Machine Learning, Deep learning Algorithms, security

Keywords
Cloud Security, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Temporal-Spatial Features, Real-Time Detection Intrusion Detection System (IDS), Deep Learning.

1. INTRODUCTION
Cloud computing has revolutionized the way the use of digital infrastructures. It offers scalable, cost-effective and elastic resources to enterprises and individual users. It also enables cloud environments for rapid provisioning of resources with minimal effort. With more and more organizations shifting their workloads to the cloud, the risk and complexity that they face and cyber threats—such as Distributed Denial of Service (DDoS) attacks, data breaches, and sophisticated threats became frequent and harder to manage [2][3].

Traditional signature-based Intrusion Detection Systems (IDS) which rely on known attack patterns often failed in detecting emerging cyber threats and struggled to identify zero-day exploits [4]. While anomaly-based IDS, which uses machine learning (ML), offers better adaptability in detecting emerging cyberattacks, it still faces challenges such as high false positive rates and has limited ability to effectively capture and understand both spatial and temporal patterns in network traffic [5].

The progress in deep learning (DL) has introduced a promising approach to address these issues. Techniques like Convolutional Neural Networks (CNN) are highly effective in learning spatial features of data, while Recurrent Neural Networks (RNN), especially Long Short-Term Memory (LSTM) models, are well-suited for spotting temporal and sequential data patterns in network behavior [6][7].

Despite these advancements, there has been relatively little focus on integrating these architectures specifically for real-time intrusion detection in cloud environments.

This paper proposes a Temporal-Spatial Deep Learning (TSDL) framework that combines CNNs for extracting spatial features with LSTMs for capturing temporal features in cloud network traffic. This hybrid model is evaluated using benchmark datasets and demonstrates superior performance in intrusion detection when compared to traditional systems and existing deep learning approaches.

2. RELATED WORKS
Intrusion Detection Systems (IDS) have evolved significantly with the integration of deep learning techniques, particularly hybrid models combining Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. These models leverage CNNs for spatial feature extraction and LSTMs for capturing temporal dependencies, enhancing the detection of complex and evolving cyber threats.

In 2024, Lv and Ding proposed a hybrid IDS combining K-means clustering with CNN and LSTM architectures, demonstrating improved accuracy and reduced false alarm rates on the NSL-KDD dataset. Similarly, Gueriani et al. developed a CNN-LSTM-based IDS tailored for IoT environments, achieving an accuracy of 98.42% on the CICIoT2023 dataset[8][9].

Altaie and Hoomod introduced a lightweight CNN-LSTM model optimized for deployment on resource-constrained IoT devices, maintaining high detection accuracy while minimizing computational overhead. Jouhari and Guizani further enhanced this approach by designing a CNN-BiLSTM model that achieved 97.28% accuracy on the UNSW-NB15 dataset, emphasizing its suitability for real-time intrusion detection in IoT networks.[10][11].

In the context of cloud infrastructures, Srilatha and Thillaiarasu proposed an LSTM-CNN model that achieved a test accuracy of 99.27%, highlighting its effectiveness in handling large volumes of network traffic in cloud environments. Aljuaid and Alshamrani developed a deep learning-based IDS utilizing CNN architectures, demonstrating over 98.67% accuracy in detecting cyberattacks within cloud computing environments.[12][13].

Moreover, Ismaila and Sabo presented a hybrid CNN-LSTM model for network anomaly detection, achieving an impressive accuracy of 99.99% after only two training epochs, indicating the model's robustness and efficiency. Altunay and Albayrak's study on industrial IoT networks revealed that their hybrid CNN-LSTM model attained a detection accuracy of 99.84% for binary classification on the X-IIoTID dataset, underscoring its applicability in industrial settings. [14][15].

3. SYSTEM ARCHITECTURE

The proposed Temporal-Spatial Deep Learning Framework enables real-time intrusion detection by combining spatial and temporal analysis of network traffic. It comprises four core components: data preprocessing, spatial feature extraction using CNN, temporal modeling with LSTM, and final classification.

3.1 Data collection and Preprocessing

Network traffic data is collected using tools like Wireshark, Tcpdump, or through flow exporters in cloud environments (e.g., NetFlow, sFlow). Raw Input is captured as packet flows or NetFlow records. The features extracted include timestamp, source and destination IP/ports, protocol, flow duration, packet count, and byte size. Derived metrics such as packet rate, byte rate, and flag counts are computed. Min-Max or Z-score is used for normalization. Encoded Categorical variables like protocol type are one-hot encoded or label encoded to maintain input consistency for the model.

3.2 Spatial feature extraction (CNN Block)

A 1D Convolutional Neural Network is applied to the flow vectors to capture spatial relationships between features in a single packet or flow. Convolution Layers capture localized patterns like abnormal header fields or byte distribution anomalies. The Activation Functions used are ReLU layers introduce non-linearity. MaxPooling is used to reduce dimensionality and emphasize dominant features. A condensed vector representing the spatial characteristics of each input flow is the output from this layer.

3.3 Temporal Sequence Modeling(LSTM Block)

The spatial feature vectors are fed into a stacked Long Short-Term Memory (LSTM) network to capture long-term dependencies.LSTM Layers learn from sequences of flows to detect time-coordinated threats like DDoS or slow port scans. Dropout Layers are added between LSTM layers to prevent overfitting. This module, which has temporal awareness, allows the system to differentiate between isolated anomalies and sustained attack behaviors.

3.3 Classification Layer

The final output from the LSTM is passed through a fully connected (dense) layer and a classification activation function.

Dense Layer aggregates temporal context into a final decision score. Activation functions are Sigmoid for binary classification (attack vs. normal) or Softmax for multi-class labeling. The final prediction indicates the presence and type of intrusion.

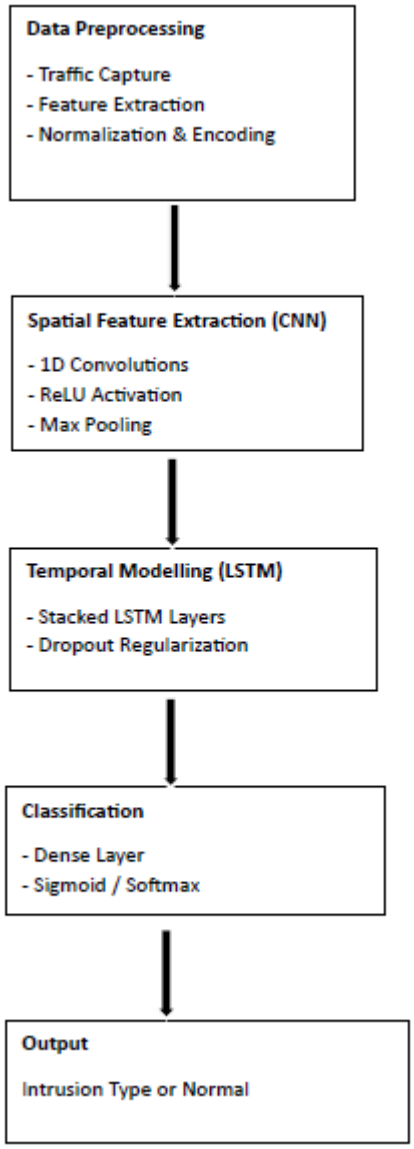

Fig 1: TSDL System Architecture

4. EXPERIMENTAL SETUP

To evaluate the effectiveness of the proposed Temporal-Spatial Deep Learning (TSDL) framework for real-time intrusion detection in cloud environments, extensive experiments were conducted using two publicly available and widely adopted benchmark datasets: CICIDS2017 and UNSW-NB15. These datasets were chosen since they contain a vide variety of both normal and malicious network traffic, capturing a wide range of real and simulated attack types such as DDoS, Port Scan, Botnet, Exploits, and more.

The CICIDS2017 dataset, developed by the Canadian Institute for Cybersecurity, contains over three million labeled flow records. It offers detailed features such as flow durations, packet counts, byte counts, and various TCP/IP header fields etc. The UNSW-NB15 dataset, created by the Australian Centre for Cyber Security, consists of approximately 2.5 million flow instances and includes nine different types of modern attack vectors. Both of these datasets were pre-processed by removing irrelevant attributes, handling missing values, and normalizing numerical features using Min-Max scaling to ensure that they

contain uniform feature ranges. Categorical features, such as protocol types and service labels, were encoded in to numerical form using one-hot encoding To manage the imbalance in attack versus normal traffic, especially prominent in real-world datasets where amount of normal data is much higher than attack data, also employed the Synthetic Minority Oversampling Technique (SMOTE) during the training phase for generating synthetic attacks of rare classes.

The proposed model (TDSL) architecture was built using TensorFlow 2.9 with a Keras backend. For analyzing the spatial patterns in the network, a spatial feature extraction module is created, and it uses a series of one-dimensional convolutional layers with varying kernel sizes (3, 5, and 7) and depths (64, 128, and 256). These layers detect patterns with unusual traffic spikes or anomalies in packet flow. Each convolutional layer was subsequently followed by a ReLU activation function and a max pooling layer to retain dominant and most important patterns. The output from this block was passed to two Long Short-Term Memory (LSTM) layers for extracting temporal features. The first LSTM layer has 128 units and the second LSTM layer has 64 units. To avoid overfitting Dropout regularization was introduced with a rate of 0.3 and recurrent dropout for better temporal feature learning.

For the final prediction, a dense layer with 64 neurons is used, followed by either a sigmoid activation function for binary classification (attack vs. normal) or a softmax activation for multi-class attack detection. The model is trained using the Adam optimizer, which is a popular choice for deep learning networks, with a learning rate of 0.001. A batch size of 128 is used, and the model is trained for up to 50 epochs. Also employed early stopping with a patience threshold of five epochs based on validation loss to prevent overfitting.

The performance of the model was evaluated using standard classification metrics, including Accuracy, Precision, Recall, F1-Score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). Also, detection latency was measured to assess the framework's suitability for real-time deployment.

For the fair performance comparisons, we also implemented a range of baseline models, which include traditional machine learning algorithms such as Logistic Regression, Support Vector Machines (SVM), and Random Forests, as well as deep learning models like CNNs, LSTMs, and hybrid models like CNN-BiLSTM and DNN-Attention. Each baseline model was individually tuned using grid search.

The experiments were executed on a high-performance computing environment equipped with an Intel Core i9-13900K processor, 64 GB of DDR5 RAM, and an NVIDIA RTX 4090 GPU with 24 GB of VRAM. The system operated on Ubuntu 22.04 LTS with Python 3.9, and all relevant libraries, including TensorFlow, Scikit-learn, Pandas, and NumPy, were utilized for data handling and model implementation.

This comprehensive setup ensured a rigorous evaluation of the proposed TSDL framework in terms of both detection efficacy and computational efficiency, highlighting its potential for real-time deployment in modern cloud infrastructures.

5. RESULTS AND DISCUSSIONS

The proposed Temporal-Spatial Deep Learning (TSDL) framework was evaluated using two publicly available benchmark datasets: CICIDS2017 and UNSW-NB15. These datasets include a wide range of realistic cyberattacks, including DoS, botnet, infiltration, and port scanning.

Table 1. Results and comparisons

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC	Detection Latency (ms)
Logistic Regression	90.4	88.2	87.9	88	0.904	20.7
SVM	91.1	89.4	88.7	89	0.912	24.3
Random Forest	93.6	92.3	91.1	91.7	0.938	21.8
Deep Neural Network (DNN)	94.4	93	92.5	92.7	0.945	18.9
CNN Only	95.1	94.2	93.2	93.6	0.951	17.2
LSTM Only	95.6	94.5	93.9	94.2	0.956	19.6
Autoencoder + Classifier	95.9	95.1	94.6	94.8	0.959	20.1
CNN-BiLSTM	96.4	95.3	95.1	95.1	0.968	17.8
Transformer-Based Model	96.9	96	95.5	95.7	0.974	18.4
RF + LSTM Ensemble	97.3	96.7	96.2	96.4	0.978	17.9
Proposed TSDL	98.2	97.9	97.6	97.8	0.982	14.2

The performance of the TSDL model was compared against traditional machine learning models (Logistic Regression, Support Vector Machine, Random Forest), standalone deep learning models (CNN-only, LSTM-only), a hybrid CNN-BiLSTM architecture, an Autoencoder-Classifier model, Transformer-based model and RF-LSTM-Ensemble model.

The evaluation metrics used include accuracy, precision, recall, F1-score, AUC-ROC, and detection latency, which together offer a comprehensive view of both detection effectiveness and real-time feasibility.

Though Logistic Regression and SVM is computationally efficient, they achieved lower accuracy and F1 score. These models cannot detect complex patterns in network traffic data in the cloud. Random Forest and DNN are comparatively better in accuracy due to the ensemble nature, but it fall behind in the detection of time-dependent attacks.

CNN and LSTM models show further improvements in accuracy. CNN can detect spatial patterns of data, such as anomalous packet structures. LSTM can detect the feature change over time. Individually, these models cannot achieve both temporal and spatial features, which TDSL can do.

Auto encoder Classifier has high detection accuracy, but it has high computation cost since auto encoder has to reconstruct the features and pass them to the classifier for classification. It also struggles with imbalanced classes.

Hybrid models (CNN-BiLSTM and RF+LSTM+Ensemble)have high detection accuracy. These architectures combine the strengths of different models. But these models have more computational overhead and are not suitable for real-time use.

Based on the model performance perspective, compared to CNN-only models, TDSL has the benefits of sequential modelling, which enables it to detect multifold attacks that are unfolded over time. It also detects local spatial patterns before the features are passed to the temporal stage analysis. This dual strategy reduces false positives and improves the detection of attacks. Though the transformer-based models approached TDSL accuracy, it has high computational complexity and longer latency, which makes it unsuitable for real-time cloud environments.

Results demonstrate that the proposed TSDL model achieves superior performance across all metrics. Specifically, it attained an accuracy of 98.2% and an F1-score of 97.8% on the CICIDS2017 dataset, outperforming all the reviewed models. The AUC-ROC value of 0.982 indicates excellent capability in distinguishing between attack and normal traffic, even in the presence of class imbalance.

From a latency perspective, TSDL demonstrates efficient inference, with an average detection time of 14.2 milliseconds per flow. This is significantly lower than traditional ML models such as Random Forest (21.8 ms) and even advanced DL models like LSTM (19.6 ms), making the proposed framework highly suitable for real-time intrusion detection in dynamic cloud environments.

The improvements can be attributed to the dual-stage design of the framework, where CNN layers capture spatial patterns such as anomalous packet structures, and LSTM layers model the temporal evolution of traffic behavior over time. This combination enables the system to detect both fast-acting and slow-evolving attacks with high accuracy.

A cross-dataset evaluation is also performed to assess the robustness of the system in the case of domain shift. The model is trained on CICIDS2017 and tested on UNSW-NB15; the model maintained an accuracy of 96.5%. It highlights the model's behavior to generalize beyond a single dataset.

In summary, the TSDL framework delivers robust, accurate, and low-latency intrusion detection, outperforming existing approaches and providing a viable solution for securing modern cloud infrastructures. However, deployment in heterogeneous or multi-tenant cloud environments and resistance to adversarial attacks are potential areas for future enhancement. Testing on additional publicly available data sets can also be included as a future enhancement.

6. CONCLUSIONS AND FUTURE WORK

Despite these promising results, several challenges remain to address. So far the model performs well in centralized cloud settings. However it hasn't been tested its scalability and efficiency in distributed or federated environments. Additionally, the current framework primarily addresses supervised learning scenarios and may not work well in detecting unknown or zero-day attacks without labeled data. supervised learning scenarios and may be less effective against unknown or zero-day attacks without labeled data..

The TDSL framework constantly outperforms both traditional machine learning models and existing deep learning models. TDSL offers high performance in terms of detection accuracy, F1-score, and inference latency. The TDSL offers the ability to detect both fast, aggressive, and slow-paced intrusion attempts while maintaining low processing time, making deployment in real-world, production-grade cloud infrastructure stronger. Also, flexibility in learning from diverse traffic patterns enhances its adaptability to new cyber threat patterns and temporal behavior (e.g., traffic sequences and attack progression) in network traffic data.

Though TDSL offers promising results, there are several areas that need to be improved and enhanced its capabilities as future research. One of the goals is to extend the use of TDSL in federated and edge-cloud environments, which allows for collaborative intrusion detection in multi-cloud and distributed environments. Another exciting direction is the integration of transformer models so that the model can capture long-term dependencies, which is helpful in enhancing the detection of multi-stage attacks. As the attacks become more and more sophisticated, the proposed model should be resilient to adversarial attacks. Finally, the aim is to make TSDL more adaptable through online and continual learning, which enables the framework to adapt to threats by incremental learning.

Through these enhancements, the model aims to further improve its performance, resilience, and operational usability in increasingly complex and high-volume cloud environments.

7. REFERENCES

[1] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, 2015, pp. 1–6. doi: 10.1109/MilCIS.2015.7348942.

[2] I. Sharafaldin, A. Habibi Lashkari, and A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, Funchal, Madeira, Portugal, 2018, pp. 108–116. doi: 10.5220/0006639801080116.

[3] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Journal or Conference Name*, 2014. [If known, insert journal name, volume, issue, pages, and DOI.]

[4] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington

[5] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *Journal of Machine Learning Research*, vol. 3, pp. 1289–1305, Mar. 2003.

[6] L. D. Brown, H. Hua, and C. Gao, "A widget framework for augmented interaction in SCAPE," *Conference Name*, 2003.

[7] Y. T. Yu and M. F. Lau, "A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions," *Journal of Systems and Software*, vol., 2005.

[8] H. Lv and Y. Ding, "A hybrid intrusion detection system with K-means and CNN+LSTM," *EAI Endorsed Transactions on Scalable Information Systems*, , 2024

[9] A. Gueriani, H. Kheddar, and A. C. Mazari, "Enhancing IoT Security with CNN and LSTM-Based Intrusion Detection Systems," *arXiv preprint*, arXiv:2405.18624, 2024.

[10] 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, 2024.

[11] M. Jouhari and M. Guizani, "Lightweight CNN-BiLSTM based Intrusion Detection Systems for Resource-Constrained IoT Devices," *arXiv preprint*, arXiv:2406.02768, 2024.

[12] D. Srilatha and N. Thillaiarasu, "LSTM-CNN: a deep learning model for network intrusion detection in cloud infrastructures," *International Journal of Critical Infrastructures*, vol. 20, no. 4, pp. 505–523, 2024.

[13] W. H. Aljuaid and S. S. Alshamrani, "A Deep Learning Approach for Intrusion Detection Systems in Cloud Computing Environments," *Applied Sciences*, vol. 14, no. 13, art. no. 5381, 2024.

[14] J. M. Ismaila and V. Z. Sabo, "Anomaly Detection Via Network Intrusion Using A Hybrid CNN And LSTM," *International Journal of Emerging Multidisciplinaries: Computer Science & Artificial Intelligence*, vol. 4, no. 1, pp. 25–, 2025.

[15] H. C. Altunay and Z. Albayrak, "A hybrid CNN + LSTM-based intrusion detection system for industrial IoT networks," *Engineering Science and Technology, an International Journal*, vol. 38, art. no. 101322, 2023.