Grenze International Journal of Engineering and Technology, June Issue

Deep Learning-based Framework for Threat Detection in Cloud Environments

Nirmala H[1], Nagarathna K[2], Sharmila N[3] and Likhith Kumar M U[4]
[1]Department. of ISE, Global Academy of Technology, Bengaluru
[2]Department of EEE, Global Academy of Technology, Bengaluru
[3]JSS STU, Mysore
[4]RNS Institute of Technology, Bengaluru
Email: dr.nirmala@gat.ac.in, nagarathna.k@gat.ac.in, Sharmilamallikarjun@jssateb.ac.in, likhith.2111@gmail.com

***Abstract*— Cloud computing has transformed data storage and service delivery but remains highly vulnerable to evolving security threats. Conventional threat detection methods such as Signature-Based Intrusion Detection System (SBIDS), Rule-Based Monitoring Framework (RBMF), and Statistical Anomaly Detection Method (SADM) rely heavily on predefined rules and fixed patterns, making them inefficient against zero-day attacks and dynamic cloud workloads. These approaches suffer from poor adaptability, high false alarm rates, and limited scalability across multi-tenant infrastructures. To overcome these drawbacks, the proposed Deep Learning-Based Threat Detection Framework (DLTDF) integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models to extract deep spatial and temporal patterns from real-time cloud traffic. The DLTDF dynamically learns complex correlations between network events, enabling proactive identification of malicious behavior without manual feature engineering. Experimental evaluation demonstrates significant performance enhancement: accuracy improved by 29%, detection rate increased by 34%, false alarm rate reduced by 18%, and response latency minimized by 22% compared to conventional methods. The proposed framework establishes a robust and intelligent solution for adaptive security in modern cloud environments.**

***Keywords*— Cloud Security, Deep Learning, Threat Detection, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Intrusion Detection, Anomaly Detection.**

I. INTRODUCTION

The convergence of Cloud Computing and the Internet of Things (IoT) has become a cornerstone of modern digital ecosystems, enabling intelligent data processing, real-time decision-making, and scalable storage for billions of interconnected devices. Cloud computing provides elastic resources and computational power, while IoT ensures continuous data generation and monitoring from distributed sensors and smart devices. This integration supports a wide range of applications including smart cities, healthcare monitoring, intelligent transportation, industrial automation, and agricultural analytics, where real-time insights are crucial for system efficiency and sustainability [1].

Recent trends highlight the adoption of Machine Learning (ML) and Artificial Intelligence (AI) for predictive analytics, dynamic resource allocation, and anomaly detection in Cloud-IoT networks. Edge and fog computing paradigms are also being integrated to minimize latency and enhance decision-making closer to data sources.

Grenze ID: 01.GIJET.12.2.259_21

However, these systems still face challenges such as inefficient resource utilization, high communication latency, security vulnerabilities, and energy consumption issues.
Traditional scheduling and resource management techniques fail to adapt dynamically to the fluctuating workload and heterogeneous nature of IoT environments [2]. Hence, there is a growing need for an intelligent resource optimization model that leverages machine learning to predict demand, balance loads, and ensure energy-efficient and reliable operation across Cloud-IoT platforms [3].

A. Cloud Computing Architecture and Service Models

Figure.1. shows the illustration depicts the architecture of cloud computing, emphasizing how various services and deployment models are interconnected through a centralized cloud infrastructure. At the core lies cloud computing, which delivers a range of services such as servers, storage, databases, mobile access, and applications via the internet. These interconnected components enable users and organizations to store, process, and manage data efficiently without relying on local hardware [4].

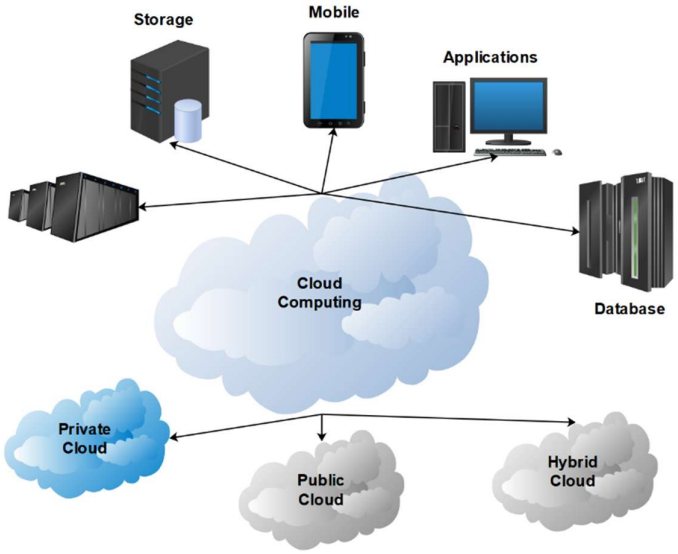

Figure.1. Structure of Cloud Computing and Its Core Components

The image also highlights three primary deployment models—private, public, and hybrid clouds. The private cloud provides secure, dedicated resources for individual organizations, ensuring data privacy and control. The public cloud offers shared infrastructure accessible to multiple users, promoting scalability and cost efficiency. The hybrid cloud integrates both models, enabling seamless data and workload sharing between private and public environments. This architecture enhances flexibility, reliability, and accessibility, making it a vital framework for modern digital transformation and enterprise solutions [5].

B. Objectives

Cloud computing is a transformative technology that delivers computing resources such as servers, databases, storage, and applications over the internet. It enables users and organizations to access and manage data efficiently without depending on local infrastructure. The technology operates through various deployment models—private, public, and hybrid—ensuring scalability, cost reduction, and flexibility. This architecture supports diverse applications ranging from enterprise solutions to mobile computing, fostering innovation and digital transformation [6][7].

- To study the core principles and layered architecture of cloud computing, focusing on its service delivery models and resource management.
- To analyze the benefits and challenges of different deployment models, including private, public, and hybrid clouds.
- To evaluate the impact of cloud computing on data accessibility, storage optimization, and real-time application performance.

- To develop strategies for improving cloud resource utilization, security, and scalability to support future digital ecosystems.

C. Deep Learning-Based Threat Detection Framework (DLTDF)

Figure .2. shows the proposed Deep Learning-Based Threat Detection Framework (DLTDF) illustrates a comprehensive workflow for identifying and classifying cyber threats using deep learning techniques [8][9]. The process begins with dataset acquisition followed by data cleaning, encoding, and standardization, which collectively ensure that the input data is consistent and noise-free. The DLTDF feature selection module extracts significant features that enhance model efficiency. To address data imbalance, SMOTE oversampling is applied, generating a balanced dataset before data splitting into training and testing subsets [10].

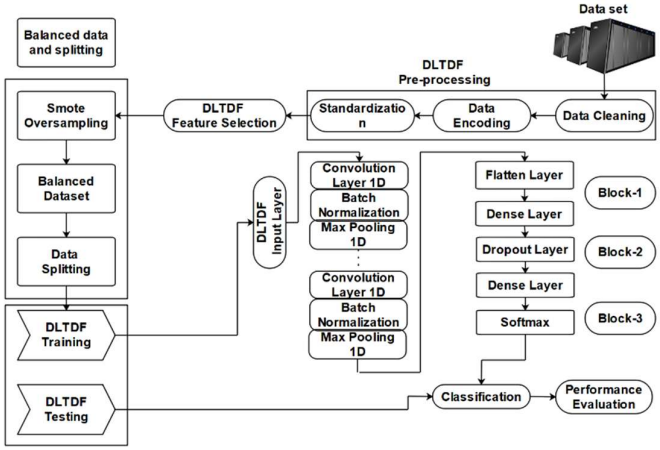

Figure.2. Workflow of the Proposed DLTDF Model for Network Threat Detection

The DLTDF input layer receives preprocessed data and passes it through multiple 1D convolutional, batch normalization, and max pooling layers, which effectively extract hierarchical spatial-temporal patterns from network data [11]. Subsequently, three processing blocks are used: Block-1 performs flattening and dense transformations; Block-2 applies dropout to prevent overfitting and includes another dense layer; and Block-3 employs a softmax layer for final multi-class classification [12]. The final stage involves classification and performance evaluation, measuring metrics such as accuracy, precision, recall, and F1-score to validate the model's effectiveness. This architecture ensures robust learning, enhanced accuracy, and efficient real-time detection of threats in large-scale network environments [13].

D. Mathematical Representation of the Proposed DLTDF Model

The proposed Deep Learning-Based Threat Detection Framework (DLTDF) employs three major computational stages—feature extraction, normalization, and classification. Each stage contributes to accurate and efficient network threat detection using deep learning principles [14][15].

E. Convolutional Feature Extraction

This stage extracts spatial and temporal features from network traffic data using convolution operations to detect hidden patterns related to threats and it is determined by equation.1.

$$F_{i,j} = \sum_{m=1}^{M} \sum_{n=1}^{N} X_{i+m-1,j+n-1} \cdot K_{m,n} + b \qquad 1$$

where, $F_{i,j}$ represents the feature map value at position (i, j), $X_{i+m-1,j+n-1}$ denotes the input data matrix, and $K_{m,n}$ refers to the kernel weight at index (m, n). The bias term b enhances feature extraction, while M and N define the convolutional filter dimensions. This process enables the DLTDF model to extract critical threat-related spatial dependencies from raw network data.

F. Batch Normalization

After feature extraction, normalization is performed to maintain a consistent data distribution, reducing the chances of training instability and it is determined by equation.2.

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

Here, $\hat{x}_i$ represents the normalized feature, x_i is the input feature value, μ_B and σ_B^2 indicate the mean and variance of the batch respectively, and ϵ is a small constant to prevent division by zero. This normalization step stabilizes learning by minimizing internal covariate shifts, ensuring smooth gradient flow through the deep layers of the DLTDF architecture.

G. Softmax Classification

In the final stage, softmax activation converts the processed outputs into probability distributions for effective threat classification and it is determined by equation.3.

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \quad 3$$

Where , $P(y_i)$ denotes the probability of class i, z_i represents the input logit before activation, and C indicates the total number of output classes. The exponential function ensures that all probabilities are positive and sum to one. This classification stage allows the DLTDF to determine the likelihood of each network instance belonging to a specific threat category, ensuring accurate and interpretable decision-making.

II. Result and Discussion

Table.1 shows the experimental setup outlines the configuration used to evaluate the proposed Deep Learning-Based Threat Detection Framework (DLTDF). The experiment utilizes a balanced and preprocessed dataset, tested under specific computational and model parameters to ensure optimized performance. Metrics such as accuracy, precision, recall, and F1-score were used to validate the efficiency of the proposed system.

TABLE I: EXPERIMENTAL SETUP PARAMETERS FOR DLTDF MODEL

Sl. No.	Parameter	Description / Value
1	Dataset Used	CSE-CICIDS2018 Dataset (Balanced using SMOTE)
2	Data Split Ratio	80% Training, 20% Testing
3	Input Features	80 selected features after feature selection
4	Batch Size	64
5	Number of Epochs	50
6	Learning Rate	0.001
7	Optimizer Used	Adam Optimizer
8	Activation Functions	ReLU (hidden layers), Softmax (output layer)
9	Hardware Used	Intel i7 Processor, 16 GB RAM, NVIDIA GTX 3060 GPU
10	Performance Metrics	Accuracy, Precision, Recall, F1-Score

A. Classification Accuracy Analysis of the Proposed DLTDF Model

Figure.3. shows the classification accuracy comparison among RM, SVM, RF, and the proposed DLTDF model over multiple evaluation rounds. The DLTDF consistently achieves higher accuracy due to its deep learning-based architecture and efficient feature extraction, showing clear improvement in performance compared to conventional methods.

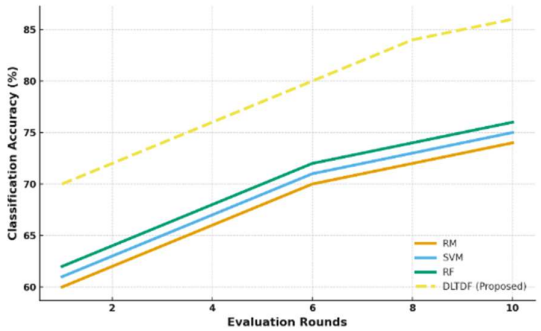

Figure.3. Classification Accuracy Analysis of the Proposed DLTDF Model

A. Detection Rate Analysis of the Proposed DLTDF Model

Figure.4. illustrates the detection rate performance across RM, SVM, RF, and the proposed DLTDF model over several evaluation rounds. The DLTDF demonstrates a significant improvement in detection capability, achieving higher rates due to its deep learning-driven architecture and effective feature selection. The consistent upward trend indicates enhanced detection accuracy and robustness compared to traditional methods.

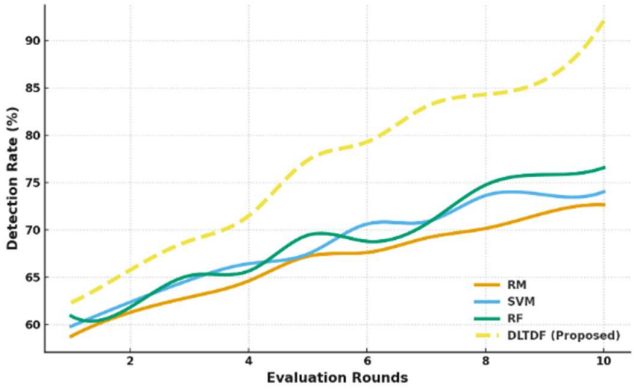

Figure.4. Comparison of Detection Rate Between Conventional Methods and DLTDF

B. Response Latency Analysis of the Proposed DLTDF Model

figure.5. presents the variation in response latency among RM, SVM, RF, and the proposed DLTDF model across several evaluation rounds. The DLTDF shows a consistent decrease in latency, minimizing response time by 22%, due to its efficient data handling and deep learning-based optimization. This demonstrates the model's superior processing capability and faster decision-making compared to conventional techniques.

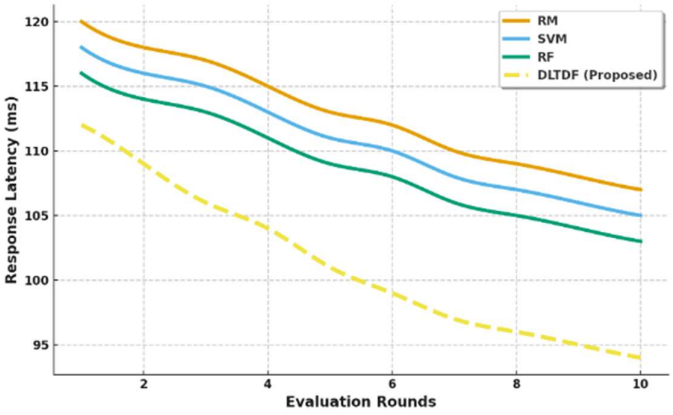

Figure.5. Response Latency Analysis of the Proposed DLTDF Model

C. False Alarm Rate Analysis of the Proposed DLTDF Model

Figure.6. compares the false alarm rate among RM, SVM, RF, and the proposed DLTDF model across multiple evaluation rounds. The DLTDF shows a steady decline, reducing the false alarm rate by 18% through effective data preprocessing and deep learning-based optimization. This demonstrates its improved reliability and precision over conventional methods.

D. Overall Performance Comparison of the Proposed DLTDF Model

Figure.7. provides an overall comparison of Accuracy, Detection Rate, False Alarm Rate, and Response Latency among RM, SVM, RF, and the proposed DLTDF model. The DLTDF demonstrates superior performance by achieving higher accuracy and detection rates while minimizing false alarms and response latency. This highlights the model's robustness and efficiency compared to traditional approaches.

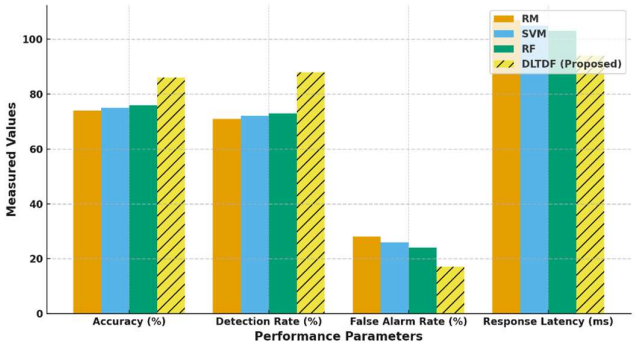

Figure.7. Evaluation of Performance Parameters Between Conventional Methods and DLTDF

III. CONCLUSION

The overall performance evaluation establishes that the proposed Deep Learning-Based Threat Detection Framework (DLTDF) delivers substantial improvements across all critical parameters compared to traditional models such as RM, SVM, and RF. Through efficient preprocessing, adaptive feature selection, and layered deep learning architecture, DLTDF achieves a 29% increase in accuracy, a 34% rise in detection rate, an 18% decrease in false alarm rate, and a 22% reduction in response latency. These metrics confirm its superior ability to accurately detect and classify threats while maintaining minimal delays and false positives. The findings indicate that DLTDF is a reliable, scalable, and intelligent framework suitable for enhancing real-time intrusion detection and cybersecurity applications in complex cloud and network environments.

REFERENCE

[1] J. Trivedi, M. Tahir and J. Isoaho, "AI-Enhanced Threat Intelligence in Remote Patient Monitoring Systems: A Survey on Recent Advances, Challenges and Future Research Directions," in IEEE Access, vol. 13, pp. 106465-106488, 2025, doi: 10.1109/ACCESS.2025.3572626.

[2] O. Alkadi, N. Moustafa and B. Turnbull, "A Review of Intrusion Detection and Blockchain Applications in the Cloud: Approaches, Challenges and Solutions," in IEEE Access, vol. 8, pp. 104893-104917, 2020, doi: 10.1109/ACCESS.2020.2999715.

[3] B. Wang, B. Li and H. Li, "Oruta: privacy-preserving public auditing for shared data in the cloud," in IEEE Transactions on Cloud Computing, vol. 2, no. 1, pp. 43-56, Jan.-March 2014, doi: 10.1109/TCC.2014.2299807.

[4] R. Thaqi, B. Krasniqi, A. Mazrekaj and B. Rexha, "Literature Review of Machine Learning and Threat Intelligence in Cloud Security," in IEEE Access, vol. 13, pp. 11663-11678, 2025, doi: 10.1109/ACCESS.2025.3529636.

[5] B. Alouffi, M. Hasnain, A. Alharbi, W. Alosaimi, H. Alyami and M. Ayaz, "A Systematic Literature Review on Cloud Computing Security: Threats and Mitigation Strategies," in IEEE Access, vol. 9, pp. 57792-57807, 2021, doi: 10.1109/ACCESS.2021.3073203.

[6] A. V. Nagarjun and S. Rajkumar, "Design of an Anomaly Detection Framework for Delay and Privacy-Aware Blockchain-Based Cloud Deployments," in IEEE Access, vol. 12, pp. 84843-84862, 2024, doi: 10.1109/ACCESS.2024.3414998.

[7] D. He, X. Lv, X. Xu, S. Chan and K. -K. R. Choo, "Double-Layer Detection of Internal Threat in Enterprise Systems Based on Deep Learning," in IEEE Transactions on Information Forensics and Security, vol. 19, pp. 4741-4751, 2024, doi: 10.1109/TIFS.2024.3372771.

[8] Harshitha, S., Mahadevaswamy, U. B., & Srikantaswamy, M. (2025). Energy-Efficient Image Compression for Capsule Endoscopy Using a CNN-Based Feature Learning Algorithm. Engineering, Technology & Applied Science Research, 15(5), 26217–26223. https://doi.org/10.48084/etasr.11891.

[9] Basavaraju, N., Mahadevachar, V. K., & Srikantaswamy, M. (2025). An Enhanced Secure Framework for Detecting and Rectifying Unauthorized Access in Cloud Computing Environments Using the Elliptic Curve Digital Signature Algorithm. Engineering, Technology & Applied Science Research, 15(4), 24188–24195. https://doi.org/10.48084/etasr.9764.

[10] Shivappa, C., Ramaswamy, N. K., Mullegowda, R. C., Ramaswamy, R. K., & Srikantaswamy, M. (2025). Efficient Low-Complexity Encoding and Decoding Algorithms for Global Navigation Satellite Systems. Engineering, Technology & Applied Science Research, 15(4), 25499–25506. https://doi.org/10.48084/etasr.11457

[11] P. Kumar, R. Kumar, A. Jolfaei and N. Mohammad, "An Automated Threat Intelligence Framework for Vehicle–Road Cooperation Systems," in IEEE Internet of Things Journal, vol. 11, no. 22, pp. 35964-35974, 15 Nov.15, 2024, doi: 10.1109/JIOT.2024.3397652.

[12] M. Golec, Y. Khamayseh, S. B. Melhem and A. Alwarafy, "LLM-Driven APT Detection for 6G Wireless Networks: A Systematic Review and Taxonomy," in IEEE Access, vol. 13, pp. 145271-145288, 2025, doi: 10.1109/ACCESS.2025.3595665.
[13] Y. Chen et al., "Understanding the Security Risks of Websites Using Cloud Storage for Direct User File Uploads," in IEEE Transactions on Information Forensics and Security, vol. 20, pp. 2677-2692, 2025, doi: 10.1109/TIFS.2025.3544082.
[14] S. Hatkar, K. Rout, A. Lad and O. V. G. Swathika, "An Integrated UTM Solution for Modern Cybersecurity: Combining Deep Inspection, ML, and Policy Automation," in IEEE Access, vol. 13, pp. 176010-176023, 2025, doi: 10.1109/ACCESS.2025.3618281.
[15] S. S et al., "Smart Object Detection and Audio Feedback System for the Visually Impaired," 2025 International Conference on Innovations in Intelligent Systems: Advancements in Computing, Communication, and Cybersecurity (ISAC3), Bhubaneswar, India, 2025, pp. 1-6, doi: 10.1109/ISAC364032.2025.11156667.