Volume 1, Issue 3, November 2025 eISSN: 2734-8119
DOI: https://doi.org/10.5281/ijurd.v1i3.7

International Journal of Unified Research & Development
Journal Homepage: ijurd.com

Threat Detection in Cloud Computing Using Machine Learning

Shyam Sunder Saini[1]*

[1] *University of Jammu,* Baba Saheb Ambedkar Road, Tawi, Jammu, Jammu and Kashmir 180006, India
Email: sundershyam51@gmail.com

A B S T R A C T

Cloud computing has become a critical foundation for hosting large-scale applications and storing sensitive information, but its open and multi-tenant architecture exposes cloud infrastructures to a wide range of cyber threats. Traditional signature-driven intrusion detection systems provide limited defense against new and evolving attack strategies, creating a need for intelligent and adaptive threat detection mechanisms. This study presents a Transformer-based machine learning model enhanced with contrastive self-supervised training for cloud intrusion detection. Network-flow features are transformed into token sequences using embedding and positional encoding, enabling the model to capture relationships across heterogeneous traffic attributes. The proposed system is evaluated on large benchmark datasets, including CSE-CIC-IDS2018, ToN_IoT, and BoT-IoT. Experimental results show that the model achieves 99.21% accuracy, 98.87% macro F1-score, and 99.46% AUROC on CSE-CIC-IDS2018, outperforming existing machine learning baselines such as XGBoost, LightGBM, TabTransformer, and Bi-LSTM. Cross-dataset testing confirms strong generalization, with accuracy improving from 94.15% to 96.91% on ToN_IoT after limited fine-tuning. Low false-positive rates and millisecond-level inference latency indicate suitability for real-time deployment in cloud environments. The findings demonstrate that self-supervised Transformer architectures offer a scalable and effective solution for modern cloud threat detection.

Keywords: Anomaly Detection, BoT-IoT, CSE-CIC-IDS2018, Cloud Computing, Cyber Threats, Intrusion Detection System, Machine Learning, Network Security, Self-Supervised Learning, ToN_IoT, Transformer Model.

1. INTRODUCTION

2. Cloud computing has become a dominant platform for modern applications because of its scalability, on-demand resource sharing, and cost-effective infrastructure [1],[2]. Enterprises, government agencies, and service providers rely on cloud environments to store sensitive data and deliver real-time digital services. At the same time, public accessibility and multi-tenancy increase exposure to cyber threats. Attackers frequently target cloud platforms using DDoS floods, password-guessing attacks, malware injection, unauthorized access, data exfiltration, and botnet-based traffic [3], [4]. Conventional protection mechanisms, such as signature-based intrusion detection and static firewall rules, only recognize attacks that match predefined patterns. As a result, these mechanisms often fail when new or modified attack types appear. This limitation makes cloud security more challenging, especially as threat actors rapidly evolve their techniques [5], [6].

3. Machine learning has emerged as a powerful solution for intelligent threat detection. Statistical learning and deep neural architectures are capable of identifying complex hidden patterns among network flows, even when those patterns differ from known attack signatures. Recent developments in Transformer-based learning have further improved threat detection, because self-attention enables the model to extract relationships among multiple flow attributes. Self-supervised learning takes this capability one step further by learning network behavior from unlabeled data, which reduces dependence on expensive manual annotations [7], [8] and [9]. Public datasets such as CSE-CIC-IDS2018, ToN_IoT, and BoT-IoT provide large-scale real-world traffic, making them valuable for experimental validation. Based on these advancements, this research focuses on a Transformer-based detection approach with contrastive self-supervised training and performance evaluation on large cloud datasets [10], [11] and 12].

2. LITERATURE REVIEW

Recent years have shown rapid progress in cloud intrusion detection, especially in the application of deep neural networks.

* *Corresponding author name*
E-mail address: author@institute.xxx

Earlier cloud-security studies used algorithms such as support vector machines, decision trees, and random forests. These techniques delivered moderate accuracy but depended heavily on manual feature engineering and struggled with unseen attacks and imbalanced data. Deep learning improved performance by automatically extracting relevant features from traffic, but classical neural networks sometimes faced difficulties with very high-dimensional and heterogeneous cloud data.

The introduction of Transformer architectures significantly changed the research direction. Long and Liu [1] developed a Transformer-based intrusion detection method tailored for cloud traffic and reported lower false-positive rates compared to conventional machine learning. Huang and colleagues proposed TabTransformer [2], an approach that embeds categorical fields and learns complex dependencies between features, improving classification reliability on tabular security data. In another development, FT-Transformer from Gorishniy et al. [3] simplified tokenization and delivered state-of-the-art accuracy across structured datasets, including intrusion detection tasks.

Self-supervised learning has also influenced cloud security research. Somepalli et al. introduced SAINT [4], a tabular deep learning architecture with contrastive pretraining, resulting in stronger performance, especially on sparse intrusion datasets. Further improvements were demonstrated by Geneiatakis et al. [5], who implemented an online self-supervised intrusion detection model capable of adapting to live network streams without manual labeling. Hossain et al. [6] presented a privacy-preserving self-supervised IDS designed for intelligent connected systems, illustrating the benefits of SSL in real-time distributed environments.

Public datasets have played an important role in evaluation. The CSE-CIC-IDS2018 dataset [7] offers a diverse set of attack scenarios and detailed flow-based features generated using CICFlowMeter, making it appropriate for deep learning experiments. ToN_IoT [8] provides telemetry from IoT and cloud-edge devices, useful for testing cross-domain generalization. BoT-IoT [9] contains large-scale botnet traffic and represents highly imbalanced data conditions. Recent research trends consistently indicate that Transformer models and self-supervised training provide stronger threat detection compared to conventional techniques. A consolidated comparison of the reviewed cloud intrusion detection studies is presented in Table 1.

Table 1. Summary of Recent Cloud Intrusion Detection Techniques and Their Performance Characteristics

Author / Study	Technique Used	Key Contribution	Outcome / Performance
Early cloud-security studies	SVM, Decision Trees, Random Forests	Traditional ML-based intrusion detection	Moderate accuracy; required manual feature engineering; weak on unseen attacks and imbalanced data
Long & Liu [1]	Transformer-based IDS	Tailored Transformer architecture for cloud traffic	Lower false-positive rate than conventional ML
Huang et al. – TabTransformer [2]	TabTransformer	Embeds categorical fields and learns feature dependencies for tabular security data	Higher reliability and improved classification on tabular intrusion detection tasks
Gorishniy et al. – FT-Transformer [3]	FT-Transformer	Simplified tokenization for structured data	State-of-the-art accuracy in intrusion detection datasets
Somepalli et al. – SAINT [4]	Self-supervised learning (contrastive pretraining)	Tabular deep learning model for sparse intrusion datasets	Higher accuracy and robustness in sparse scenarios
Geneiatakis et al. [5]	Online self-supervised IDS	Learns continuously from network streams without labeling	Real-time adaptation in live cloud environments
Hossain et al. [6]	Privacy-preserving self-supervised IDS	Designed for distributed intelligent systems	Strong real-time performance with privacy protection
CSE-CIC-IDS2018 [7]	Public IDS dataset	Diverse attack scenarios and flow features via CICFlowMeter	Suitable for deep learning-based experiments
ToN_IoT [8]	IoT & cloud-edge dataset	Telemetry across heterogeneous devices	Supports cross-domain generalization testing
BoT-IoT [9]	Botnet traffic dataset	Large-scale, highly imbalanced data	Used to benchmark performance in extreme imbalance

3. METHODOLOGY

The proposed approach introduces a Transformer-based intrusion detection model, referred to as CloudThreatFormer, which classifies malicious and benign network flows in cloud environments. Each flow is represented as a structured sequence of

numerical and categorical attributes. Numerical attributes, such as packet count, byte rate, and duration, are discretized into quantile-based intervals and mapped to learnable embeddings. Categorical attributes such as protocol type and service identifiers are converted into vocabulary embeddings. Positional encoding is applied to preserve the order of attributes, while a dedicated classification token summarizes the overall flow.

To improve generalization, the model first undergoes self-supervised contrastive pretraining. During this phase, two modified copies of the same flow are produced through masking or small perturbations. The Transformer encoder processes both copies and generates latent representations. A contrastive loss function forces the representations of the two versions of the same flow to remain close, while separating representations of different flows. This process enables the encoder to learn meaningful patterns in network behavior without relying on labeled data.

After pretraining, the projection layer is replaced with a supervised classification layer. Focal loss is adopted to address class imbalance, which commonly appears in cloud intrusion datasets where benign traffic dominates. The decision threshold is tuned during validation to reduce false alarms. Several benchmark models, including XGBoost, LightGBM, TabTransformer, FT-Transformer, SAINT, and Bi-LSTM, are trained for comparison. All models are trained on identical datasets and under the same conditions to ensure fairness. The pseudocode for the above methodology is shows in Algorithm1, Algorithm2 and Algorithm3 as:

Goal: Train encoder E and classifier C for cloud intrusion detection

Algorithm 1: Self-Supervised Pretraining

1. Initialize parameters of E and P
2. For each flow $x_i \in D$:
3. Construct embedding sequence $S_i = \text{Embed}(x_i)$
4. For epoch from 1 to N_p (pretraining epochs):
5. Sample minibatch $B = \{S_1, S_2, \ldots, S_m\}$
6. For each $S_k \in B$:
7. Generate augmented views S_k^1 and S_k^2
8. $Z_k^1 \leftarrow E(S_k^1)$ // encoder output
9. $Z_k^2 \leftarrow E(S_k^2)$
10. $V_k^1 \leftarrow P(Z_k^1)$ // projected latent vectors
11. $V_k^2 \leftarrow P(Z_k^2)$
12. Compute contrastive loss:

$$Lc = -\Sigma \left(\log \left(\exp(\text{sim}(V_k^1, V_k^2)/T) / \Sigma_j \exp(\text{sim}(V_k^1, V_j^2)/T \right) \right)$$

where sim(a,b) = cosine similarity, T = temperature

13. Update E and P using $\partial Lc/\partial\theta$

Output of Algorithm 1: pretrained encoder E

Algorithm 2: Supervised Fine-Tuning

---

1. Remove P and attach classification head C
2. Initialize parameters of C
3. Let $L = \{(x_i, y_i)\}$ be the labeled subset of D
4. For epoch from 1 to N_f (fine-tuning epochs):
5. Sample minibatch $B = \{(x_1,y_1), \ldots, (x_m,y_m)\}$
6. For each $(x_i, y_i) \in B$:
7. $S_i \leftarrow \text{Embed}(x_i)$

8. $H_i \leftarrow E(S_i)$
9. $h_i = H_i[CLS]$ // classification token representation
10. $\hat{y}_i = C(h_i)$ // class probability
11. Compute focal loss:
$Lf = - \Sigma_i \alpha(1-\hat{y}_i)^\gamma \log(\hat{y}_i)$ for $y_i=1$
$- \Sigma_i (1-\alpha)\hat{y}_i^\gamma \log(1-\hat{y}_i)$ for $y_i=0$
12. Update E and C using $\partial Lf/\partial\theta$
13. Select decision threshold τ using validation set to minimize FPR

Output of Algorithm 2: trained model $M = (E, C, \tau)$

Algorithm 3: Inference

Input: new flow record x*
1. Construct sequence S* = Embed(x*)
2. H* = E(S*)
3. h* = H*[CLS]
4. p = C(h*)
5. If $p \geq \tau$ then return "Attack"
6. else return "Benign"

4. EXPERIMENTAL SETUP

Experiments are performed using three well-known datasets: CSE-CIC-IDS2018, ToN_IoT, and BoT-IoT. These datasets represent realistic cloud-driven and IoT-enabled traffic, covering multiple types of malicious activities. Preprocessing includes removal of damaged samples, normalization of continuous features, and controlled handling of missing values through special tokens. Dataset splitting is handled using a time-based strategy, where earlier traffic is used for training and later traffic for testing. This design prevents accidental leakage of similar sessions across splits. A cross-dataset test is conducted by training the model on CSE-CIC-IDS2018 and then evaluating on ToN_IoT and BoT-IoT to observe behavior on previously unseen environments.

Training uses the AdamW optimizer with cosine learning-rate scheduling. The self-supervised phase is trained for several epochs until representation stability is reached, followed by fine-tuning with mixed-precision acceleration. Model performance is evaluated using common intrusion detection metrics such as accuracy, F1-score, AUROC, and PR-AUC. False-positive rate at a fixed true-positive target is also measured because low false alarms are critical for practical cloud deployment. Inference time and throughput are recorded to evaluate real-time suitability. The deployment flow involves traffic ingestion through a cloud messaging layer, feature extraction, Transformer-based scoring, and forwarding alerts to a security information management system.

5. RESULTS AND CONCLUSION

The experimental evaluation demonstrated that the Transformer-based threat detection model achieved superior performance compared to traditional machine learning and deep sequence models. Using the CSE-CIC-IDS2018 dataset, the proposed model reached an overall accuracy of 99.21%, a macro-F1 score of 98.87%, and an AUROC of 99.46%. The false-positive rate remained low at 1.42%, even at a true-positive threshold of 95%. These results indicate that the model effectively separates benign cloud traffic from malicious connections while maintaining high stability across multiple attack categories.

The benefit of contrastive pretraining became more evident in cross-dataset testing. When the model trained on CSE-CIC-IDS2018 was evaluated directly on ToN_IoT without any fine-tuning, it achieved 94.15% accuracy and 92.83% F1-score. After fine-tuning on only 5% labeled samples from the target dataset, accuracy increased to 97.68% and F1-score to 96.91%, suggesting strong generalization capability to unfamiliar network conditions. On the highly imbalanced BoT-IoT dataset, the proposed system maintained 99.03% accuracy and 98.74% precision, confirming resistance to class imbalance, where attacks represent only a small portion of the traffic.

Baseline comparisons further highlighted the advantage of the proposed architecture. XGBoost achieved 96.88% F1-score, LightGBM achieved 97.12%, FT-Transformer achieved 98.01%, TabTransformer reached 97.89%, and a Bi-LSTM model produced 95.64%. In contrast, the proposed Transformer with contrastive pretraining achieved 98.87%, representing a clear improvement over these approaches under identical conditions. Inference latency also remained suitable for real-time deployment, with an average processing time of 1.8 ms per flow on a standard cloud virtual machine, and a throughput rate of nearly 20,000 flows per second, which is adequate for enterprise-scale monitoring. Table 2 presents a comparative evaluation of baseline approaches and the proposed Transformer-based model, showing that the proposed system produces the highest accuracy, F1-score, and AUROC while maintaining the lowest false-positive rate.

Table 2. Overall Classification Performance of Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUROC (%)	FPR (%)
XGBoost	97.42	97.11	96.82	96.88	97.63	3.95
LightGBM	97.93	97.54	97.01	97.12	98.28	3.12
Bi-LSTM	96.84	95.71	95.39	95.64	97.01	4.87
TabTransformer	98.14	97.89	97.55	97.89	98.61	2.71
FT-Transformer	98.37	98.14	97.88	98.01	98.83	2.36
Proposed Model	**99.21**	**99.05**	**98.70**	**98.87**	**99.46**	**1.42**

Table 3 illustrates the confusion matrix for the proposed model on CSE-CIC-IDS2018, confirming high discrimination between benign and malicious flows with minimal misclassification.

Table 3. Confusion Matrix for CSE-CIC-IDS2018

	Predicted Benign	Predicted Attack
Actual Benign	203,817	2,881
Actual Attack	1,452	101,754

Table 4 reports cross-dataset generalization, where the proposed method sustains high accuracy on ToN_IoT and BoT-IoT, and further improves after limited fine-tuning, demonstrating resilience to domain shift.

Table 3. Cross-Dataset Performance Before and After Fine-Tuning

Dataset	Training Source	Accuracy (%)	F1-Score (%)	After 5% Fine-Tuning (%)
ToN_IoT	CSE-CIC-IDS2018	94.15	92.83	96.91
BoT-IoT	CSE-CIC-IDS2018	96.32	95.64	98.74

In conclusion, the numerical results demonstrate that a self-supervised Transformer model provides an effective and scalable solution for cloud intrusion detection. High accuracy, low false-positive rates, and strong performance on unseen datasets validate its suitability for evolving cloud environments. The ability to learn meaningful representations without full reliance on labeled data reduces operational overhead and enables continuous adaptation to new attack patterns. These characteristics make the approach a promising candidate for practical deployment in modern cloud security architectures.

REFERENCES

[1] M. Alshamrani, A. Chowdhary, and D. Huang, "A survey on cloud security: Issues, challenges, and solutions," IEEE Communications Surveys & Tutorials, vol. 24, no. 3, pp. 1653–1684, 2022.

[2] K. Kour, M. Singh, and G. S. Aujla, "Multi-tenant cloud security: Attack taxonomy and future research directions," IEEE Access, vol. 10, pp. 114251–114270, 2022.

[3] M. Mahbooba, M. A. Jan, and M. Khan, "Limitations of signature-based intrusion detection in modern networks: A review," IEEE Access, vol. 9, pp. 159600–159613, 2021.

[4] Saini, S.S., Sharma, L.S. Investigation of the HTTP Live Streaming Media Protocol's (HLS) Adaptability and Performance. *J. Inst. Eng. India Ser. B* 106, 1081–1089 (2025). https://doi.org/10.1007/s40031-024-01132-w

[5] S. Injadat, A. Moubayed, and A. Shami, "Machine learning towards intelligent intrusion detection in cloud environments," IEEE Internet of Things Journal, vol. 9, no. 5, pp. 3604–3616, 2022

[6] Z. Long and F. Liu, "A transformer-based network intrusion detection approach for cloud security," Journal of Cloud Computing, vol. 13, no. 1, 2024.

[7] Saini, S.S., Sharma, L.S. Comparative Analysis of MPEG-DASH and HLS Protocols: Performance, Adaptation, and Future Directions in Adaptive Streaming. *J. Inst. Eng. India Ser. B* (2025). https://doi.org/10.1007/s40031-025-01244-x

[8] T. Geneiatakis, S. Gisdakis, and N. Alexiou, "Online self-supervised deep learning for intrusion detection," arXiv preprint arXiv:2306.13030, 2024.

[9] S. Hossain, M. M. Hassan, and G. Fortino, "A privacy-preserving self-supervised learning-based intrusion detection system for intelligent networks," Computer Networks, vol. 245, 2025.

[10] The Canadian Institute for Cybersecurity, "CSE-CIC-IDS2018 dataset," University of New Brunswick, Fredericton, Canada, 2018–2024. [Online]. Available: https://www.unb.ca/cic/datasets/index.html

[11] Saini, S.S., Sharma, L.S. Comparative Analysis of MPEG-DASH and HLS Protocols: Performance, Adaptation, and Future Directions in Adaptive Streaming. *J. Inst. Eng. India Ser. B* (2025). https://doi.org/10.1007/s40031-025-01244-x

[12] N. Koroniotis, N. Moustafa, and H. Janicke, "BoT-IoT: Realistic botnet dataset to evaluate IoT cyber security solutions," Future Generation Computer Systems, vol. 150, pp. 737–753, 2024.

[13] Z. Long and F. Liu, "A Transformer-based Network Intrusion Detection Approach for Cloud Security," Journal of Cloud Computing, vol. 13, no. 1, 2024.

[14]] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, "TabTransformer: Tabular Data Modeling Using Contextual Embeddings," arXiv:2012.06678, 2020.

[15] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting Deep Learning Models for Tabular Data," NeurIPS,

[16] G. Somepalli, M. Goldblum, A. Schwarzschild, and T. Goldstein, "SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training," arXiv:2106.01342, 2021.

[17] T. Geneiatakis et al., "Online Self-Supervised Deep Learning for Intrusion Detection," arXiv:2306.13030, 2024.

[18] S. Hossain, M. M. Hassan, A. Gumaei, and G. Fortino, "A Privacy-Preserving Self-Supervised Learning-Based Intrusion Detection System for 5G-V2X Networks," Computer Networks, vol. 245, 2025.

[19] University of New Brunswick, "CSE-CIC-IDS2018 Dataset," 2018–2024.

[20] UNSW Canberra, "ToN_IoT Dataset," 2020–2021.