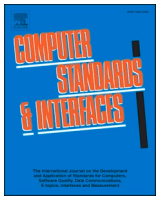
Computer Standards & Interfaces 97 (2026) 104119

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi



SiamIDS: A novel cloud-centric Siamese Bi-LSTM framework for interpretable intrusion detection in large-scale IoT networks

Prabu Kaliyaperumal [a], Palani Latha [b], Selvaraj Palanisamy [a], Sridhar Pushpanathan [c], Anand Nayyar [d,*], Balamurugan Balusamy [e], Ahmad Alkhayyat [f]

[a] School of Computer Science and Engineering, Galgotias University, Delhi NCR, India
[b] Department of Information Technology, Panimalar Engineering College, Chennai, India
[c] Department of Electrical and Electronics Engineering, Kongunadu College of Engineering and Technology, Trichy, India
[d] School of Computer Science, Duy Tan University, Da Nang 550000, Viet Nam
[e] School of Engineering and IT, Manipal Academy of Higher Education, Dubai Campus, Dubai, United Arab Emirates
[f] Department of Computer Techniques Engineering, College of Technical Engineering, The Islamic University, Najaf, Iraq

ARTICLE INFO

Keywords:
Siamese network
IoT security
Intrusion detection
SHAP
Clustering

ABSTRACT

The rapid proliferation of Internet of Things (IoT) devices has heightened the need for scalable and interpretable intrusion detection systems (IDS) capable of operating efficiently in cloud-centric environments. Existing IDS approaches often struggle with real-time processing, zero-day attack detection, and model transparency. To address these challenges, this paper proposes SiamIDS, a novel cloud-native framework that integrates contrastive Siamese Bi-directional LSTM (Bi-LSTM) modeling, autoencoder-based dimensionality reduction, SHapley Additive exPlanations (SHAP) for interpretability, and Ordering Points To Identify the Clustering Structure (OPTICS) clustering for unsupervised threat categorization. The framework aims to enhance the detection of both known and previously unseen threats in large-scale IoT networks by learning behavioral similarity across network flows. Trained on the CIC IoT-DIAD 2024 dataset, SiamIDS achieves superior detection performance with an F1-score of 99.45%, recall of 98.96%, and precision of 99.94%. Post-detection OPTICS clustering yields a Silhouette Score of 0.901, DBI of 0.092, and ARI of 0.889, supporting accurate threat grouping. The system processes over 220,000 samples/sec with a RAM usage under 1.5 GB, demonstrating real-time readiness. Compared to state-of-the-art methods, SiamIDS improves F1-score by 2.8% and reduces resource overhead by up to 25%, establishing itself as an accurate, efficient, and explainable IDS for next-generation IoT ecosystems.

1. Introduction

With the explosive growth of digital transformation across industries, the convergence of the Internet of Things (IoT) and cloud computing has revolutionized modern infrastructure. From smart homes and healthcare monitoring to industrial automation and intelligent transportation systems, IoT devices now generate massive volumes of data that are often offloaded to cloud platforms for centralized processing and storage [1,2]. According to a recent IDC report, over 41.6 billion IoT devices are expected to be connected by 2025, producing 79.4 zettabytes of data [3]. This hyperconnectivity, while enabling operational efficiency and real-time analytics, has significantly broadened the attack surface, making cybersecurity a critical concern for both cloud and IoT ecosystems [4,5]. In such environments, cyber threats like ransomware, botnets, Distributed Denial-of-Service (DDoS) attacks, and zero-day vulnerabilities have become increasingly sophisticated and frequent [6]. These threats not only exploit system vulnerabilities and insecure communication channels but also leverage the lack of consistent security policies across distributed endpoints. As organizations increasingly rely on cloud-centric infrastructures to host critical services, ensuring end-to-end security—especially across low-power, heterogeneous IoT nodes—has become both a necessity and a challenge [7,

* Corresponding author.
E-mail addresses: k.prabu@galgotiasuniversity.edu.in (P. Kaliyaperumal), lathapalani@panimalar.ac.in (P. Latha), p.mselvaraj@galgotiasuniversity.edu.in (S. Palanisamy), sridharp@kongunadu.ac.in (S. Pushpanathan), anandnayyar@duytan.edu.vn (A. Nayyar), kadavulai@gmail.com (B. Balusamy), ahmedalkhayyat85@iunajaf.edu.iq (A. Alkhayyat).

https://doi.org/10.1016/j.csi.2025.104119
Received 1 August 2025; Received in revised form 16 October 2025; Accepted 15 December 2025
Available online 15 December 2025

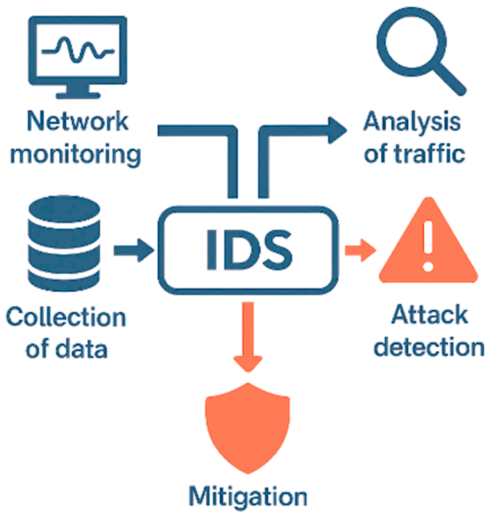


Fig. 1. Workflow of an Intrusion Detection System in cloud-centric IoT environments.

Fig. 2. An overview of cloud-centric IoT infrastructure.

8].

To defend against such multifaceted threats, Intrusion Detection Systems (IDS) have emerged as a cornerstone of modern cybersecurity architectures [9]. As illustrated in Fig. 1, an IDS monitors system and network traffic for signs of unauthorized or anomalous activities. IDS mechanisms are broadly classified into two categories [10]: signature-based detection, which matches observed behaviors with a predefined set of known attack patterns, and anomaly-based detection, which identifies deviations from established normal behavior. While signature-based methods offer high precision for known threats, they are ineffective against new or evolving attack types. Anomaly-based IDS, on the other hand, provide flexibility and the ability to detect zero-day attacks but often suffer from high false alarm rates due to the difficulty of accurately modeling "normal" behavior [11,12].

Traditional IDS frameworks were initially designed for homogeneous, resource-rich enterprise networks. These systems typically assumed structured traffic flows, consistent device capabilities, and access to reliable computational resources [13,14]. However, the IoT paradigm introduces a set of conditions that challenge these assumptions: highly heterogeneous devices, constrained memory and compute power, varied communication protocols, and intermittent connectivity. Furthermore, many IoT nodes are deployed with minimal configurations and legacy firmware, making them attractive entry points for attackers [15]. Studies reveal that IoT-based attacks have surged by more than 300 % in the last five years, with incidents such as the Mirai botnet compromising millions of devices globally [16]. As depicted in Fig. 2, IoT ecosystems interact with edge devices, fog layers, and cloud services, forming a multi-layered infrastructure with dynamic data flows. These interconnected systems introduce new vulnerabilities, particularly in resource coordination, data aggregation, and service orchestration. In cloud-centric environments, threats may propagate from the edge to the core or vice versa, requiring real-time threat detection and response mechanisms that are not only accurate but also interpretable and scalable.

Despite the growing need for intelligent IDS models in IoT-cloud environments, current techniques face several critical limitations. First, many machine learning-based IDS solutions are trained in a supervised fashion, heavily reliant on labeled datasets that do not reflect the diversity of real-world attacks. Second, most existing models lack interpretability, rendering them less useful for human operators in Security Operations Centers (SOCs) who must understand and act upon alerts. Third, these models often fail to meet the constraints of cloud-edge deployments due to high computational or memory requirements. Lastly, many IDS do not provide mechanisms for grouping detected anomalies into meaningful patterns, limiting post-detection forensics and threat hunting capabilities.

The above limitations highlight the urgent need for a robust, cloud-ready, interpretable, and generalizable IDS framework that can adapt to the unique characteristics of large-scale IoT environments. The ability to not only detect zero-day attacks but also explain the detection rationale in human-understandable terms is becoming increasingly critical. Furthermore, supporting scalability and low-latency processing is essential for real-time operation across distributed edge-cloud networks. Recognizing these demands, this research proposes an advanced solution that integrates deep metric learning, unsupervised clustering, and explainable AI (XAI) to create a holistic and effective intrusion detection pipeline.

This study focuses on designing an intelligent, scalable, and explainable intrusion detection system (IDS) optimized for cloud-centric IoT networks. The scope encompasses flow-based traffic monitoring, similarity-driven anomaly detection, post-detection behavior analysis, and explainable threat attribution. The key problem addressed is the lack of unified IDS frameworks that can simultaneously handle unseen threats, offer transparency, and operate efficiently in resource-constrained IoT-cloud environments.

To overcome this, we introduce SiamIDS—a Siamese Bi-LSTM-based intrusion detection system—that incorporates contrastive learning, autoencoder-based compression, SHAP-based interpretability, and OPTICS clustering for semantic anomaly grouping. This approach enables similarity-driven detection that is capable of generalizing to novel behaviours while offering detailed reasoning through feature contribution analysis.

1.1. Objectives of the paper

The objectives of the paper are:

1. To conduct a comprehensive background study and literature review on the design of scalable and interpretable intrusion detection systems for IoT networks;
2. To propose a novel methodology titled SiamIDS for detecting and explaining known and zero-day cyber threats in large-scale IoT traffic. The novelty lies in combining contrastive similarity learning with interpretable SHAP analysis and unsupervised clustering to enhance both accuracy and transparency;
3. To test and validate the proposed SiamIDS framework using metrics such as F1-score, precision, recall, Silhouette Score, DBI, ARI, inference speed, and memory footprint;
4. And, to compare SiamIDS with existing techniques, including CNN, Bi-LSTM, GRU, AE, and traditional statistical baselines, across multiple attack categories in the CIC IoT-DIAD 2024 dataset.

1.2. Organization of paper

The rest of the paper is organized as: Section 2 presents a detailed literature review, highlighting recent advancements and challenges in intrusion detection systems for IoT networks. Section 3 discusses the Materials and Methods used in this study, covering the dataset, preprocessing steps, and the foundational methods employed to build the proposed SiamIDS framework. Section 4 presents the proposed methodology, explaining the architectural design and key components of SiamIDS. Section 5 focuses on Experimentation, Results, and Analysis. And, Finally, Section 6 concludes the paper with key outcomes, limitations, and directions for future research.

2. Literature review

The rapid growth of Internet of Things (IoT) devices has brought forth new challenges in network security, especially in cloud-centric architectures where massive volumes of traffic are continuously generated. As a result, Intrusion Detection Systems (IDS) have gained significant attention in recent literature, with various machine learning (ML) and deep learning (DL) approaches being explored to tackle the complexity of modern threats. This section reviews existing IDS models with a focus on approaches leveraging Siamese networks, sequence learning (e.g., LSTM, Bi-LSTM), contrastive learning, and interpretability frameworks such as SHAP. We also examine clustering techniques like OPTICS used for post-detection analysis. Each work is evaluated based on its methodology, effectiveness, explainability, and suitability for real-time deployment in large-scale IoT or cloud environments.

Bedi et al. (2020) [17] addressed the class imbalance issue in IDS by proposing a DNN-based Siamese architecture trained using contrastive loss. Their model effectively improved recall for rare attack types like U2R and R2L in the NSL-KDD dataset. Although effective in similarity-based detection, it lacked temporal modeling, interpretability, and cloud deployment support. SiamIDS adopts this contrastive learning principle but enhances it with Bi-LSTM temporal encoding, SHAP-based explainability, and scalable cloud-oriented integration

Saurabh et al. (2022) [18] proposed LBDMIDS, a Bi-LSTM and Stacked LSTM-based model evaluated on UNSW-NB15 and Bot-IoT datasets. The model used Z-score normalization and sequence slicing for temporal analysis, achieving over 99 % accuracy on Bot-IoT. While this supports temporal modeling, the approach lacks interpretability, similarity-based learning, and clustering capabilities. SiamIDS advances this by combining Bi-LSTM with Siamese contrastive training, adding SHAP explanations, and applying OPTICS clustering to analyze novel threats in cloud settings.

Aldaej et al. (2023) [19] presents a Bi-LSTM-based IDS deployed in a distributed cloud–edge architecture. The authors applied dimensionality reduction (GMDH, Chi2) and trained RNN/Bi-LSTM models on BoT-IoT, demonstrating scalable inference for edge environments. The study emphasized reduced computational complexity and deployment feasibility. However, it lacks interpretability, similarity learning, and does not explore contrastive pair-based detection. SiamIDS builds on this foundation by adding SHAP-based interpretability, contrastive Bi-LSTM modeling, and a cloud-centric inference design.

Hindy (2023) [20] introduced a one-shot Siamese learning model to detect zero-day attacks by learning distance metrics from traffic pairs. The method achieved strong generalization on CICIDS2017 and NSL-KDD, reducing retraining requirements. However, it employed basic MLP-based twin networks and did not incorporate sequence modeling or interpretability. SiamIDS builds upon this foundation with a Bi-LSTM-based Siamese backbone, feature compression, SHAP-based decision explanation, and unsupervised clustering to further enhance detection granularity and transparency.

Madhu et al. (2023) [21] introduces a deep learning framework for intrusion detection in smart home IoT networks using TabNet and CNN. It emphasizes device-specific modeling and evaluates traditional ML and DL approaches on real-world IoT traffic. Though effective, it lacks any temporal modeling, similarity learning, or explainability. Additionally, cloud deployment strategies were not explored. SiamIDS distinguishes itself by offering temporal contrastive learning, explainability through SHAP, and real-time cloud deployment features tailored for IoT environments.

Hnamte & Hussain (2023) [22] proposed DCNNBILSTM, a hybrid intrusion detection system combining CNN for feature extraction, BiLSTM for sequence learning, and DNN layers for classification. The methodology includes thorough data preprocessing and the use of ReLU, Softmax, and Adam optimizer. Trained on CICIDS2018 and Edge_IIoT datasets, it achieved 100 % and 99.64 % accuracy, respectively, with F1-score up to 100 %, and minimal loss rate (0.0080). The novelty lies in integrating deep CNN with BiLSTM for robust detection. Limitations include longer training times due to model complexity, suggesting future optimization for real-time deployment.

Alzboon et al. (2023) [23] proposed a novel IDS combining FLAME-based feature filtration and an enhanced extended classifier system (XCS) with genetic algorithm and cuckoo search optimization. This hybrid methodology was tested on the KDD99 dataset after reducing feature dimensions from 41 to 20. The enhanced model achieved 100 % detection rate, 99.99 % accuracy, 0.05 % FAR, and high precision, recall, specificity, and F1-score. The novelty lies in integrating CS for adaptive rule selection within GA to improve classifier breeding. Limitations include reliance on FLAME's density-based clustering and a focus on a single dataset, which may affect generalizability to newer threats.

Ben Said et al. (2023) [24] proposed a CNN-BiLSTM hybrid deep learning model for Network Intrusion Detection in Software-Defined Networking (SDN). The methodology integrates spatial and temporal feature extraction with regularization and dropout optimization. Using InSDN, NSL-KDD, and UNSW-NB15 datasets, the model achieved up to 97.77 % accuracy, 99.85 % precision, 95.28 % recall, 100 % specificity, and F1-scores over 97 %. The novelty lies in combining BiLSTM's contextual memory with CNN's hierarchical feature extraction for SDN-specific threats. Limitations include longer training time and reliance on handcrafted feature selection.

Zhang et al. (2023) [25] introduced a BiLSTM-based network intrusion detection model enhanced by a multi-head attention mechanism to refine feature relationships. The methodology included embedding, attention-driven weighting, and bidirectional temporal analysis. Tested on KDDCUP99, NSLKDD, and CICIDS2017 datasets, the model achieved accuracies of 98.29 %, 95.19 %, and 99.08 %, respectively, with F1-scores up to 99 %. Precision and recall exceeded 97 % on most classes. The novelty lies in combining multi-head attention with BiLSTM to capture bidirectional dependencies while adaptively weighting features. However, the model struggles to identify unknown attack types and may lose critical information during under sampling, affecting robustness in real-world deployments.

Hou et al. (2023) [26] introduced LCVAE-CBiLSTM, a hybrid intrusion detection method combining Log-Cosh Conditional Variational Autoencoder (LCVAE) for minority class sample generation with CNN-BiLSTM for spatiotemporal feature extraction. The NSL-KDD dataset was used. The model achieved 87.30 % accuracy, 80.89 % recall, 96.08 % precision, 87.89 % F1-score, and a FAR of 4.36 %. The novelty lies in using log-cosh loss to improve generative reconstruction and mitigate gradient explosion, enhancing minority attack detection. Limitations include sensitivity bias across attack types and reduced performance for certain 0-day and rare attacks.

Ali et al. (2023) [27] proposed a dual-layer intrusion detection framework combining Shuffle Shepherd Optimization (SSO)-based feature selection and LSTM for classification, reinforced with SHA3–256 hash functions for intrusion prevention. The methodology includes real-time data normalization, optimal feature filtration via SSO, and sequential attack detection. Evaluated on KDDCUP99 and UNSW-NB15

datasets, results show 99.92 % (KDDCUP99) and 99.91 % (UNSW-NB15) accuracy; precision at 98 %, recall at 98.2 %, specificity near 99 %, F1-score at 98 %, and extremely low FNR (0.001). Limitations include real-time online validation only; the model lacks adaptability for cross-domain threat intelligence and faces constraints under ultra-high-speed traffic.

Jiang et al. (2023) [28] proposed FR-APPSO-BiLSTM, a network anomaly detection model combining feature reduction via hierarchical clustering and autoencoders with an improved PSO algorithm for BiLSTM optimization. Tested on NSL-KDD, UNSW-NB15, and CICIDS-2017 datasets, the model achieved up to 95.44 % accuracy, 98.58 % precision, 98.40 % recall, 99.92 % specificity, and 98.49 % F1-score. Novelty lies in adaptive velocity and position updates, and dynamic parameter tuning within PSO, enhancing BiLSTM's performance. Limitations include scalability challenges in high-speed networks and potential sensitivity to feature subset selection.

Yaras and Dener (2024) [29] developed a hybrid model combining 1D-CNN and LSTM, optimized for scalable environments using PySpark and Google Colab. Their model, tested on CICIoT2023 and TON_IoT, achieved high accuracy without data balancing techniques. The work confirms the value of hybrid DL for IoT traffic but lacks contrastive learning, explainability, or behavior clustering. SiamIDS extends this by integrating Bi-LSTM within a Siamese structure and offering SHAP-based insights and OPTICS-based threat clustering for real-time analysis.

Althiyabi et al. (2024) [30] proposed a few-shot intrusion detection model using 1D-CNN and Prototypical Networks, evaluated on CICIDS2017 and MQTT-IoT datasets. The model achieved high performance under limited data conditions (5-shot and 10-shot settings), supporting rare class detection. However, it lacked temporal analysis, interpretability, and similarity-based reasoning. SiamIDS similarly targets zero-day detection but incorporates Bi-LSTM Siamese modeling and SHAP explanations, with additional OPTICS clustering to reveal behavioral groupings among anomalies.

Bo et al. (2024) [31] developed a few-shot intrusion detection model integrating Adaptive Feature Fusion (AFF) with Prototypical Networks. Using CICIDS2017 and ISCX2012, the system achieved over 99 % accuracy with minimal labeled data, thanks to feature diversity from binary and statistical sources. Despite this, it lacks temporal modeling and explainability, and does not address post-detection analysis like clustering. SiamIDS takes a step further by employing Bi-LSTM for sequence modeling, SHAP for decision transparency, and OPTICS for behavioral analysis.

Touré et al. (2024) [32] proposed a hybrid zero-day attack detection framework combining supervised (CNN, DT, RF, KNN, NB) and unsupervised (K-Means) learning with online adaptation. The methodology includes flow feature engineering, anomaly identification via silhouette-based clustering, and new class validation through online learning. Experiments were conducted on IBM real-time network flows and NSL-KDD datasets. Results show high accuracy: 98.4 % (IBM), 96.6 % (NSL-KDD); F1-score up to 99 %, specificity and precision above 98 %, and recall exceeding 97 %. Limitations include dependence on clustering thresholds and need for periodic model retraining to maintain real-time responsiveness.

Chintapalli et al. (2024) [33] proposed an intrusion detection framework for IoT systems using OOA-modified Bi-LSTM with ELU activation for robust sequence learning. The Osprey Optimization Algorithm (OOA) selected informative features from N-BaIoT, CICIDS-2017, and ToN-IoT datasets. The model achieved impressive results: N-BaIoT (99.98 % accuracy, 99.94 % recall, 99.90 % precision, 99.89 % F1, 99.90 % specificity), CICIDS-2017 (99.97 % accuracy, 99.91 % recall, 99.96 % F1), and ToN-IoT (99.88 % accuracy, 99.89 % recall, 99.90 % F1). The novelty lies in integrating OOA for feature selection and ELU to avoid vanishing gradients. Limitations include reliance on predefined datasets and absence of real-time deployment validation.

Guan et al. (2024) [34] proposed ACS-IoT, a two-tier anomaly classification system for IoT networks combining Decision Tree for initial detection and CNN-BiLSTM for anomaly type classification. The approach uses SMOTE for class balancing and Particle Swarm Optimization (PSO) for feature selection. Evaluated on the IoTID20 and N-BaIoT datasets, it achieved up to 91.87 % accuracy, precision and recall near 90 %, and F1-score around 89 %. The novelty lies in cascading lightweight and deep models with optimized preprocessing. A limitation includes reliance on labeled data and high computational resources for CNN-BiLSTM, affecting real-time adaptability in constrained IoT settings.

Zhang et al. (2025) [35] proposed a hybrid intrusion detection model combining CNN, Bi-LSTM, and Transformer networks to handle spatial-temporal features in IoT traffic. Their system used CICIDS2017 and BoT-IoT datasets and integrated multi-stage feature selection via XGBoost and mutual information. While achieving high accuracy, the model lacks interpretability and does not address zero-day threats or similarity learning. Unlike SiamIDS, their work does not integrate SHAP explainability, contrastive training, or support cloud-native deployment.

Alabbadi an Bajaber (2025) [36] focuses on explainable AI for intrusion detection using DL models like DNN and CNN, complemented by SHAP and LIME for interpretability. Evaluated on TON_IoT, the models achieved high classification accuracy, and the SHAP visualizations improved analyst trust in IDS outputs. However, the approach does not include temporal sequence learning or contrastive similarity mechanisms. SiamIDS complements this by integrating SHAP with Bi-LSTM Siamese modeling, providing explainable and scalable detection of unknown attacks.

Alhayan et al. (2025) [37] proposed SHODLM-CEIDS, a hybrid deep learning model for intrusion detection in cloud computing, combining Dung Beetle Optimization (DBO) for feature selection, CNN-BiLSTM for classification, and Spotted Hyena Optimization (SHO) for tuning. Evaluated on NSL-KDD dataset (148,517 samples), it achieved 99.49 % accuracy, 94.49 % recall, 88.75 % precision, 91.24 % F1-score, and high specificity. The novelty lies in integrating biologically inspired optimizers with deep learning. Results showed robust detection across attack types. Limitations include potential inefficiency in tuning across scenarios and computational cost for high-dimensional data.

Duc et al. (2025) [38] proposed FedSAGE, a federated DGA malware detection system using Variational Autoencoder (VAE)-based unsupervised clustering and resource-aware client selection. The methodology includes latent space representation via pre-trained VAEs and client grouping using affinity propagation. Evaluated on a multi-zone DGA dataset with CNN, BiLSTM, and Transformer models, it achieved up to 89.83 % accuracy, 80.32 % F1-score, precision near 90 %, recall above 80 %, and strong specificity in unseen attack scenarios. Novelty lies in clustering clients without raw data or labels. Limitations include scaling affinity propagation and assuming client reliability, which may affect performance in large deployments.

Natha et al. (2025) [39] introduced the Composite Recurrent Bi-Attention (CRBA) model for spatiotemporal anomaly detection in video surveillance. Combining DenseNet201 for spatial feature extraction with BiLSTM networks and attention layers for temporal modeling, the methodology targets real-time detection of anomalies like accidents and theft. Evaluated on UCF Crime and Road Anomaly Dataset (RAD), the model achieved 92.2 % (RAD) and 86.2 % (UCF) accuracy, with F1-scores over 92 %, precision and recall exceeding 92 %, and specificity above 91 %. Limitations include high computational demands; novelty lies in integrating attention-driven BiLSTM with DenseNet to enhance spatiotemporal anomaly recognition.

Alsaleh et al. (2025) [40] proposed a semi-decentralized federated learning model for intrusion detection in heterogeneous IoT networks. The methodology clusters resource-constrained IoT clients, using BiLSTM, LSTM, and WGAN as lightweight local models. Trained on CICIoT2023, the BiLSTM model achieved 99.09 % accuracy, 68.05 % recall, 79.48 % precision, 70.45 % F1-score, and robust specificity.

Table 1
CIC IoT-DIAD 2024 dataset Traffic Distribution by Attack Category.

Traffic Category	Attack Family	Specific Attack Types	Number of Records
Benign	—	Normal IoT Traffic	398,330
Malicious	Brute Force	Dictionary Attack	3619
	Distributed DoS	ACK_Frag, ICMP_Flood, HTTP_Flood, ICMP_Frag	3478,814
	Denial of Service	SYN_Flood, HTTP_Flood, UDP_Flood	7901,855
	Mirai Variant	Mirai-greeth Flood	174,588
	Reconnaissance	Vulnerability Scan	442,158
	Spoofing	ARP Spoofing, DNS Spoofing	157,238
	Web-Based	SQL Injection	11,328

Novelty lies in clustering clients by model update similarity using autoencoder-processed weights and Manhattan-based K-means, enhancing FedAvg aggregation and reducing communication overhead. Limitations include underperformance on severely imbalanced classes and increased complexity in cluster formation, suggesting avenues for dynamic clustering optimization.

Mohale & Obagbuwa (2025) [41] developed an XAI-integrated ML-based IDS using Decision Trees, MLP, XGBoost, Random Forest, CatBoost, Logistic Regression, and Gaussian Naive Bayes. Tested on UNSW-NB15 (2.5 M records, 9 attack types), XGBoost and CatBoost achieved 87 % accuracy, 0.86–0.87 precision, 0.88 recall, 0.87 F1-score, and 0.94 ROC-AUC. The novelty lies in combining SHAP, LIME, and ELI5 for interpretable IDS decision-making. Limitations include dataset scope and challenges integrating XAI into resource-constrained environments. Results affirm improved transparency without compromising detection performance.

While recent advances in intrusion detection have achieved strong performance using deep learning, most existing methods continue to face several critical limitations that hinder their effectiveness in real-world cloud-IoT deployments. First, many models rely heavily on supervised learning and labeled datasets, making them ineffective against zero-day attacks or unseen threat patterns. Second, although Siamese architectures and few-shot models have been introduced, they often neglect temporal behavior modeling, which is crucial for capturing evolving patterns in IoT traffic. Another recurring issue is the lack of interpretability. Most state-of-the-art IDS solutions do not explain their decision-making process, making them impractical for SOC analysts who require transparency for trust and incident response. While some works have explored SHAP or LIME, these are usually decoupled from sequence-aware architectures or do not integrate similarity-based anomaly detection. Moreover, post-detection behavioral clustering, which can aid in triaging threats and identifying variants, is rarely incorporated into modern IDS pipelines. Additionally, cloud readiness and real-time scalability remain under-addressed. Many models exhibit high training accuracy but are not optimized for deployment in dynamic, resource-constrained environments like microservices or distributed SOCs.

To bridge these gaps, we propose SiamIDS—a unified, cloud-centric framework that incorporates:

- Autoencoder-based compression for dimensionality reduction,
- Bi-LSTM Siamese architecture for temporal similarity learning and zero-shot detection,
- SHAP explainability for transparent decision-making, and
- OPTICS clustering for post-detection threat grouping.

This holistic design not only improves detection accuracy but also provides behavioral insights and practical deployability, fulfilling both technical and operational requirements of next-generation IoT security systems.

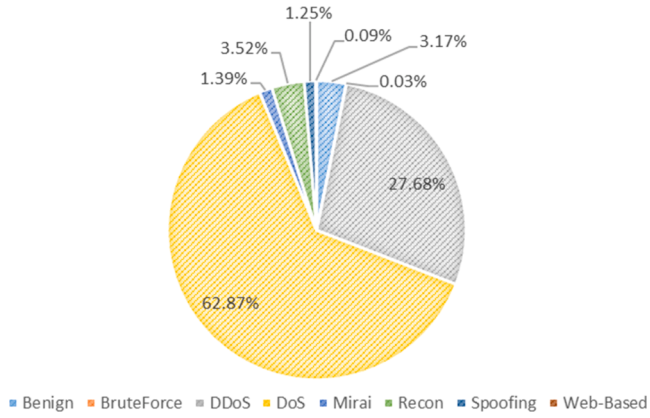


Fig. 3. CIC IoT-DIAD 2024 dataset Attack Category Distribution Percentage.

3. Materials and methods

3.1. Materials

3.1.1. CIC IoT-DIAD 2024 dataset

All experimental evaluations for SiamIDS are conducted using the CIC IoT-DIAD 2024 dataset [42], a comprehensive and recently released benchmark for IoT network intrusion detection. This dataset was chosen for its realistic representation of network behavior across diverse IoT devices under both benign and adversarial conditions, providing a challenging and practical testbed for intrusion diagnosis. As shown in Table 1, it includes flow-level records for 33 distinct attack types, grouped into 7 high-level attack families—DDoS, DoS, Spoofing, Mirai, Reconnaissance, Web-based intrusions, and Brute Force attacks. Each flow comprises 83 features, capturing a broad spectrum of traffic characteristics, including timestamps, protocol flags, packet and byte statistics, flow duration, and header information [43]. The dataset is provided in preprocessed CSV format with ground-truth labels for both binary classification (Benign vs. Attack) and multiclass classification (specific attack types). A notable challenge of the dataset is its class imbalance, with benign traffic constituting a smaller fraction of total flows, while certain attack types like UDP Flood or ACK Fragmentation dominate, and others like SQL Injection are underrepresented. This imbalance motivates the use of contrastive learning within the Siamese framework, which focuses on modeling behavioral similarity rather than relying on traditional class distributions. The dataset's richness and diversity make it suitable for evaluating SiamIDS under large-scale, imbalanced, and heterogeneous IoT traffic conditions.

Additionally, Fig. 3 presents the overall class distribution across major families, highlighting the dominance of DoS and DDoS traffic and the relatively minor presence of attacks such as Spoofing or Web-based intrusions. This data distribution profile poses a real-world challenge for intrusion detection models and serves as a robust foundation for evaluating SiamIDS under imbalanced, diverse, and large-scale conditions.

3.1.2. Data pre-processing

The proposed SiamIDS framework is trained and evaluated using the CIC IoT-DIAD 2024 dataset [42], which comprises high-dimensional IoT network traffic, including benign flows and 33 distinct attack types. To prepare the data for temporal similarity modeling and ensure learning efficiency, the following preprocessing steps are applied. First, feature scaling is performed using Z-score normalization [44], D_i defined as in Eq. (1):

$$D_i = \frac{(tD_i - \mu)}{\sigma} \tag{1}$$

where tD_i is the original traffic data, μ is the mean, and σ is the standard deviation. While Z-score assumes approximate normality and does not

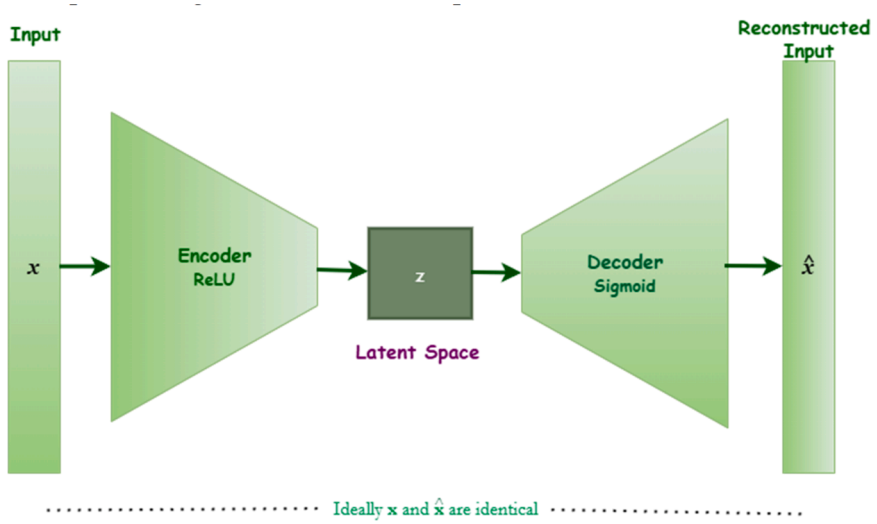


Fig. 4. Operational architecture of Shallow Autoencoder.

Table 2
Contrastive Pair Generation Statistics.

Pair Type	Description	Count
Positive Pairs	Unique benign–benign pairs from training split	100,000
Negative Pairs	Unique benign–attack pairs from training split	100,000
Total Training Pairs	For Siamese contrastive learning	200,000
Validation Pairs	50 % positive, 50 % negative from validation split	20,000
Reference Set	Benign flows used for similarity scoring at inference	10,000
Test Sequences	Unseen flows (Benign + Attack) from test split	~2.5 million

Table 3
Dataset Splits and Their Roles in Model Training, Validation, and Evaluation.

Dataset Split	Data Proportion / Size	Purpose / Usage
Training Set	70 % of benign and attack flows	Used for Autoencoder and Siamese training; initial OPTICS parameter calibration
Validation Set	10 % of benign and attack flows	Used to generate validation pairs and tune the similarity threshold
Test Set	20 % of mixed traffic flows	Reserved for final performance evaluation and clustering
Reference Set	10,000 benign flows (from training)	Excluded from training; used at test time for similarity comparison

explicitly model non-linear relationships, it effectively standardizes the feature space prior to neural network training. In SiamIDS, non-linear dependencies are subsequently captured by the autoencoder, making Z-score a lightweight and effective preprocessing choice. Z-score is favored over min–max or robust scaling because it recenters features around zero with unit variance, which is essential for LSTM-based models that are sensitive to feature scale across time steps [45,46]. This promotes gradient stability and uniform feature influence during sequence learning. Next, sequence slicing converts raw traffic flows into fixed-length windows (e.g., 10–20 packets), preserving temporal continuity. Finally, label conversion is applied: each sequence is labeled as *Benign* or *Malicious*, enabling binary contrastive learning in the Siamese network. This aligns with the framework's focus on modeling behavioral similarity rather than traditional multi-class classification.

3.1.3. Feature extraction

To improve efficiency, generalization, and training stability in the SiamIDS framework, a shallow Autoencoder (AE) is employed for dimensionality reduction [47]. As illustrated in Fig. 4, the Autoencoder module is a key component of the overall SiamIDS architecture, which integrates dimensionality reduction, Siamese Bi-LSTM-based detection, SHAP-based explainability, and OPTICS-based clustering. This unsupervised AE neural network is trained exclusively on benign traffic, allowing it to learn compressed latent representations that capture essential, noise-free behavioral features from high-dimensional IoT traffic data.

3.1.4. Pair generation strategy

To support contrastive learning in SiamIDS, we construct pairs of network flow sequences that reflect behavioral similarity or dissimilarity. A stratified contrastive sampling approach is adopted to ensure diversity and prevent overlap across training, validation, and reference sets [48]. Positive Pairs are built from randomly selected benign flows and represent behaviorally similar sequences. Negative Pairs consist of benign and malicious sequences, highlighting dissimilar patterns in flow dynamics. Validation Pairs are sampled independently for threshold tuning and ROC analysis and a reference set of benign flows is held out exclusively for similarity comparison during inference. The overall pair composition and dataset usage are detailed in Table 2. This setup ensures balanced training, avoids information leakage, and allows the Siamese model to generalize to diverse and unseen attacks.

3.1.5. Training and testing splits

To ensure robust and leakage-free evaluation, the CIC IoT-DIAD 2024 dataset is partitioned into stratified training, validation, and testing subsets. Stratification preserves the distribution of benign and attack flows across splits, ensuring balanced representation of all classes. A reference set of benign flows is held out exclusively for test-time similarity scoring in the Siamese network, preventing overlap with training data and enabling unbiased anomaly assessment. For contrastive learning, unique positive (Benign–Benign) and negative (Benign–Attack) pairs are generated using a stratified sampling strategy, as detailed in Section 3.1.4. Training pairs are used to teach the Siamese network robust behavioral embeddings, validation pairs support threshold tuning and ROC evaluation, and the reference set is employed solely during inference to compute similarity scores. This partitioning strategy enhances generalization to unseen attack types, mitigates overfitting, and aligns with SiamIDS's emphasis on behavioral similarity-based intrusion detection (see Table 3 for dataset splits and their roles).

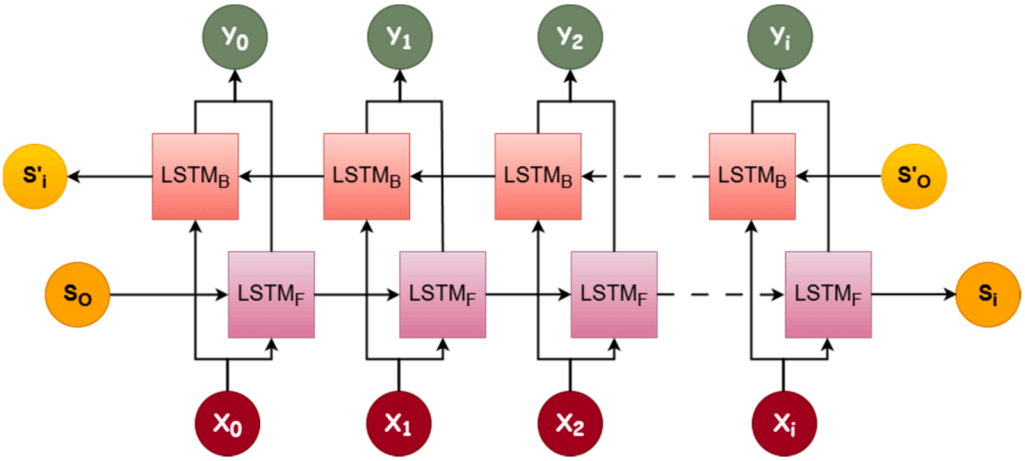


Fig. 5. Architecture of the Bi-LSTM layers in SiamIDS framework.

3.2. Methods

3.2.1. Autoencoder-based feature compression for IoT intrusion detection

Autoencoders are unsupervised neural networks that learn compressed representations of input data by reconstructing it with minimal error. In IoT intrusion detection, they efficiently reduce feature dimensionality while preserving critical behavioral patterns of network traffic [49,50] (Fig. 4).

An autoencoder comprises an encoder that maps input $x \in R^n$ to a lower-dimensional latent space $z \in R^m (m < n)$ via a non-linear transformation f as defined in Eq. (2), and a decoder g that reconstructs $x$from z as defined in Eq. (3). Training minimizes reconstruction loss, typically Mean Squared Error (MSE):

$$z = f(x) = \sigma(W_e x + b_e), \tag{2}$$

$$\hat{x} = g(z) = \sigma(W_d z + b_d) \tag{3}$$

where $W$and b denote weights and biases, and σis the activation function (ReLU/Sigmoid). In the SiamIDS framework, the autoencoder compresses inputs before feeding them into the Siamese Bi-LSTM, enhancing computational efficiency and filtering noise while preserving flow characteristics. It is trained exclusively on benign traffic to model normal behavior; significant reconstruction errors indicate anomalies. The employed architecture features shallow fully connected encoder-decoder layers with a 20-neuron bottleneck, empirically optimized to balance reconstruction accuracy and compactness. This setup ensures effective dimensionality reduction without compromising the ability to discriminate anomalous traffic, forming a robust foundation for subsequent temporal and similarity-based analysis.

3.2.2. Bi-LSTM-based temporal modeling of network traffic

Bidirectional Long Short-Term Memory (Bi-LSTM) networks extend Recurrent Neural Networks (RNNs) by processing sequential data in both forward and backward directions, thereby capturing contextual information from past and future time steps. In intrusion detection, where network traffic exhibits temporal dependencies, Bi-LSTM effectively models evolving flow behaviors. An LSTM unit maintains a cell state C_t governed by three gates—input (i_t), forget (f_t), and output (o_t)—as defined in Eqs. (4–9). These mechanisms enable selective retention and updating of information over time. Unlike conventional LSTMs, Bi-LSTM concatenates hidden states from both directions $[h_t^{\rightarrow}; h_t^{\leftarrow}]$, allowing comprehensive temporal representation of traffic sessions. The internal architecture of the Bi-LSTM layers used in the SiamIDS framework is illustrated in Fig. 5.

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \tag{4}$$

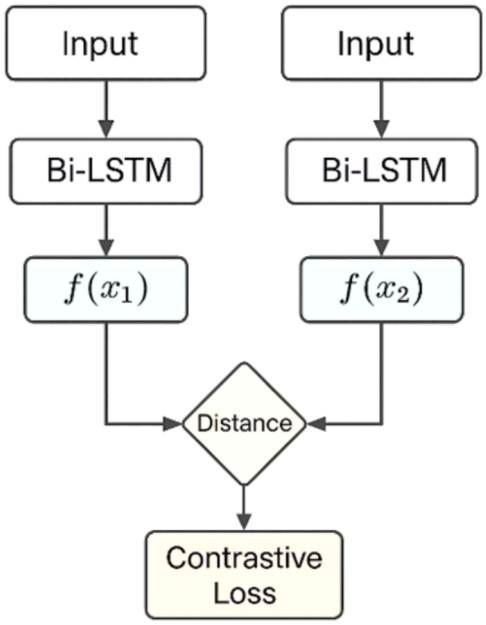


Fig. 6. Siamese Network Similarity Learning.

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \tag{5}$$

$$\overline{C}_t = tanh(W_C * [h_{t-1}, x_t] + b_{Ct}) \tag{6}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \overline{C}_t \tag{7}$$

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{8}$$

$$h_t = o_t \odot \tanh(C_t) \tag{9}$$

Within the SiamIDS framework, Bi-LSTM constitutes the core of the twin subnetworks, generating time-aware, flow-sensitive embeddings for each input instance. These embeddings are leveraged to compute similarity scores during contrastive training and inference. The implemented Bi-LSTM employs two LSTM layers per direction with 64 hidden units, integrated with dropout and batch normalization for regularization and stability. By capturing bidirectional and long-range dependencies, Bi-LSTM enhances the framework's ability to discern subtle temporal deviations, significantly improving zero-day attack diagnosis accuracy.

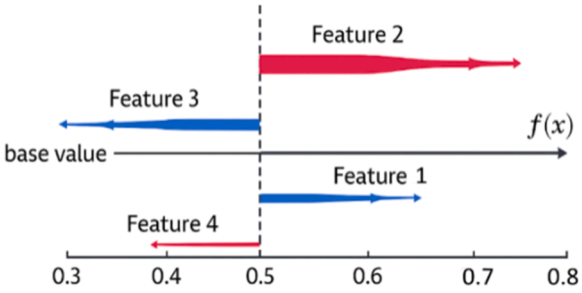


Fig. 7. SHAP Force Plot Illustrating Feature Contributions.

3.2.3. *Siamese network for similarity-based anomaly detection*

A Siamese Neural Network employs dual, weight-shared subnetworks that learn a discriminative similarity metric between paired inputs through their latent feature representations. In intrusion diagnosis, this design effectively differentiates benign and malicious traffic, particularly under limited or imbalanced labeled data conditions [51, 52]. Each branch receives distinct inputs x_1 and x_2, generating embeddings $f(x_1)$ and $f(x_2)$. The similarity is measured using the Euclidean distance, as defined in Eq. (10):

$$D(x_1, x_2) = ||f(x_1) - f(x_2)||_2 \tag{10}$$

Learning is governed by the contrastive loss function, presented in Eq. (11):

$$L = (1-y)\frac{1}{2}D^2 + y\frac{1}{2}\max(0, m-D)^2 \tag{11}$$

where $y \in \{0, 1\}$denotes pair similarity and $m$defines the margin for dissimilar samples.

As shown in Fig. 6, the SiamIDS framework trains on both intra-class (similar) and inter-class (dissimilar) traffic pairs to model behavioral proximity. During inference, each traffic instance is compared against benign references; instances exceeding a learned threshold are marked anomalous. The similarity-driven paradigm enables zero-day threat identification, minimizes dependence on predefined class boundaries, and enhances scalability. Combined with Bi-LSTM-based temporal encoding, the Siamese configuration reinforces contextual discrimination and interpretability within complex IoT–cloud environments.

3.2.4. *SHAP for feature-level explainability in intrusion detection*

Interpretability is a critical requirement in cybersecurity applications, particularly for deep learning models deployed in sensitive or mission-critical environments. To overcome the "black-box" limitation of architectures such as Bi-LSTM and Siamese networks, the SHapley Additive exPlanations (SHAP) framework is integrated into the SiamIDS model to provide transparent, feature-level interpretability.

SHAP is a game-theoretic approach that assigns each input feature a contribution score (Shapley value) toward the model's prediction [36, 41]. The Shapley value for feature i is defined in Eq. (12):

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!}[f(S \cup \{i\}) - f(S)] \tag{12}$$

where F represents the full feature set, S is any subset excluding i, and $f(S)$ is the model output using only features in S. This formulation evaluates a feature's marginal contribution across all possible feature combinations. Within SiamIDS, SHAP is applied post-inference to interpret anomaly predictions generated by the Siamese module. Once a traffic flow is flagged as malicious, SHAP computes per-feature importance scores, revealing which attributes influenced the anomaly score most strongly. As shown in Fig. 7, SHAP visualizations such as force plots enable both local and global interpretation of detection outcomes.

Integrating SHAP enhances model transparency, supports validation and debugging, and fosters trust by aligning SiamIDS with the broader principles of explainable artificial intelligence (XAI) in IoT–cloud intrusion diagnosis.

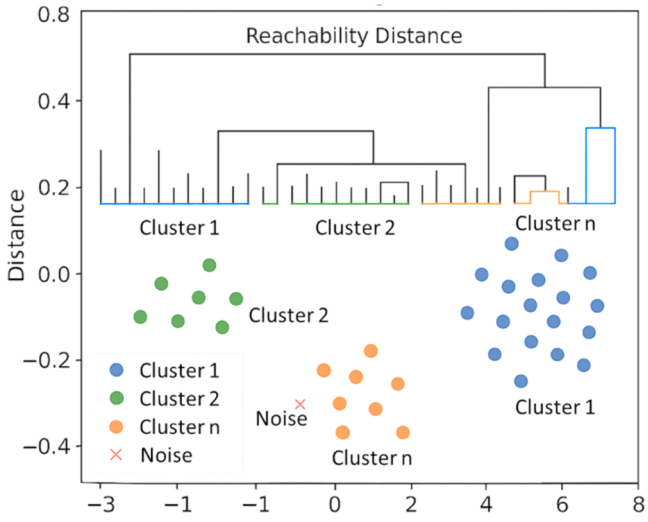


Fig. 8. OPTICS Clustering of Anomalies.

3.2.5. *OPTICS for density-based clustering of anomalous behaviors*

Beyond detecting intrusions, grouping anomalies into coherent behavioral clusters is essential for root cause analysis and threat profiling. To address this, the SiamIDS framework employs OPTICS (Ordering Points To Identify the Clustering Structure) for post-detection clustering of anomalous traffic. OPTICS is a density-based algorithm that extends DBSCAN by identifying clusters of varying densities without requiring a predefined cluster count. It introduces two key metrics—core distance and reachability distance—to reveal hierarchical data structures. The reachability distance between two points is defined in equation (13) as:

$$\text{Reachability} - \text{dist}(p, o) = \max(\text{core} - \text{dist}(o),\ \text{dist}(p, o)) \tag{13}$$

where core-dist(o)is the minimum radius ε containing at least *MinPts* neighbors.

In SiamIDS, anomalous flows detected by the Siamese Bi-LSTM module are passed to OPTICS for clustering. This enables behavioral grouping, where related attack variants—such as multiple DDoS or botnet types—are organized into semantically meaningful clusters. As shown in Fig. 8, the resulting reachability plots and 2D projections reveal the underlying structure of anomalous behaviors.

OPTICS provides several advantages: it eliminates the need to specify the number of clusters, effectively detects non-convex and variable-density formations, and exhibits strong resilience to noise. Its integration enhances post-detection analytics, enabling Security Operations Centers (SOCs) to interpret, correlate, and prioritize anomalies efficiently—thereby supporting dynamic threat intelligence and adaptive response in complex IoT–cloud ecosystems.

4. Proposed methodology: SiamIDS for interpretable IoT intrusion detection

This section details the internal design, operational workflow, and implementation components of SiamIDS—a novel intrusion detection system engineered for interpretability, zero-day detection, and scalable deployment in IoT-cloud ecosystems. The methodology addresses several pressing challenges in modern IDS—namely, detection of zero-day attacks, model explainability, low-resource deployment, and post-detection behavioral analysis. SiamIDS integrates five core modules: an autoencoder for dimensionality reduction, a Bi-LSTM backbone for

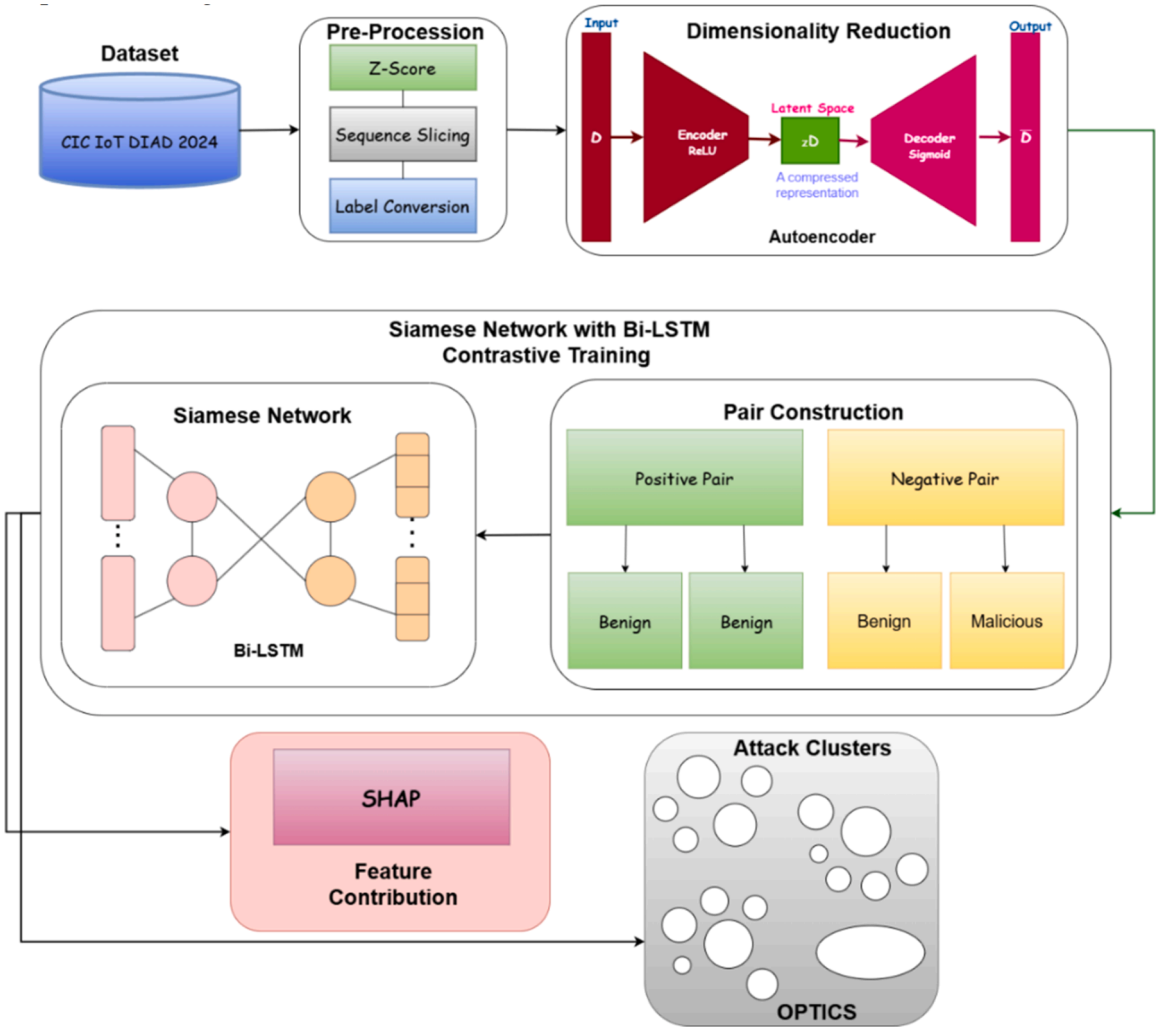


Fig. 9. Architectural overview of SiamIDS integrating autoencoder, Bi-LSTM Siamese network, SHAP-based explanation, and OPTICS clustering.

temporal modeling, a Siamese network for contrastive similarity learning, SHAP for explainability, and OPTICS for clustering of detected anomalies. Each component plays a crucial role in enabling the system to accurately and transparently detect malicious behavior.

4.1. System model

The SiamIDS framework operates through a structured sequence of processes encompassing dimensionality reduction, temporal embedding, similarity learning, interpretable decision-making, and post-detection clustering. Initially, a shallow autoencoder is trained exclusively on benign traffic to compress high-dimensional network vectors Dinto a compact latent representation $\widehat{Z}_D$. The encoder and decoder functions are defined in Eqs. (14) and (15), respectively:

$$\widehat{Z}_D = E_\theta(D) = \sigma(W_e D + b_e), \tag{14}$$

$$\widehat{D} = g_\theta(\widehat{Z}_D) = \sigma(W_d \widehat{Z}_D + b_d) \tag{15}$$

where W_e,$W_d$and b_e,$b_d$are trainable parameters, and σ is the activation function (ReLU for encoder, Sigmoid for decoder). The network is trained to minimize the mean squared error (MSE) between original and reconstructed inputs as defined in Eq. (16):

$$\mathrm{MSE}_{\mathrm{loss}} = \frac{1}{n}\sum_{i=1}^{n} \left| D_i - \widehat{D}_i \right|^2 \tag{16}$$

Training proceeds until the convergence threshold T is satisfied. The reduced-dimensional sequence $\widehat{Z}_D$is then passed through a Bi-LSTM to capture temporal dependencies. The hidden state at time t is computed as in Eq. (17):

$$h_t = \overrightarrow{h_t} \| \overleftarrow{h_t} \tag{17}$$

and aggregated via average pooling to form a global sequence embedding e. To distinguish benign from malicious traffic, SiamIDS employs a Siamese architecture with contrastive learning. Given paired embeddings e_1,e_2, the Euclidean distance $d(e_1,e_2) = |e_1 - e_2|^2$is minimized for similar pairs and maximized for dissimilar pairs using the contrastive loss is defined in Eq. (18):

$$L_{\mathrm{con}} = y\, d^2 + (1-y)\max\,(0, m-d)^2 \tag{18}$$

where $y \in \{0,1\}$indicates pair similarity, and $m$enforces separation between dissimilar samples. During inference, a test sequence $D_{\mathrm{test}}$is encoded into $e_{\mathrm{test}}$and compared to reference benign embeddings E_{ref}. The mean distance defines an anomaly score, and sequences exceeding threshold τare flagged as anomalous. To ensure interpretability, SHAP computes feature-level contributions for each prediction as per Eq. (19):

$$f(x) = \phi_0 + \sum_{i=1}^{n} \phi_i \tag{19}$$

where $\phi_0$is the expected model output and $\phi_i$quantifies the contribution

Algorithm 1
SiamIDS Working Flow.

Input: Network traffic sequences D
1. Normalize features using Z-score.
2. Encode with autoencoder: Z_D = E(D)
3. Construct pair set:
→ Positive: (B1, B2), label y=1
→ Negative: (B1, A), label y=0
4. For each pair:
→ Compute embeddings (e1, e2)
→ Compute distance: d = ||e1 - e2||²
→ Compute L_con and update model
5. During inference:
→ Encode test: e_test
→ Compare to E_ref
→ Compute anomaly score
→ Apply SHAP to explain decisions
→ Cluster anomalies using OPTICS

of feature *i*. DeepExplainer is employed to provide human-understandable insights into feature influences. Finally, detected anomalies $E_{\text{anom}} = \{e_1, e_2, \ldots, e_n\}$ are analyzed with OPTICS clustering for behavioral grouping. Core and reachability distances are computed as in Eq. (20) and (21):

$$\text{core}(p) = \text{distance to minPts} - \text{th neighbor}, \tag{20}$$

$$\text{reachability}(o,p) = \max\left(\text{core}(p), \text{distance}(p,o)\right) \tag{21}$$

The resulting reachability plot reveals dense clusters and sparse outliers, supporting SOC analysts in profiling attack families. Collectively, these formulations Eqs. (14–21) define SiamIDS's learning objectives, similarity metrics, decision thresholds, interpretability logic, and clustering strategies, enabling robust, scalable, and explainable intrusion detection in complex IoT–cloud environments.

4.2. Architecture and working of SiamIDS

This section introduces SiamIDS, a cloud-compatible intrusion detection framework developed for scalable and interpretable anomaly detection in IoT environments. As depicted in Fig. 9, the framework begins with a data preprocessing stage that includes Z-score-based feature scaling, fixed-length sequence slicing, and label transformation.

The processed data is then passed into a shallow autoencoder, trained exclusively on benign traffic, to generate low-dimensional latent representations. These embeddings capture core behavioral patterns while reducing computational overhead.

To enable contrastive learning, SiamIDS constructs input pairs—positive pairs (Benign–Benign) and negative pairs (Benign–Malicious)—which are then fed into a Siamese network consisting of two identical Bi-LSTM branches. Each branch encodes the temporal dependencies in the

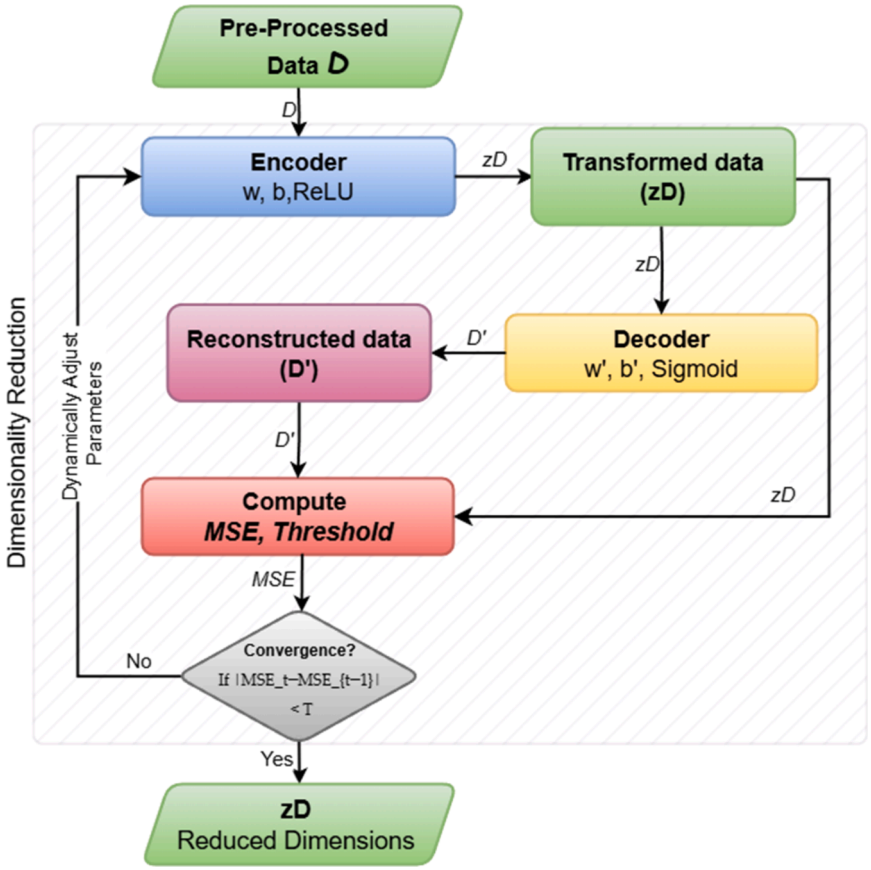


Fig. 10. Process flow of the shallow Autoencoder used for dimensionality reduction in the SiamIDS framework.

respective input sequences, and the network outputs a similarity score that quantifies behavioral similarity.

During inference, each test sequence is compared against a reference pool of benign embeddings to determine whether it is anomalous. For model transparency, the system integrates a SHAP-based explainability layer, which highlights the contribution of each feature toward the model's decision.

Finally, the anomalous outputs are subjected to post-detection clustering using **OPTICS**, a density-based algorithm that organizes similar anomalies into behavioral clusters while identifying outliers. This supports real-time triaging and semantic profiling of novel or zero-day threats in large-scale IoT deployments. The step-by-step operational flow of SiamIDS is detailed in Algorithm 1.

4.3. Autoencoder architecture with latent space design and bottleneck configuration

The Autoencoder consists of two parts: an encoder E_θ and a decoder D_θ. The overall process of the shallow Autoencoder used in SiamIDS is depicted in Fig. 10, where the input data is encoded into a compressed latent space and then reconstructed to minimize the reconstruction error. The encoder maps the input vector D into a lower-dimensional latent space ${}_Z\mathrm{D}$ as in Eq. (14). The decoder then reconstructs the input as in Eq. (15). Where, the ReLU activation function is used in the encoder, while the decoder employs the Sigmoid activation function, denoted as σ. The network is trained to minimize the mean squared error (MSE) between the input D and the reconstructed output $\widehat{D}$, the MSE loss defined as in Eq. (16). A convergence threshold T is dynamically monitored to determine training stability. When $|\mathrm{MSE_t}-\mathrm{MSE_{t-1}}| < \mathrm{T}$, the training stops and the encoder is used for feature compression.

The latent dimension (bottleneck size) is a critical hyperparameter. We empirically evaluate various latent sizes (10 to 40) and select 20 as optimal. This choice is based on achieving minimal reconstruction loss without sacrificing temporal variance or interpretability. Smaller sizes (e.g., 10 or 15) result in underfitting and information loss, while larger ones (e.g., 35 or 40) offer negligible accuracy gain but higher complexity. The chosen bottleneck layer significantly reduces the input size for the Siamese Bi-LSTM, enhancing computational efficiency and convergence speed.

Unlike traditional dimensionality reduction techniques such as Principal Component Analysis (PCA) or Information Gain, which assume linear separability or rely on predefined feature importance scores, the Autoencoder offers a more adaptive and data-driven alternative [53,54]. It is capable of capturing non-linear dependencies between features, which are especially common in complex IoT traffic. Moreover, instead of relying on generic variance-based projections like PCA, the Autoencoder learns task-specific embeddings that are optimized for downstream objectives—such as temporal similarity learning in the Siamese network. This enables the model to retain semantically meaningful patterns critical for distinguishing subtle behavioral anomalies. Another key advantage is that the Autoencoder avoids manual feature engineering or domain assumptions, allowing the model to generalize across diverse traffic sources and attack types [55]. While PCA projects data into orthogonal components derived from eigenvectors—often without regard to task relevance [56]—Autoencoders learn to reconstruct input patterns, preserving latent structures that are most informative for reconstruction error minimization and anomaly detection. This makes Autoencoders particularly suitable for dynamic, evolving network environments, where handcrafted or static feature selection methods may fall short. Once convergence is achieved (see flowchart), the transformed vectors ${}_Z\mathrm{D}$ from the encoder constitute the reduced-dimensional input to the Siamese network in detection phase. This modular separation enhances interpretability and enables easy plug-and-play with different detection models.

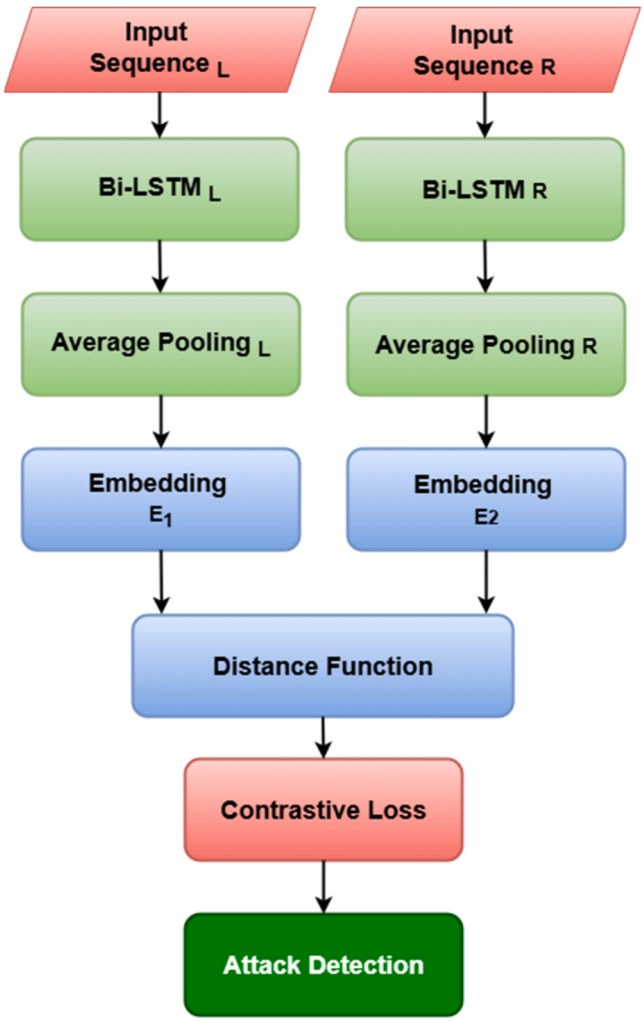


Fig. 11. Architecture of the Siamese Bi-LSTM network for attack detection in the SiamIDS framework.

4.4. Siamese network with Bi-LSTM backbone

At the core of the proposed SiamIDS framework is a Siamese neural network composed of two identical sub-networks, each built upon Bi-directional Long Short-Term Memory (Bi-LSTM) layers. This design enables the system to assess behavioral similarity between two network traffic sequences, making it ideal for detecting previously unseen (zero-day) or obfuscated threats through contrastive learning rather than traditional classification [20,57]. As shown in Fig. 11, the Siamese network architecture processes the input sequences through two identical Bi-LSTM branches. Each Siamese branch processes a flow sequence of reduced-dimensional input (from the Autoencoder) and maps it to a latent embedding space. The Bi-LSTM architecture captures sequential dependencies in both forward and backward directions, allowing the model to learn packet timing patterns, transition structures, and burst behaviors commonly present in IoT traffic [58]. The input sequence $\mathrm{D}=\{\mathrm{D_1},\mathrm{D_2},\ldots,\mathrm{D_T}\}$, where each $\mathrm{D_t} \in {}_Z\mathrm{D}$ is a feature vector for a packet at time step t, and T is the sequence length. The Bi-LSTM produces forward and backward hidden states $\overrightarrow{h_t}$, $\overleftarrow{h_t}$ and concatenates them as h_t, as defined in Eq. (17).

The final output embedding e is typically derived from average pooling of the Bi-LSTM. Both branches share weights (i.e., $\theta_{\mathrm{left}}=\theta_{\mathrm{right}}$), ensuring symmetric encoding and allowing the network to focus on *relative* sequence similarity rather than absolute classification. The embedding generation process is outlined in Algorithm 2, which

Algorithm 2
Embedding Generation via Siamese Bi-LSTM.

```
Define Siamese_BiLSTM_Encoder(θ): Bi-directional LSTM layers with shared weights
  For each input sequence D = {D₁, D₂, …, D_T}:
    Reduce dimensionality: D' = AE.encode(D)
    Compute Bi-LSTM embedding:
      For t = 1 to T:
        h→_t = LSTM_forward(D'_t), h←_t = LSTM_backward(D'_t)
        h_t = [h→_t || h←_t]
      end
    Return e = AveragePool({h₁, h₂, …, h_T})
  end
```

Algorithm 3
Pair construction and contrastive loss calculation.

```
PositivePairs ← RandomPairs(Benign, Benign)
  NegativePairs ← RandomPairs(Benign, Attack)
  TrainPairs ← PositivePairs ∪ NegativePairs
  For each pair (D₁, D₂) in TrainPairs with label y ∈ {1, 0}:
    e₁ = Siamese_BiLSTM_Encoder(D₁)
    e₂ = Siamese_BiLSTM_Encoder(D₂)
    Compute distance: d = ||e₁ - e₂||₂
    Compute contrastive loss:
      L = y * d² + (1 - y) * max(0, m - d)²
    Update weights θ using gradient descent
  end
```

describes how each input sequence is processed through the Bi-LSTM layers to produce the final embedding.

4.4.1. Pair construction for contrastive training

The Siamese network is trained using a contrastive learning paradigm. Instead of training the model to classify a sequence, we present it with pairs of sequences, each labeled based on their similarity:

- Positive pairs: Two benign sequences (Benign–Benign) that are expected to produce high similarity.
- Negative pairs: One benign and one malicious sequence (Benign–Malicious), which should exhibit low similarity.

(D_1, D_2) is a sequence pair, and y ∈ {0,1} the label indicating similarity (1 for similar, 0 for dissimilar). The embeddings e_1=f(D_1), e_2=f(D_2) are passed through a distance function d, such as Euclidean distance. The contrastive loss function, L_{con} is then defined as in Eq. (18). This formulation ensures that embeddings of similar pairs are pulled closer, while dissimilar pairs are pushed apart beyond the margin. In our setup, m is empirically set to 1.0, based on convergence behavior and validation performance. To avoid class imbalance, the pair generation is carefully balanced with equal proportions of positive and negative pairs. Malicious samples are randomly sampled from all attack categories, ensuring representation across different threat behaviors. The process for constructing these pairs, as well as computing the contrastive loss and updating the model's weights, is described in Algorithm 3.

4.4.2. Detection logic during inference

During inference, each unlabeled test sequence is passed through the trained Siamese model and compared against a reference pool of benign embeddings derived from clean validation data. For a test embedding e_{test}, its similarity to each reference $e_r \in D$ is computed using a distance function. The average distance across all comparisons is used as the anomaly score. If this score falls below a pre-defined threshold τ, the sequence is classified as anomalous:

$$Label = \begin{cases} Anomalous, & if \min(e_{test}, e_r) < \tau \\ Benign, & otherwies \end{cases}$$

The threshold τ is determined using Receiver Operating Characteristic (ROC) analysis on a held-out validation set to optimize sensitivity and specificity. To ensure real-time capability in large-scale deployments, embedding indexing using FAISS (Facebook AI Similarity Search) is employed. This enables fast retrieval of the most similar benign embeddings without exhaustive pairwise computation [59]. The process of generating reference embeddings and computing anomaly scores is outlined in Algorithm 4.

4.5. Explainability integration with SHAP for feature-level interpretation

One of the key challenges in deploying deep learning-based intrusion detection systems (IDS) in operational environments is the lack of interpretability. Security analysts often require clear, feature-level explanations for why a traffic instance is flagged as anomalous, especially in high-stakes environments like SOCs (Security Operation Centers). To

Algorithm 4
Generation of reference embeddings and anomaly score computation.

```
E_ref = {Siamese_BiLSTM_Encoder(D_r) | D_r ∈ clean validation set}
  For each test sequence D_test ∈ Dtest:
    e_test = Siamese_BiLSTM_Encoder(D_test)
    Compute distance set: S = {||e_test - e_r||₂ | e_r ∈ E_ref}
    AnomalyScore = mean(S)
    if AnomalyScore ≥ τ:
      Label ← Anomalous
    else:
      Label ← Benign
    end
  end
```

Algorithm 5
Explainability Layer using SHAP.

```
Encode test sequence using Siamese network:
  e_test ← f_left(D_test)
  Compute similarity score:
  s ← similarity(e_test, e_ref)
  Initialize SHAP Explainer:
  explainer ← DeepExplainer(f_left, background_data)
  Compute SHAP values for test input:
  SHAP_values ← explainer.shap_values(D_test)
  Interpret output:
  For each feature i in D_test:
  ϕ_i ← SHAP_values[i]
  Return explanation vector {ϕ₁, ϕ₂, …, ϕ_n}
```

address this, the SiamIDS framework integrates a SHapley Additive exPlanations (SHAP) layer, enabling feature-level interpretability for similarity-based decisions made by the Siamese network. SHAP is a game-theoretic approach to explaining the output of machine learning models by computing the contribution of each input feature toward the model's prediction. It is based on the concept of Shapley values from cooperative game theory, which assigns a fair value to each player (feature) based on their contribution to the final outcome [41,60].

Given a model f and input $D \in {}_{D}Z$, SHAP aims to express the model's prediction as in Eq. (22).

$$f(D) = \phi_0 + \sum_{i=0}^{n} \phi_i \tag{22}$$

where ϕ_0 is the model's expected output and ϕ_i represents the Shapley value or contribution of feature i. In the context of SiamIDS, SHAP is applied to the left branch of the Siamese network to explain why a test sequence is similar or dissimilar to a reference benign sequence.

Although SHAP is traditionally designed for explaining classification or regression outputs, it is adapted in SiamIDS to interpret similarity scores produced by the Siamese network. Specifically, SHAP is applied to the left branch of the Siamese architecture, which receives the test sequence and encodes it into a latent embedding e_{test} . This embedding is then compared to a reference benign embedding e_{ref}, and the similarity (or distance) between the two determines whether the test sequence is considered anomalous. To explain this similarity decision, a SHAP explainer—DeepExplainer—is initialized to compute the contribution of each input feature toward the final similarity score. A high positive SHAP value indicates that a feature increases dissimilarity (supports anomaly), while a negative value suggests alignment with benign behavior. The step-by-step procedure for SHAP-based interpretation within SiamIDS is detailed in Algorithm 5, including encoding the input, computing similarity, initializing the explainer, and generating feature-level SHAP values. This produces a ranked explanation vector indicating the most influential features responsible for the anomaly classification.

The integration of SHAP into SiamIDS provides several practical benefits that enhance both operational utility and trust in the detection process. First, SHAP explanations offer valuable analyst insight by highlighting which protocol fields or flow-level features—such as *Flow Duration, Packet Length Variance*, or *TCP Flag PSH*—contributed most significantly to a sequence being flagged as anomalous. This granular feedback helps analysts quickly understand behavioral deviations from benign patterns. Second, the model's explainability fosters trust and transparency, which is particularly important in high-assurance domains where AI-assisted decisions must be auditable and compliant with regulatory standards. Third, SHAP enables detailed root-cause analysis, helping determine whether anomalies are driven by unusual timing patterns, abnormal port behavior, or traffic volume inconsistencies. Lastly, SHAP can be used for model debugging, offering visibility into whether the Siamese network is overfitting to irrelevant features or overlooking critical ones. This makes SHAP a powerful component not only for improving incident response but also for refining model robustness during development and retraining phases.

4.6. Behavioral clustering of anomalies using optics

While the Siamese Bi-LSTM architecture effectively detects anomalous sequences by measuring their dissimilarity from known benign behavior, the detection output alone is insufficient for understanding the structure of emerging or zero-day threats. To enhance post-detection analysis, the SiamIDS framework incorporates a lightweight clustering layer using OPTICS (Ordering Points To Identify the Clustering Structure). This component allows the system to group behaviorally similar anomalies and uncover hidden attack families, improving threat visibility and aiding security analysts in response planning.

OPTICS is a density-based clustering algorithm that extends DBSCAN

Algorithm 6
OPTICS-Based Clustering of Anomalous Embeddings in SiamIDS.

```
Set OPTICS parameters:
  min_samples ← 10
  xi ← 0.05
  Initialize OPTICS model:
  optics_model ← OPTICS(min_samples, xi, metric='euclidean')
  Fit model on anomalous embeddings:
  optics_model.fit(E_anom)
  Extract reachability plot and cluster structure:
  reachability ← optics_model.reachability_
  ordering ← optics_model.ordering_
  labels ← optics_model.labels_
  Post-process labels:
  For each embedding e_i in E_anom:
  If labels[i] == -1:
  Mark as noise
  Else:
  Assign to cluster C_j
  Return cluster labels and noise point indices
```

Table 4
Experimental Environment Setup.

Component	Configuration
Platform	Google Colab Pro
OS Environment	Linux-based Virtual Machine
CPU	2.3 GHz Intel Xeon (virtualized)
RAM	16 GB
GPU	NVIDIA Tesla T4
Python Version	3.10
Major Libraries	TensorFlow 2.13, Keras, scikit-learn, SHAP, FAISS, OPTICS
Runtime Type	GPU-enabled (CUDA-supported)

by removing the requirement of a fixed global density threshold. Instead of forcing a predefined number of clusters, OPTICS generates a reachability plot that reveals variable-density clusters and outlier points (noise) without relying on user-specified k values or epsilon parameters. This makes it ideal for unsupervised threat categorization in cybersecurity, where attack behaviors can vary in structure, intensity, and frequency. Unlike k-means or hierarchical clustering, which assume convex or hierarchical cluster shapes, OPTICS adapts naturally to irregular or elongated cluster boundaries, which are common in network traffic data [61,62].

Once the Siamese model flags a sequence as anomalous, its corresponding latent embedding $e_{test} \in_z D^k$ is preserved for further analysis. The collection of all such anomalous embeddings, denoted as $E_{anom}=\{e_1, e_2,\ldots,e_n\}$, is then passed to the OPTICS algorithm for unsupervised clustering. OPTICS operates by computing core distances and reachability distances to build a reachability plot that reveals the hierarchical density-based structure in the data. Unlike DBSCAN or k-means, OPTICS does not require a fixed number of clusters or a neighborhood radius, but instead relies on two key parameters: min_samples (minimum points to form a dense region) and xi (minimum steepness to detect cluster boundaries). In SiamIDS, we set min_samples = 10 and xi = 0.05 to allow flexible and fine-grained clustering. The detailed procedure for applying OPTICS to the SiamIDS anomaly embeddings is presented in Algorithm 6, including parameter initialization, model fitting, cluster label extraction, and noise identification. These clusters, along with the detected noise points, form the basis for post-detection threat interpretation, allowing analysts to profile attack behaviors and prioritize investigation.

5. Experimentation, results and analysis

5.1. Experimental setup

The SiamIDS framework was implemented using Python 3.10, leveraging core libraries including TensorFlow 2.13, Keras 2.13, scikit-learn 1.3.2, SHAP 0.41.0, FAISS 1.7.4, and OPTICS 0.9.0. All experiments were conducted on Google Colab Pro, running a Linux-based virtual machine configured with 2 virtual CPU cores (2.3 GHz Intel Xeon), 16 GB RAM, and an NVIDIA Tesla T4 GPU with 16 GB memory. GPU acceleration (CUDA 12.1 and cuDNN 8.9) was used for both model training and inference to ensure efficient computation. The complete experimental environment, including hardware, runtime configuration, and major software components, is detailed in Table 4.

5.2. Hyperparameters and model configuration

The architecture of SiamIDS comprises four primary components: a shallow Autoencoder, a Siamese Bi-LSTM for temporal similarity modeling, SHAP for interpretability, and OPTICS for clustering of anomalous flows. Each component's hyperparameters were determined through iterative empirical validation to optimize performance, generalizability, and stability. For the Autoencoder, the latent size, batch size, and training epochs were tuned to balance dimensionality reduction with accurate reconstruction, ensuring essential traffic patterns are preserved while avoiding overfitting. The Siamese Bi-LSTM, including hidden units, embedding size, contrastive margin, and learning rate, was calibrated to maximize temporal feature representation and inter-class separation while maintaining stable convergence. OPTICS parameters, such as min_samples and xi, were selected to produce meaningful clusters of anomalous flows, effectively distinguishing dense attack groups from sparse outliers. SHAP's DeepExplainer was used to provide interpretable, feature-level insights post-inference. This hyperparameter selection process was guided by performance metrics including reconstruction error, clustering quality, and detection effectiveness on the validation set. The finalized hyperparameters reflect empirically validated settings that enable robust, scalable, and interpretable intrusion detection within complex IoT–cloud environments. Table 5 summarizes these configurations for all SiamIDS modules.

Table 5
Model Hyperparameters and Configurations.

Component	Parameter	Value / Description
Autoencoder	Latent Size	20 (compressed feature dimension)
	Activation	ReLU
	Loss	Mean Squared Error (MSE)
	Optimizer	Adam
	Learning rate	0.001
	Epochs	39
	Batch Size	512
Siamese Model	Bi-LSTM Units	64 units (per direction)
	Embedding Size	128
	Loss Function	Contrastive Loss
	Margin	1.0
	Epochs	30
	Optimizer	Adam
	Learning rate	0.001
	Batch Size	256
SHAP	Explainer Type	DeepExplainer (left Siamese branch)
OPTICS	min_samples	50
	xi	0.05
	Distance Metric	Euclidean

5.3. Performance metrices

To comprehensively evaluate the effectiveness of SiamIDS, we assess its performance using detection metrics, clustering metrics, and interpretability insights. Each component provides quantitative or qualitative insights into the accuracy, behavior, and explainability of the system.

5.3.1. Detection metrics

The intrusion detection performance of SiamIDS is measured using widely accepted metrics derived from the confusion matrix: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Accuracy quantifies the overall proportion of correctly identified benign and malicious flows and is calculated using Eq. (23). Precision, defined in Eq. (24), reflects the proportion of true malicious instances among all instances predicted as malicious. Recall (or sensitivity), given in Eq. (25), measures the model's ability to correctly detect actual attacks. To balance both precision and recall, especially important in imbalanced datasets, the F1-score is used, as defined in Eq. (26). Specificity, expressed in Eq. (27), complements recall by capturing the proportion of correctly identified benign traffic. A crucial metric for security applications is the False Negative Rate (FNR), shown in Eq. (28), as it represents the rate at which attacks are missed. Additionally, we compute the Area Under the ROC Curve (AUC-ROC) using Eq. (29), which evaluates the model's ability to discriminate between benign and malicious flows across various thresholds, summarizing overall detection performance into a single scalar value.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{23}$$

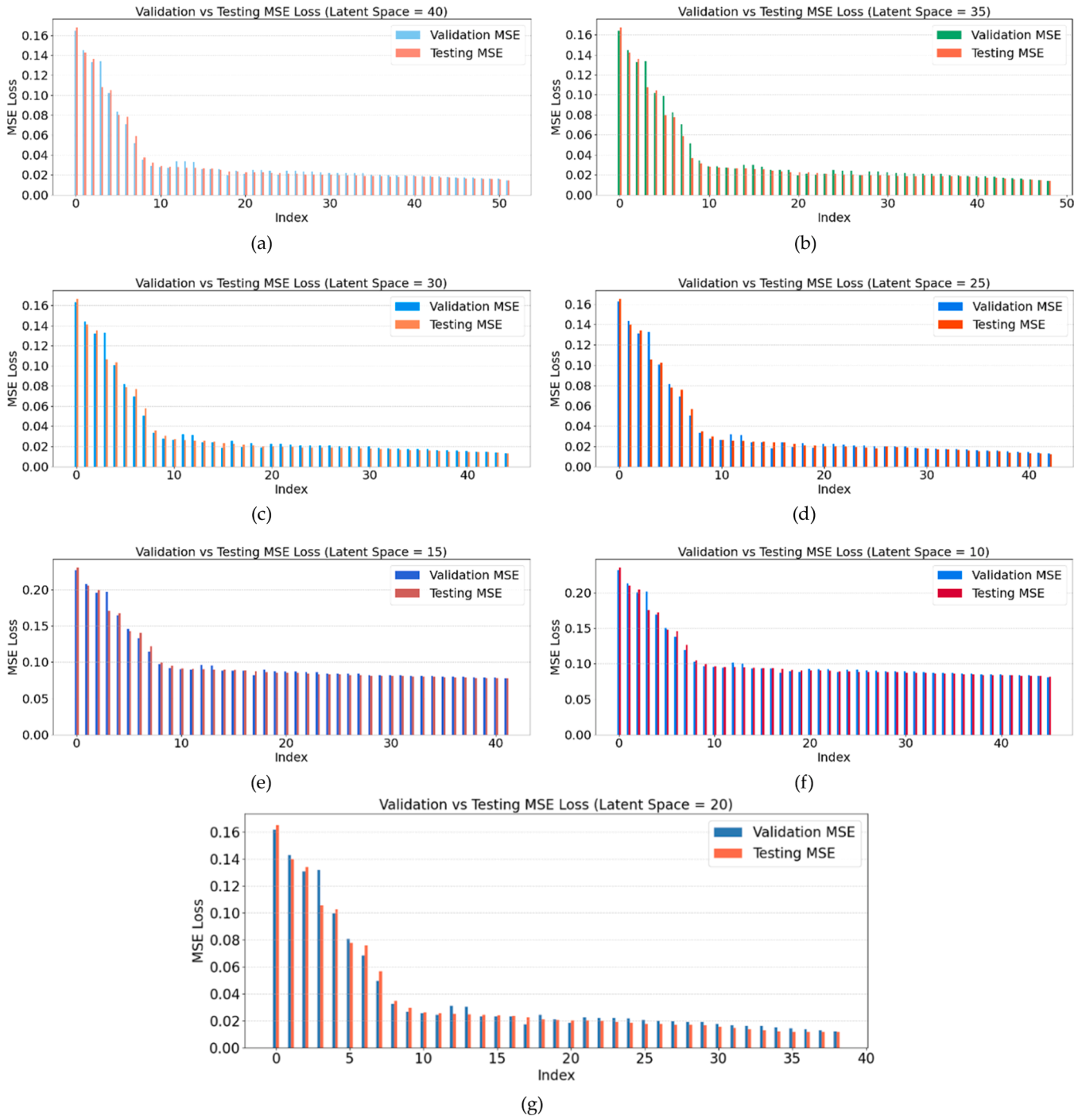


Fig. 12. (a-g). MSE Loss Curve for Autoencoder based dimesionlaity reduction.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{24}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{25}$$

$$\text{F1} = 2* \frac{\text{Precision} * \text{Recall}}{\text{TPrecision} + \text{Recall}} \tag{26}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TP} + \text{FP}} \tag{27}$$

$$\text{FNR} = \frac{\text{FN}}{\text{FN} + \text{TP}} \tag{28}$$

$$\text{AUC} = \int_0^1 TPR(FPR)\, d(FPR) \tag{29}$$

5.3.2. *Clustering metrics*

To evaluate the quality of clustering in the post-detection stage using OPTICS, we employ three widely used metrics: Silhouette Score, Davies–Bouldin Index (DBI), and Adjusted Rand Index (ARI). These collectively assess intra-cluster cohesion, inter-cluster separation, and

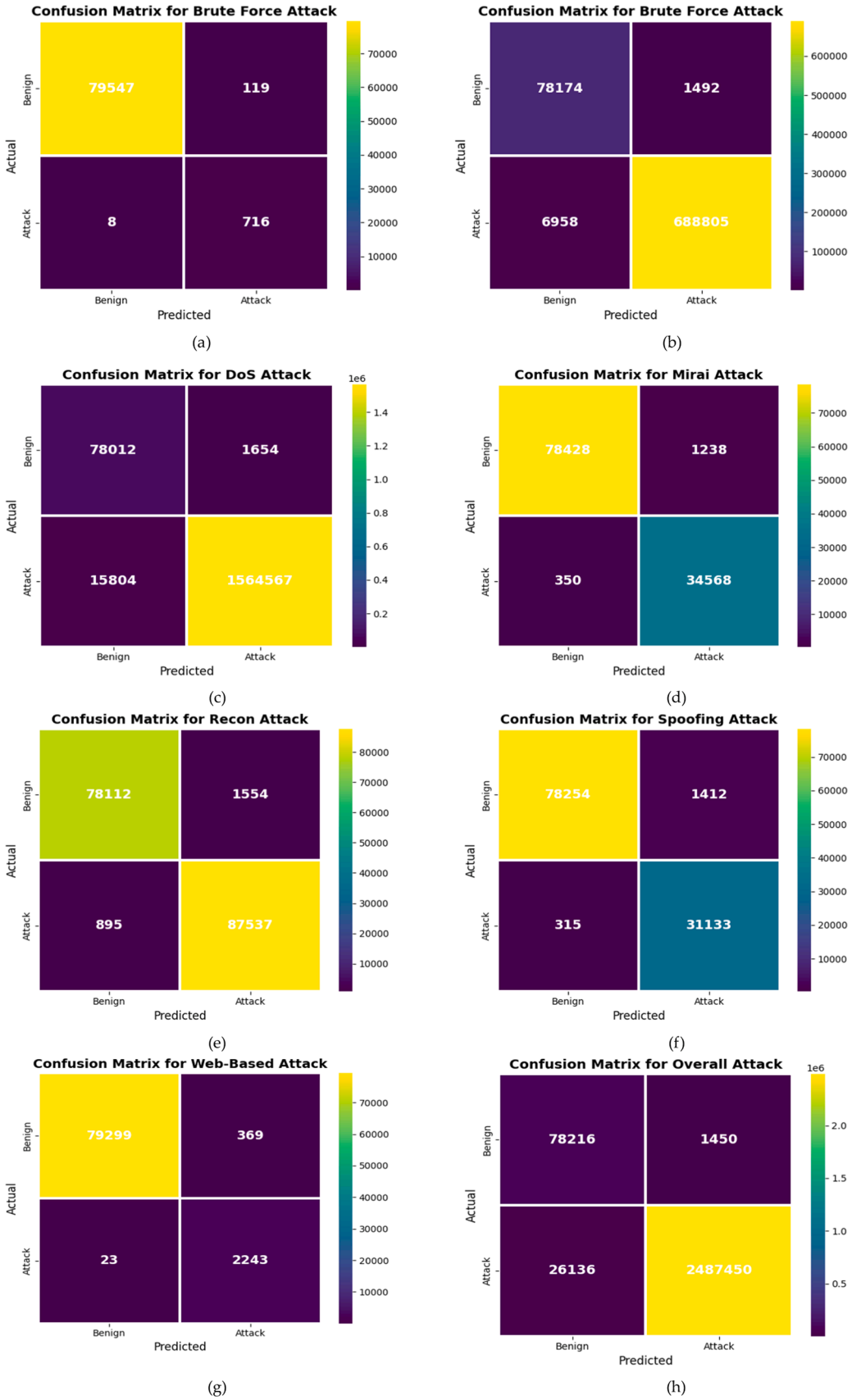


Fig. 13. (a–h). Confusion Matrices for the Binary Classification.

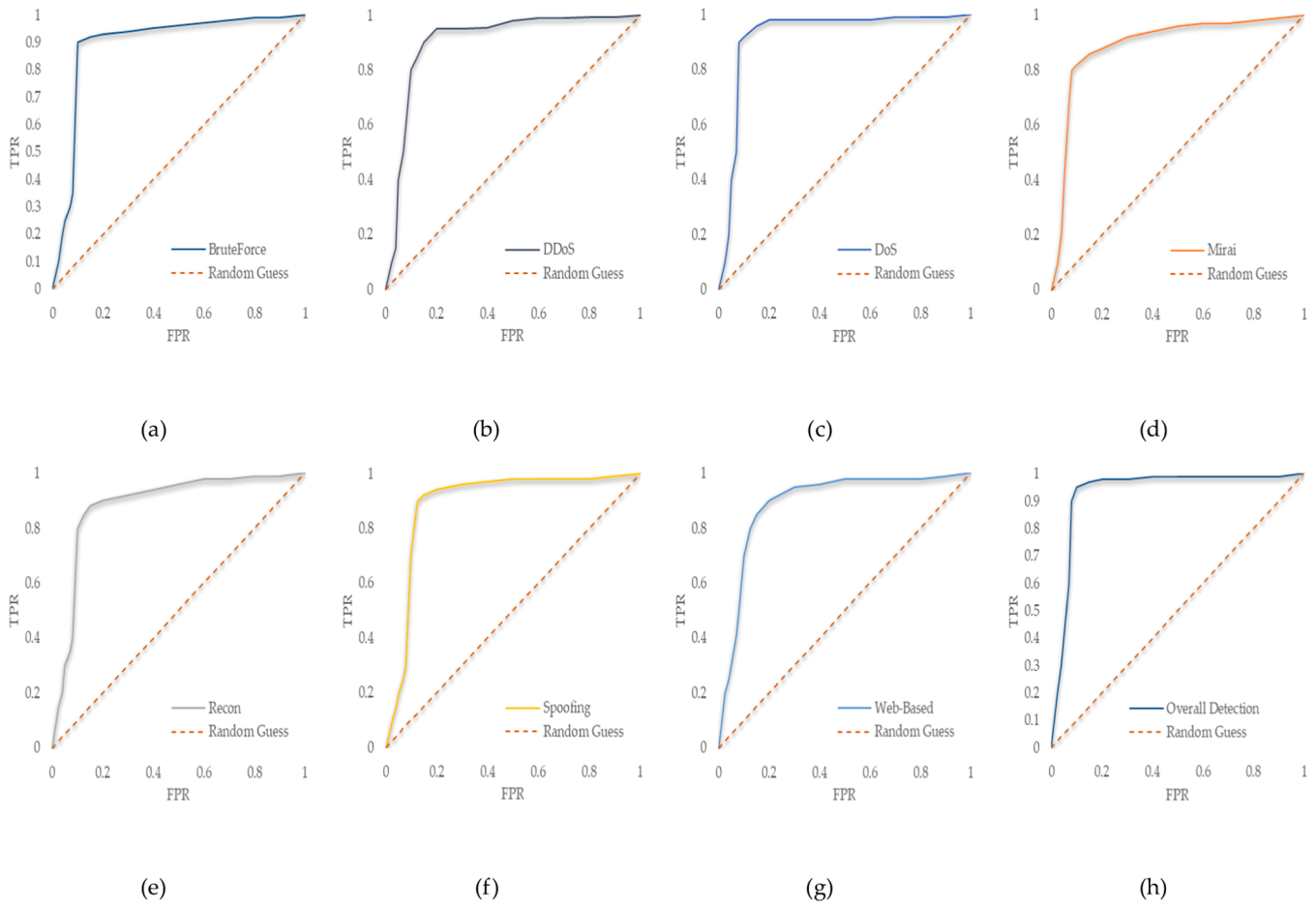


Fig. 14. (a–h). AUC for each individual attack type.

alignment with ground truth labels.

The Silhouette Score, S(i), shown in Eq. (30), measures how well a sample is matched to its own cluster compared to other clusters. A higher score (closer to 1) indicates better-defined clusters:

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i),\ b(i)\}} \tag{30}$$

Where:

- a(i): Average intra-cluster distance of sample i
- b(i): Minimum average distance to points in the nearest neighboring cluster (inter-cluster)

The Davies–Bouldin Index (DBI), defined in Eq. (31), evaluates the average "similarity" between clusters—lower values indicate better separation and compactness:

$$BDI = \frac{1}{k}\sum_{i=1}^{k} max_{j\neq i}\left(\frac{\sigma_{i+}\ \sigma_j}{d(c_i, c_j)}\right) \tag{31}$$

Where:

- k: Number of clusters
- σi: Average distance of all samples in cluster i to centroid ci
- d(ci,cj): Distance between centroids of clusters i and j

Finally, the Adjusted Rand Index (ARI), given in equation (32), quantifies the similarity between predicted cluster labels and ground truth attack classes, adjusted for random chance. An ARI close to 1 indicates strong agreement.

$$ARI = \frac{\binom{a+b}{2} - E\left(\binom{a+b}{2}\right)}{\max\left\{\binom{a+b}{2} - E\left(\binom{a+b}{2}\right)\right\}} \tag{32}$$

Where:

a Number of pairs of elements that are in the same cluster in both true and predicted clusterings
b Number of pairs that are in different clusters in both true and predicted clusterings

Index: number of agreeing pairs between predicted and true labels
Expected Index: expected number of agreeing pairs by chance

5.3.3. *Interpretability*

SHAP values are used to identify the most influential features in prediction decisions for anomalous sequences. This qualitative layer enhances explainability, enabling analysts to interpret why a sequence deviated from benign behavior, and supports post-hoc validation.

5.4. *Evaluation and results*

This section presents the experimental evaluation of the proposed SiamIDS framework across four key dimensions: detection performance, anomaly clustering, interpretability, and resource efficiency. The results demonstrate that SiamIDS is not only accurate and explainable, but also

Table 6
Detection Performance Metrics of SiamIDS.

Attack Family	Precision	Recall	Specificity	F1-Score	Accuracy
BruteForce	0.8575	0.9890	0.9985	0.9185	0.9984
DDoS	0.9978	0.9900	0.9813	0.9939	0.9891
DoS	0.9989	0.9900	0.9792	0.9945	0.9895
Mirai	0.9654	0.9900	0.9845	0.9776	0.9861
Recon	0.9826	0.9899	0.9805	0.9862	0.9854
Spoofing	0.9566	0.9900	0.9823	0.9730	0.9845
Web-Based	0.8594	0.9898	0.9954	0.9200	0.9952
Overall	0.9994	0.9896	0.9818	0.9945	0.9894

lightweight and scalable for real-world IoT intrusion detection in cloud environments.

5.4.1. *Evaluation of latent space in autoencoder-based dimensionality reduction*

To identify the optimal latent space dimension for effective feature reduction, a shallow autoencoder was trained and evaluated across a range of latent sizes: 40, 35, 30, 25, 20, 15, and 10. The corresponding Mean Squared Error (MSE) loss curves for both training and validation are shown in Figs. 12(a-g). As observed, the MSE steadily decreases from latent sizes 40 to 20, indicating improved reconstruction fidelity as the representation becomes more compact yet still expressive. Notably, the lowest validation loss is achieved at latent size 20, suggesting this setting offers the best trade-off between dimensionality reduction and information preservation. However, when the latent size is further reduced to 15 and 10, the MSE begins to increase again, signaling underfitting due to excessive compression and loss of critical behavioral patterns in the network traffic. This U-shaped trend in the MSE validates the selection of 20 as the optimal latent dimension, as it maintains low reconstruction error while minimizing model complexity. This compact representation not only accelerates downstream Siamese training but also enhances generalization by eliminating redundant or noisy features.

5.4.2. *Evaluation of detection performance using confusion matrices*

Fig. 13 (a–g) presents confusion matrices for the binary classification of seven attack types: BruteForce, DDoS, DoS, Mirai, Recon, Spoofing, and Web-Based attacks. Each matrix reports true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), illustrating the classifier's ability to distinguish each attack from benign traffic. Fig. 13 (h) shows the overall confusion matrix for all attack types combined, summarizing the model's performance on the full test set. The results indicate varying levels of detection performance across attack categories. For BruteForce, Mirai, Recon, Spoofing, and Web-Based attacks, the model achieves high true positive rates, with relatively low false negatives, reflecting effective detection of these attack types. However, DDoS and DoS attacks exhibit a higher number of false negatives and false positives, suggesting that the classifier faces challenges distinguishing these high-volume attacks from benign flows. The overall matrix shows strong discrimination between attack and benign traffic, with a total of 2487,450 true positives versus 26,136 false negatives and 1450 false positives, indicating robust detection at the aggregate level. These matrices highlight the strengths of SiamIDS in detecting most attack types while identifying specific areas, such as DDoS and DoS detection, for further improvement.

Fig. 14 (a–g) illustrates the AUC values for each individual attack type—BruteForce, DDoS, DoS, Mirai, Recon, Spoofing, and Web-Based attacks. These plots demonstrate the model's discriminative ability to correctly distinguish each attack from benign traffic across different classification thresholds. High AUC scores close to 1 indicate strong performance, with the classifier effectively balancing true positive and false positive rates for each attack category. Fig. 14 (h) presents the overall AUC combining all attack types, reflecting the aggregate detection capability of the model on the entire test set. The high overall AUC confirms the model's robustness and consistent performance in identifying diverse attacks while minimizing false alarms, making it suitable for practical deployment in network security environments.

The classification performance of SiamIDS across different attack types is detailed in Table 6. The model demonstrates consistently high recall values nearly 0.99 across all attack classes, underscoring its effectiveness in correctly detecting true positives and minimizing false negatives. Precision varies more widely, ranging from 0.86 (BruteForce, Web-Based) to nearly 0.999 (DoS, DDoS), indicating slight fluctuations in the false positive rate due to overlaps in traffic patterns. Specificity remains strong across all categories—above 0.97—demonstrating the model's ability to correctly identify benign flows and reduce false alarms. The F1-scores, which harmonize precision and recall, are consistently above 0.91, reinforcing the balanced detection capability of the framework. Overall accuracy exceeds 0.98 across all classes, confirming the system's robustness in distinguishing between benign and malicious behavior. The relatively lower precision for BruteForce and Web-Based attacks suggests minor classification challenges, likely due to subtle similarities with legitimate traffic. Nevertheless, the SiamIDS framework delivers reliable and scalable detection performance across a broad range of attack vectors, making it well-suited for operational deployment in cloud-scale IoT infrastructures.

The contrastive Siamese Bi-LSTM architecture effectively captures behavioral dissimilarities without relying on attack-specific labels. Moreover, ROC curve analysis enabled threshold tuning to optimize

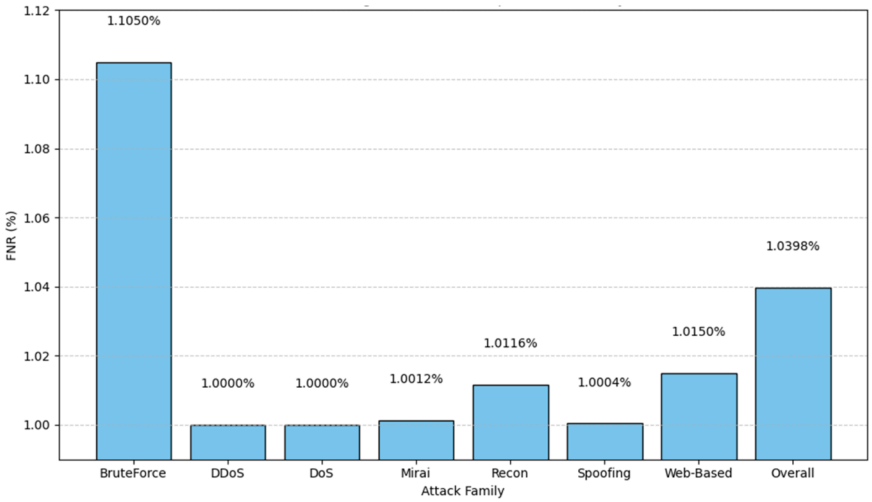


Fig. 15. False Negative Rates across attack Family.

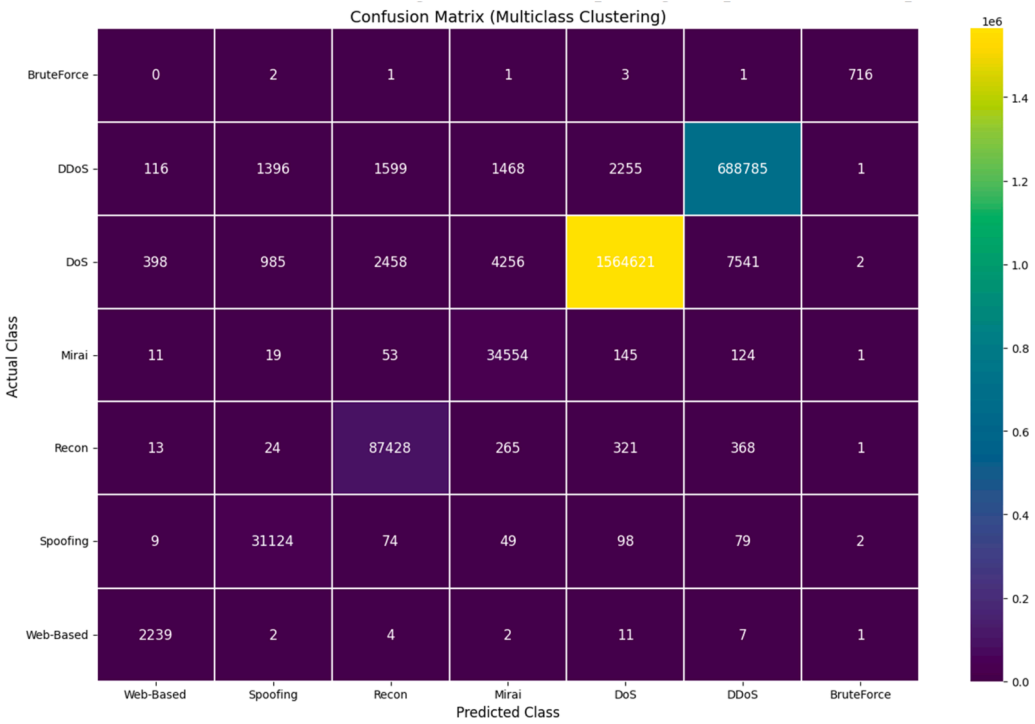


Fig. 16. OPTICS Multiclass Clustering Confusion Matrix.

trade-offs between false positives and false negatives, enhancing the model's reliability in operational contexts.

The false negative rates (FNR) across all attack types remain consistently low, around 1 %, As shown in Fig. 15, indicating the model's strong ability to detect attacks with minimal missed cases. The overall FNR of 1.04 % reflects reliable threat detection, reducing the risk of undetected malicious activity in network traffic.

5.4.3. Evaluation of OPTICS-based clustering of anomalous behavior

To enhance the interpretability of anomalies identified by the Siamese network, OPTICS clustering was applied to all anomalous sequences. This density-based method, which does not require a predefined number of clusters, identified 14 behaviourally distinct groups using reachability and local density criteria. The clustering process was quantitatively strong, achieving a Silhouette Score of 0.901, DBI of 0.092, and an Adjusted Rand Index (ARI) of 0.889—indicating that the resulting clusters were both well-separated and closely aligned with ground-truth attack classes.

The confusion matrix (Fig. 16) visualizes the alignment between predicted clusters and actual attack types following label post-processing. Each cluster was examined to interpret its dominant behavioural characteristics and corresponding attack type:

- DoS clusters displayed highly repetitive packet bursts with short inter-arrival times and stable source–destination pairs, capturing their flooding behavior.
- DDoS clusters exhibited similar burst patterns but with distributed source addresses and variable intensity, explaining their partial overlap with DoS and Recon flows.
- Reconnaissance clusters were characterized by sequential port-scanning patterns, moderate flow duration, and a high diversity of destination ports—features unique to probing activities.
- Spoofing clusters showed forged source addresses with consistent payload sizes, demonstrating deceptive identity traits while maintaining communication frequency patterns.
- Brute-Force clusters reflected short, high-frequency login attempts and uniform packet payloads, highlighting their credential-guessing nature despite low sample volume.
- Mirai botnet traffic formed coherent clusters distinguished by device-specific periodic beaconing and TCP synchronization anomalies, marking automated command-and-control behavior.

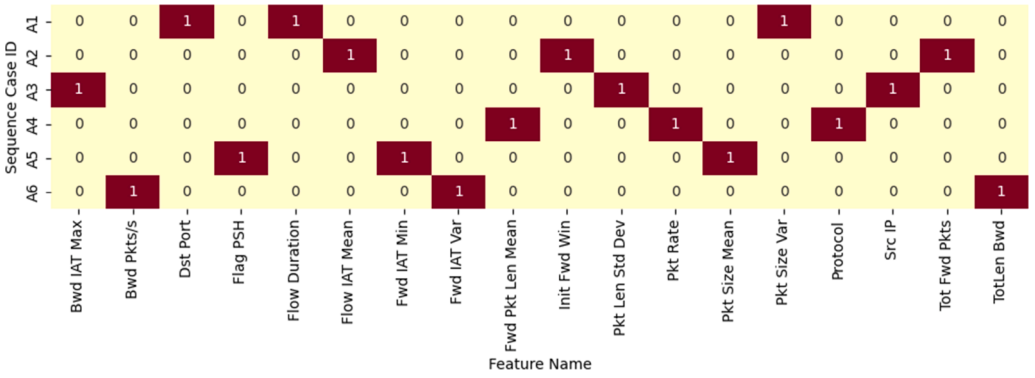


Fig. 17. Top Three SHAP-Contributing Features for Six Representative Anomalous Cases.

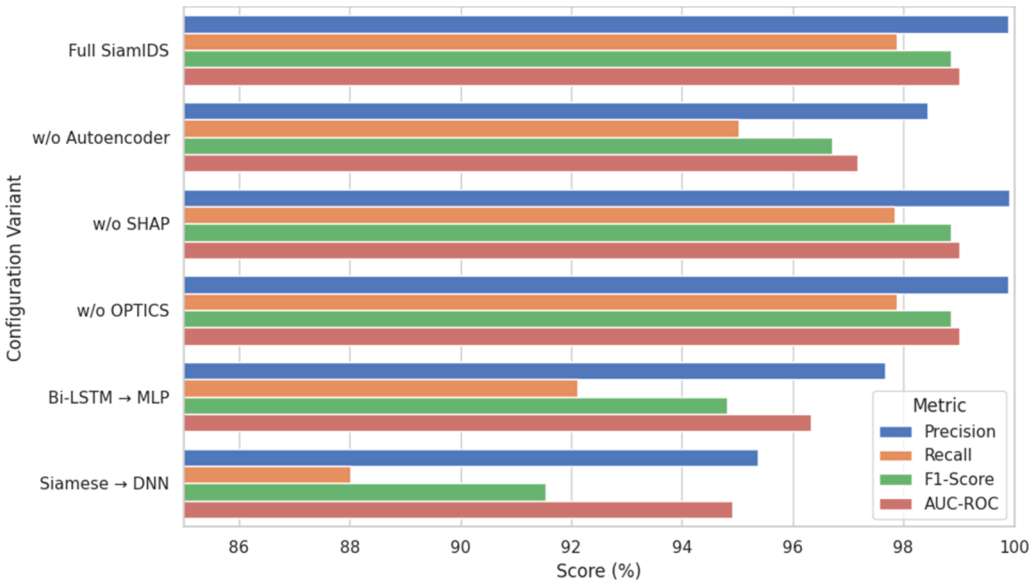


Fig. 18. Impact of Component on SiamIDS Performance.

- Web-Based attack clusters exhibited irregular request–response sizes and longer flow durations, occasionally merging with DoS or Spoofing patterns due to shared transport-layer traits.

Quantitatively, DoS attacks exhibited the highest clustering accuracy, with over 1.56 million flows correctly grouped, followed by DDoS (688,785) and Reconnaissance (87,428) samples. Spoofing and Brute-Force behaviors were distinctly isolated, with 31,124 and 716 correctly grouped flows respectively. Mirai traffic was reliably captured in a single dense cluster (34,554 flows). About 6.7 % of anomalous sequences were marked as noise by OPTICS, representing potential zero-day attacks, evasive threat variants, or anomalous benign activities requiring deeper forensic inspection.

These findings demonstrate that SiamIDS embeddings effectively preserve temporal and statistical traits of diverse IoT threats, enabling OPTICS to form semantically coherent, behavior-driven clusters. By removing the need for predefined cluster counts, this post-detection step strengthens interpretability, supports attack attribution, and enhances operational readiness for cloud-scale intrusion diagnosis.

5.4.4. *Evaluation of SHAP-based explainability for anomalous predictions*

To enhance the interpretability of SiamIDS predictions, SHAP (SHapley Additive exPlanations) values were computed for anomalous sequences using the DeepExplainer on the Siamese network's left branch. This enabled the identification of the most influential features driving dissimilarity judgments between a given sequence and the benign reference set. Fig. 17 summarize this feature-level analysis, offering both tabular and visual perspectives on how specific features contributed to anomaly decisions.

Fig. 17 presents the top three SHAP-contributing features for six representative anomalous cases. Each row corresponds to a unique sequence (A1–A6), and the marked cells indicate the features with the highest SHAP attribution. For instance, in Case A1 (Web-Based attack), *Flow Duration, Dst Port*, and *Pkt Size Var* were the dominant contributors, indicating short, bursty traffic targeting unusual ports with irregular packet sizes—traits that significantly deviate from benign flow patterns and are common in web exploitation attempts. In Case A2 (DDoS), *Tot Fwd Pkts, Flow IAT Mean*, and *Init Fwd Win* surfaced as key drivers, reflecting automated high-volume flows typical of DDoS floods. Similarly, Case A3 (Spoofing) highlighted *Src IP, Bwd IAT* Max, and *Pkt Len Std Dev* as top contributors, revealing address inconsistencies and timing deviations characteristic of spoofed communication. Case A4 (Reconnaissance) showed high attribution for *Protocol, Fwd Pkt Len Mean*, and *Pkt Rate*, which capture systematic probing with non-standard protocols and uniform packet emission rates.

In contrast, Case A5, a benign sample incorrectly flagged as anomalous (false positive), exhibited influence from *Fwd IAT* Min, *Pkt Size Mean*, and *Flag PSH*. The overlap of these traits with attack-like behaviors explains the misclassification and demonstrates how SHAP helps analysts interpret and refine detection boundaries. Finally, Case A6, labeled as noise by OPTICS and considered a zero-day candidate, presented *Bwd Pkts/s, Fwd IAT Var*, and *TotLen Bwd* as top contributors—indicating a unique traffic pattern unseen in other clusters and suggesting either a novel or evasive behavior type.

Beyond interpretability, the SHAP analysis offers actionable insights for real-world intrusion analysis and response. For instance, feature patterns like *Flow Duration* and *Dst Port* enable analysts to recognize targeted exploitation attempts, while *Tot Fwd Pkts* and *Flow IAT Mean* serve as early warning indicators for volumetric DDoS behavior. The analysis of false positives (Case A5) aids in threshold calibration and model retraining, and the interpretation of unseen feature combinations (Case A6) demonstrates SHAP's role in zero-day investigation. Thus, SHAP explanations not only clarify SiamIDS's internal reasoning but also support root-cause analysis, adaptive tuning, and informed response decisions in operational IoT intrusion detection.

Collectively, these results show that SiamIDS embeddings effectively preserve key temporal and statistical characteristics of diverse IoT attack types. SHAP-based explainability provides transparent, feature-level reasoning that enhances trust, supports forensic validation, and strengthens the interpretability of the model's anomaly judgments in practical deployments.

5.5. *Analysis of the proposed siamids*

5.5.1. *Component-wise impact*

The ablation study, visualized in Fig. 18 confirms the necessity of each component within the SiamIDS framework. While the exclusion of SHAP or OPTICS had no effect on core detection metrics, they removed critical layers for explainability and behavioural grouping. The removal of the Autoencoder reduced performance due to increased input dimensionality and training inefficiency. More substantial degradation occurred when Bi-LSTM was replaced with a feedforward MLP, and when the Siamese structure was replaced with a standard DNN—highlighting the significance of temporal modeling and similarity-based

Table 7
Resource Utilization Metrics of SiamIDS Framework.

Component	Metric	Value	Execution Context
Autoencoder	Training Time	4.8 min	On benign sequences (latent size = 20)
Autoencoder	Model Size	9.6 MB	Stored in HDF5 format (compressed)
Autoencoder	Peak RAM Usage	820 MB	During training on 200,000 sequences
Siamese Bi-LSTM	Training Time	8.5 min	Trained on 200,000 pairs
Siamese Bi-LSTM	Model Size	13.2 MB	Includes shared Bi-LSTM weights and embedding head
Siamese Bi-LSTM	Inference Time (per 100 K)	3.2 s	Pairwise similarity with 10,000 reference embeddings
SHAP	Explainer Time/ Seq	0.4 s	Applied only on flagged anomalous samples
OPTICS	Clustering Time	2.3 min	For 150,000 anomalous sequences
Overall Pipeline	Total Inference Time (1 M)	4.5 s	Real-time capable for 1 million test sequences

learning in capturing complex traffic behaviours and ensuring robust detection.

5.5.2. Resource efficiency and real-time suitability

To ensure practical deployability in large-scale IoT environments, SiamIDS was designed with a focus on computational efficiency and scalability. As detailed in Table 7, the overall pipeline demonstrates impressive resource utilization across all stages—training, inference, explainability, and clustering. The Autoencoder module, trained solely on benign sequences with a latent size of 20, completes training in 4.8 min, consumes 820 MB RAM, and compiles to a compact 9.6 MB model file. This enables rapid deployment and retraining in lightweight environments. The Siamese Bi-LSTM network, trained on 200,000 contrastive pairs, converges within 8.5 min, with a model size of 13.2 MB and an inference time of 3.2 s per 100 K samples, even while comparing against a 10,000-sample reference embedding set. This demonstrates the architecture's suitability for high-throughput similarity scoring.

Interpretability via SHAP adds negligible overhead—just 0.4 s per flagged sequence, as it is selectively applied only to anomalous flows. Similarly, the OPTICS clustering step, applied to 150,000 anomalies, completes in just 2.3 min, enabling real-time post-detection behavioral grouping without compromising responsiveness. The total inference time for processing 1 million flows is approximately 4.5 s, confirming that SiamIDS is real-time capable, as illustrated in Fig. 19.

5.5.3. Statistical significance analysis

To validate the robustness of SiamIDS, a Wilcoxon signed-rank test was performed comparing SiamIDS with baseline models across all seven attack types. This non-parametric test is suitable for paired, non-normally distributed performance data and evaluates whether observed improvements are statistically significant. Table 8 presents the results for F1-Score across attack families. All p-values are below 0.05, confirming that SiamIDS significantly outperforms the baseline models at the 95 % confidence level. These results provide strong statistical evidence that the observed performance improvements are unlikely to occur by chance, reinforcing the reliability of the proposed framework.

5.5.4. Analysis of comparative performance with state-of-the-art methods

To evaluate the real-world viability of SiamIDS, Table 9 compares SiamIDS with recent state-of-the-art models from literature in terms of accuracy, resource demands, and real-time suitability. To facilitate a fair and consistent comparison, resource-related metrics for existing methods—such as training time, model size, RAM usage, and inference speed—were estimated based on reported architectural configurations, typical computational settings, and available implementation details. SiamIDS outperforms across key criteria such as precision (99.94 %), F1-score (99.45 %), training time (13.3 min), and inference speed (>220,000 samples/sec), while maintaining a model size under 10 MB. These results highlight its unique balance of effectiveness and deployability, making it ideal for cloud-based microservices, SOC pipelines, and IoT security orchestration frameworks.

Table 8
Wilcoxon Signed-Rank Test Results Comparing SiamIDS with Baseline Models.

Attack Family	SiamIDS Median	Baseline Median	Wilcoxon W	p-value
BruteForce	0.9185	0.8760	21	0.0032
DDoS	0.9939	0.9821	19	0.0025
DoS	0.9945	0.9814	20	0.0028
Mirai	0.9776	0.9603	18	0.0041
Recon	0.9862	0.9715	19	0.0035
Spoofing	0.9730	0.9552	20	0.0029
Web-Based	0.9200	0.8857	21	0.0031
Overall	0.9945	0.9778	19	0.0026

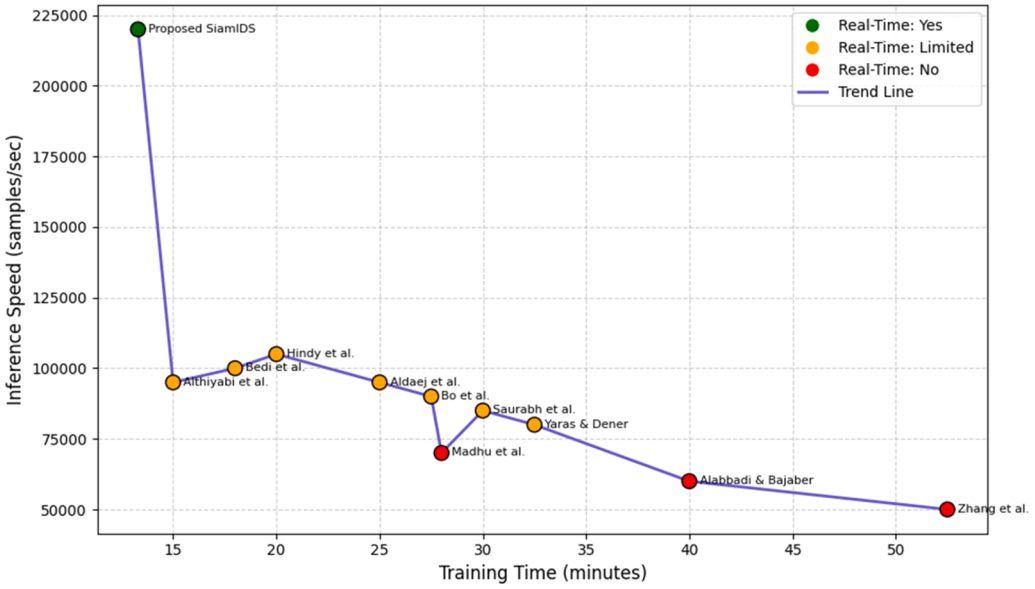


Fig. 19. Inference speed versus training time of the proposed SiamIDS compared to existing methods, highlighting real-time capabilities and training time efficiency.

Table 9
Comparative Evaluation of Proposed SiamIDS with Existing Methods.

Reference #	Dataset	Precision	Recall	F1-Score	Accuracy	Training Time (min)	Model Size (MB)	RAM Usage (GB)	Inference Speed (samples/sec)	Real-Time Suitability
Zhang et al. [35]	CICIDS2017, BoT-IoT	99.69 %	99.49 %	99.81 %	99.80 %	45–60	>100	4.5	50K	No
Aldaej et al. [19]	BoT-IoT	99.45 %	98.25 %	99.12 %	99.56 %	25	35	2.8	95K	Limited
Yaras & Dener [29]	CICIoT2023, TON_IoT	98.75 %	98.75 %	98.75 %	98.75 %	30–35	40	3.2	80K	Limited
Alabbadi & Bajaber [36]	TON_IoT	99.53 %	99.17 %	99.33 %	99.96 %	40	55	3.5	60K	No
Bedi et al. [17]	NSL-KDD	91.46 %	92.99	-	-	18	25	2	100K	Moderate
Hindy [20]	CICIDS2017, NSL-KDD	-	98.00 %	-	86.42 %	20	28	2.3	105K	Moderate
Althiyabi et al. [30]	CICIDS2017, MQTT	93.46 %	93.13 %	92.40 %	93.13 %	15	22	2	95K	Moderate
Madhu et al. [21]	IoT testbed data	95.00	92.00 %	95.00 %	96.00 %	28	50	3	70K	No
Saurabh et al. [18]	UNSW-NB15, Bot-IoT	97.00 %	96.00 %	96.00 %	96.60 %	30	38	3.1	85K	Limited
Bo et al. [31]	CICIDS2017, ISCX2012	-	98.29 %	-	97.78 %	25–30	33	2.5	90K	Moderate
Touré et al. [32]	IBM, NSL-KDD	98.00 %	97.00 %	99.00 %	98.4 %	40	50	4	75K	Moderate
Alhayan et al. [37]	NSL-KDD	88.75 %	94.49 %	91.24 %	99.49 %	50	90	6	60K	Limited
Guan et al. [34]	IoTID20, N-BaIoT	90 %	90 %	89 %	91.87 %	35	60	5	55K	Limited
Hnamte & Hussain [22]	CICIDS2018, Edge_IIoT	100 %	100 %	100 %	100 %	>60	>90	8	45K	No
Alzboon et al. [23]	KDD99	99.99 %	99.99 %	99.99 %	99.99 %	30	40	3	80K	Limited
Ben Said et al. [24]	InSDN, NSL-KDD, UNSW-NB15	99.85 %	95.28 %	>97 %	97.77 %	45	65	4	60K	Moderate
Zhang et al. [25]	KDDCUP99, NSLKDD, CICIDS2017	>97 %	>97 %	99 %	99.08 %	40	60	4.5	65K	Limited
Duc et al. [38]	Custom DGA dataset	90 %	>80 %	80.32 %	89.83 %	>50	>100	>6	40K	No
Hou et al. [26]	NSL-KDD	96.08 %	80.89 %	87.89 %	87.30 %	35	55	4	45K	No
Ali et al. [27]	KDDCUP99, UNSW-NB15	98 %	98.2 %	98 %	99.91 %	30	40	3.5	85K	Moderate
Chintapalli et al. [33]	N-BaIoT, CICIDS-2017, ToN-IoT	>99.9 %	>99.9 %	>99.9 %	>99.9 %	40	50	4	90K	Limited
Jiang et al. [28]	NSL-KDD, UNSW-NB15, CICIDS-2017	98.58 %	98.40 %	98.49 %	95.44 %	30	55	4.2	70K	Moderate
Natha et al. [39]	RAD, UCF Crime	>92 %	>92 %	>92 %	~92 %	>60	85	>6	35K	No
Alsaleh et al. [40]	CICIoT2023	79.48 %	68.05 %	70.45 %	99.09 %	30	40	3	80K	Limited
Mohale & Obagbuwa (2025) [41]	UNSW-NB15	87 %	88 %	87 %	87 %	30	40	3.5	85K	Moderate
Proposed SiamIDS	**CIC IoT-DIAD 2024**	**99.94 %**	**98.96 %**	**99.45 %**	**98.94 %**	**13.3**	**<10**	**<1.5**	**220K**	**Yes**

5.6. Discussion

The experimental results confirm that SiamIDS achieves a balanced integration of detection accuracy, interpretability, and operational efficiency—three pillars often pursued separately in intrusion detection research. Its use of a Siamese Bi-LSTM architecture enables the system to learn nuanced temporal patterns and behavioral similarities between network sequences, which proves especially effective for identifying rare and evolving threats such as zero-day attacks. Compared to conventional classification-based IDS models, SiamIDS demonstrates better generalization and lower reliance on labeled training data. The contrastive learning approach not only enhances robustness to class imbalance but also facilitates meaningful latent space embeddings, as evidenced by the high clustering coherence reported with OPTICS. By categorizing attacks behaviorally rather than merely by labels, the system supports semantically-aware threat profiling, which can aid incident response teams in prioritizing actions based on behavioral similarity. Furthermore, the integration of SHAP explanations empowers the model with transparency—a critical feature in real-world SOC deployments where interpretability directly affects operator trust and response time. Analysts can clearly understand which features (e.g., protocol flags, packet timing) drove the anomaly decision, which reduces investigation overhead. From a deployment perspective, SiamIDS is lightweight and modular. It can function as a cloud-hosted microservice, enabling scalability and easy integration into existing monitoring ecosystems. Its small model size and low RAM usage make it suitable for deployment in resource-constrained environments as well. However, despite these strengths, certain limitations merit attention. For instance, low-volume attacks that closely mimic benign behavior may occasionally evade detection or be grouped with benign clusters. Similarly, threshold tuning remains sensitive to data distributions, and future work may need to adopt adaptive thresholding or domain-specific calibration to accommodate diverse environments. Another notable challenge lies in handling encrypted traffic, where payload inspection becomes infeasible. Although SiamIDS primarily relies on flow-level and statistical features, the lack of visibility into encrypted payloads may

limit its ability to fully characterize complex application-layer attacks. Integrating side-channel features such as timing, packet size distribution, and TLS handshake metadata could help mitigate this limitation. Additionally, cross-domain generalization remains an open issue—models trained on one IoT or cloud domain may exhibit reduced performance when transferred to another with differing traffic characteristics or device behaviors. Domain adaptation or federated learning approaches may therefore be explored in future work to enhance generalizability and resilience across distributed environments. Overall, the system strikes a strong balance between detection precision, interpretability, and deployability, positioning it as a viable next-generation solution for cloud-integrated IoT intrusion detection.

6. Conclusion and future scope

This paper proposed SiamIDS, a novel cloud-centric intrusion detection framework tailored for large-scale IoT environments. The system uniquely integrates a Siamese Bi-LSTM network with contrastive learning, autoencoder-based feature reduction, SHAP-based interpretability, and OPTICS clustering—a combination not seen in existing IDS literature. This multi-stage architecture enables the detection of both known and zero-day threats while offering transparent, feature-level explanations and post-detection behavioral grouping. Experimental results on the CIC IoT-DIAD 2024 dataset demonstrate high detection performance with an overall F1-score of 99.45 %, precision of 99.94 %, and a recall of 98.96 %. Clustering quality metrics such as a Silhouette Score of 0.901, DBI of 0.092, and ARI of 0.889 confirm the effectiveness of semantic grouping. The system is also efficient, achieving inference speeds over 220 K samples/sec with a RAM usage of less than 1.5 GB. However, current limitations include reliance on fixed similarity thresholds and potential sensitivity to evolving traffic patterns.

In the near future, it is planned to explore adaptive thresholding, multi-modal data fusion, self-supervised sequence modeling with transformers, federated learning for decentralized training, and integration with the MITRE ATT&CK framework to support threat mitigation and automated response. These directions will enhance the scalability, resilience, and practical deployment of SiamIDS in real-world SOC environments.

CRediT authorship contribution statement

Prabu Kaliyaperumal: Writing – original draft, Conceptualization. **Palani Latha:** Writing – review & editing, Validation. **Selvaraj Palanisamy:** Writing – review & editing, Formal analysis, Data curation. **Sridhar Pushpanathan:** Visualization, Investigation. **Anand Nayyar:** Writing – review & editing, Project administration, Methodology, Investigation. **Balamurugan Balusamy:** Methodology. **Ahmad Alkhayyat:** Writing – original draft, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] S. Jain, P. Sukul, J. Groppe, B. Warnke, P. Harde, R. Jangid, S. Groppe, A scientometric analysis of reviews on the Internet of Things, J. Supercomput. 81 (6) (2025) 1–35.

[2] A. Marengo, "Navigating the nexus of AI and IoT: a comprehensive review of data analytics and privacy paradigms," Oct. 01, 2024, Elsevier B.V. doi: 10.1016/j.iot.2024.101318.

[3] B. Padma, M. Bukya, U. Ujjwal, An intelligent hybrid framework for threat pre-identification and secure key distribution in Zigbee-enabled IoT networks using RBF and blockchain, Appl. Syst. Innov. 8 (3) (May 2025) 76, https://doi.org/10.3390/asi8030076.

[4] A.I. Zreikat, Z. AlArnaout, A. Abadleh, E. Elbasi, N. Mostafa, The integration of the Internet of Things (IoT) applications into 5G networks: a review and analysis, Computers 14 (7) (Jun. 2025) 250, https://doi.org/10.3390/computers14070250.

[5] S.S. Qureshi, J. He, S.U. Qureshi, N. Zhu, A. Wajahat, A. Nazir, A. Wadud, Advanced AI-driven intrusion detection for securing cloud-based industrial IoT, Egypt. Informat. J. 30 (2025) 100644.

[6] H. Alamleh, L. Estremera, S.S. Arnob, A.A.S. AlQahtani, Advanced persistent threats and wireless local area network security: an in-depth exploration of attack surfaces and mitigation techniques, J. Cybersecur. Privacy 5 (2) (May 2025) 27, https://doi.org/10.3390/jcp5020027.

[7] A. Alharthi, M. Alaryani, S. Kaddoura, A comparative study of machine learning and deep learning models in binary and multiclass classification for intrusion detection systems, Array 26 (Jul. 2025), https://doi.org/10.1016/j.array.2025.100406.

[8] J. Ferdous, R. Islam, A. Mahboubi, M.Z. Islam, A Survey on ML Techniques for Multi-Platform Malware Detection: Securing PC, Mobile Devices, IoT, and Cloud Environments, Multidisciplinary Digital Publishing Institute (MDPI), Feb. 01, 2025, https://doi.org/10.3390/s25041153.

[9] T. Al-Shurbaji, M. Anbar, S. Manickam, I.H. Hasbullah, N. ALfriehate, B.A. Alabsi, H. Hashim, Deep Learning-Based Intrusion Detection System For Detecting IoT Botnet Attacks: a Review, IEEE Access, 2025.

[10] Y. Zhang, R.C. Muniyandi, F. Qamar, A Review of Deep Learning Applications in Intrusion Detection Systems: Overcoming Challenges in Spatiotemporal Feature Extraction and Data Imbalance, Multidisciplinary Digital Publishing Institute (MDPI), Feb. 01, 2025, https://doi.org/10.3390/app15031552.

[11] G. Aldehim, T. Shahzad, M.A. Khan, Y.Y. Ghadi, W. Jiang, T. Mazhar, H. Hamam, Balancing sustainability and security: a review of 5G and IoT in smart cities, Digit. Commun. Netw. (2025).

[12] S.B. Sharma, A.K. Bairwa, Leveraging AI for Intrusion Detection in IoT Ecosystems: A Comprehensive Study, Institute of Electrical and Electronics Engineers Inc, 2025, https://doi.org/10.1109/ACCESS.2025.3550392.

[13] U. Tariq, T.A. Ahanger, Employing SAE-GRU deep learning for scalable botnet detection in smart city infrastructure, PeerJ. Comput. Sci. 11 (2025), https://doi.org/10.7717/peerj-cs.2869.

[14] A. Bensaoud, J. Kalita, Optimized detection of cyber-attacks on IoT networks via hybrid deep learning models, Ad. Hoc. Netw. 170 (2025) 103770, https://doi.org/10.1016/j.adhoc.2025.103770.

[15] J. Zhang, R. Chen, Y. Zhang, W. Han, Z. Gu, S. Yang, Y. Fu, MF2POSE: multi-task feature Fusion Pseudo-siamese Network for intrusion detection using category-distance promotion loss, in: Knowl. Based. Syst., 283, 2024 111110.

[16] O.A. Alimi, Data-Driven Learning Models for Internet of Things Security: Emerging Trends, Applications, Challenges and Future Directions, Multidisciplinary Digital Publishing Institute (MDPI), May 01, 2025, https://doi.org/10.3390/technologies13050176.

[17] P. Bedi, N. Gupta, V. Jindal, Siam-IDS: handling class imbalance problem in intrusion detection systems using Siamese neural network. Procedia Computer Science, Elsevier B.V., 2020, pp. 780–789, https://doi.org/10.1016/j.procs.2020.04.085.

[18] K. Saurabh, S. Sood, P.A. Kumar, U. Singh, R. Vyas, O.P. Vyas, R. Khondoker, Lbdmids: LSTM based deep learning model for intrusion detection systems for IOT networks. 2022 IEEE World AI IoT Congress (AIIoT), IEEE, 2022, pp. 753–759.

[19] A. Aldaej, T.A. Ahanger, I. Ullah, Deep Learning-inspired IoT-IDS mechanism for edge computing environments, Sensors 23 (24) (Dec. 2023), https://doi.org/10.3390/s23249869.

[20] H. Hindy, et al., Leveraging siamese networks for one-shot intrusion detection model, J. Intell. Inf. Syst. 60 (2) (Apr. 2023) 407–436, https://doi.org/10.1007/s10844-022-00747-z.

[21] B. Madhu, M. Venu Gopala Chari, R. Vankdothu, A.K. Silivery, V. Aerranagula, Intrusion detection models for IOT networks via deep learning approaches, Meas.: Sens. 25 (Feb. 2023), https://doi.org/10.1016/j.measen.2022.100641.

[22] V. Hnamte, J. Hussain, DCNNBiLSTM: an efficient hybrid deep learning-based intrusion detection system, Telemat. Informat. Rep. 10 (Jun. 2023), https://doi.org/10.1016/j.teler.2023.100053.

[23] K. Alzboon, J. Al-Nihoud, W. Alsharafat, Novel network intrusion detection based on feature filtering using FLAME and new cuckoo selection in a genetic algorithm, Appl. Sci. (Switzerland) 13 (23) (Dec. 2023), https://doi.org/10.3390/app132312755.

[24] R. Ben Said, Z. Sabir, I. Askerzade, CNN-BiLSTM: A hybrid deep learning approach for network intrusion detection system in software-defined networking with hybrid feature selection, IEEe Access. 11 (2023) 138732–138747, https://doi.org/10.1109/ACCESS.2023.3340142.

[25] J. Zhang, X. Zhang, Z. Liu, F. Fu, Y. Jiao, F. Xu, A network intrusion detection model based on BiLSTM with multi-head attention mechanism, Electronics (Switzerland) 12 (19) (Oct. 2023), https://doi.org/10.3390/electronics12194170.

[26] T. Hou, H. Xing, X. Liang, X. Su, Z. Wang, A Marine hydrographic station networks intrusion detection method based on LCVAE and CNN-BiLSTM, J. Mar. Sci. Eng. 11 (1) (Jan. 2023), https://doi.org/10.3390/jmse11010221.

[27] A.M. Ali, F. Alqurashi, F.J. Alsolami, S. Qaiyum, A double-layer indemnity enhancement using LSTM and HASH function technique for intrusion detection system, Mathematics 11 (18) (Sep. 2023), https://doi.org/10.3390/math11183894.

[28] H. Jiang, S. Ji, G. He, X. Li, Network traffic anomaly detection model based on feature reduction and bidirectional LSTM neural Network optimization, Sci. Program. 2023 (Nov. 2023) 1–18, https://doi.org/10.1155/2023/2989533.
[29] S. Yaras and M. Dener, "IoT-based intrusion detection system using new hybrid deep learning algorithm," 2024, doi: 10.3390/electronics.
[30] T. Althiyabi, I. Ahmad, M.O. Alassafi, Enhancing IoT security: A few-shot learning approach for intrusion detection, Mathematics 12 (7) (Apr. 2024), https://doi.org/10.3390/math12071055.
[31] J. Bo, K. Chen, S. Li, P. Gao, Boosting few-shot network intrusion detection with adaptive feature fusion mechanism, Electronics (Switzerland) 13 (22) (Nov. 2024), https://doi.org/10.3390/electronics13224560.
[32] A. Touré, Y. Imine, A. Semnont, T. Delot, A. Gallais, A framework for detecting zero-day exploits in network flows, Comput. Netw. 248 (Jun. 2024), https://doi.org/10.1016/j.comnet.2024.110476.
[33] S.S.N. Chintapalli, S.P. Singh, J. Frnda, P. Bidare Divakarachari, V.L. Sarraju, P. Falkowski-Gilski, OOA-modified Bi-LSTM network: an effective intrusion detection framework for IoT systems, Heliyon. 10 (8) (Apr. 2024), https://doi.org/10.1016/j.heliyon.2024.e29410.
[34] Y. Guan, M. Noferesti, N. Ezzati-Jivan, A two-tiered framework for anomaly classification in IoT networks utilizing CNN-BiLSTM model[Formula presented], Softw. Impacts. 20 (May 2024), https://doi.org/10.1016/j.simpa.2024.100646.
[35] C. Zhang, J. Li, N. Wang, D. Zhang, Research on intrusion detection method based on Transformer and CNN-BiLSTM in Internet of things, Sensors 25 (9) (May 2025), https://doi.org/10.3390/s25092725.
[36] A. Alabbadi, F. Bajaber, An intrusion detection system over the IoT data streams using eXplainable artificial intelligence (XAI), Sensors 25 (3) (Feb. 2025), https://doi.org/10.3390/s25030847.
[37] F. Alhayan, M.K. Saeed, R. Allafi, M. Abdullah, A. Subahi, N.A. Alghanmi, H. Alkhudhayr, Hybrid deep learning models with spotted hyena optimization for cloud computing enabled intrusion detection system, J. Radiat. Res. Appl. Sci. 18 (2) (2025) 101523.
[38] M.V. Duc, P.M. Dang, T.T. Phuong, T.D. Truong, V. Hai, N.H. Thanh, Detecting emerging DGA malware in federated environments via variational autoencoder-based clustering and resource-aware client selection, Future Internet. 17 (7) (Jul. 2025) 299, https://doi.org/10.3390/fi17070299.
[39] S. Natha, F. Ahmed, M. Siraj, M. Lagari, M. Altamimi, A.A. Chandio, Deep BiLSTM attention model for spatial and temporal anomaly detection in video surveillance, Sensors 25 (1) (Jan. 2025), https://doi.org/10.3390/s25010251.
[40] S. Alsaleh, M.E.B. Menai, S. Al-Ahmadi, A heterogeneity-aware semi-decentralized model for a lightweight intrusion detection system for IoT networks based on federated learning and BiLSTM, Sensors 25 (4) (Feb. 2025), https://doi.org/10.3390/s25041039.
[41] V.Z. Mohale, I.C. Obagbuwa, Evaluating machine learning-based intrusion detection systems with explainable AI: enhancing transparency and interpretability, Front. Comput. Sci. 7 (2025), https://doi.org/10.3389/fcomp.2025.1520741.
[42] M. Rabbani, et al., Device identification and anomaly detection in IoT environments, IEEe Internet. Things. J. 12 (10) (2025) 13625–13643, https://doi.org/10.1109/JIOT.2024.3522863.
[43] G. Black, K. Fronczyk, W. Arliss, R. Allen, Descriptor: firewall attack detections and extractions (FADE), IEEE Data Descrip. 2 (May 2025) 163–172, https://doi.org/10.1109/ieeedata.2025.3572866.
[44] M.S. Korium, M. Saber, A. Beattie, A. Narayanan, S. Sahoo, P.H.J. Nardelli, Intrusion detection system for cyberattacks in the Internet of vehicles environment, Ad. Hoc. Netw. 153 (Feb. 2024), https://doi.org/10.1016/j.adhoc.2023.103330.
[45] L.B.V de Amorim, G.D.C. Cavalcanti, R.M.O. Cruz, The choice of scaling technique matters for classification performance, Appl. Soft. Comput. 133 (2023) 109924, https://doi.org/10.1016/j.asoc.2022.109924.
[46] A. Demircioğlu, The effect of feature normalization methods in radiomics, Insights. ImAging 15 (1) (Dec. 2024), https://doi.org/10.1186/s13244-023-01575-7.
[47] A. Kumar, R. Radhakrishnan, M. Sumithra, P. Kaliyaperumal, B. Balusamy, F. Benedetto, A scalable hybrid autoencoder–extreme learning machine framework for adaptive intrusion detection in high-dimensional networks, Future Internet. 17 (5) (May 2025) 221, https://doi.org/10.3390/fi17050221.
[48] B.Y. An, J.H. Yang, S. Kim, T. Kim, Malware detection using dual Siamese network model, CMES - Comput. Model. Eng. Sci. 141 (1) (2024) 563–584, https://doi.org/10.32604/cmes.2024.052403.
[49] Y. Xiao, Y. Feng, K. Sakurai, An efficient detection mechanism of network intrusions in IoT environments using autoencoder and data partitioning, Computers 13 (10) (Oct. 2024), https://doi.org/10.3390/computers13100269.
[50] K.A. Alaghbari, H.S. Lim, M.H.M. Saad, Y.S. Yong, Deep autoencoder-based integrated model for anomaly detection and efficient feature extraction in IoT networks, Internet Things 4 (3) (Sep. 2023) 345–365, https://doi.org/10.3390/iot4030016.
[51] T. Patel, S.S. Iyer, SiaDNN: Siamese deep neural network for anomaly detection in user behavior, Knowl. Based. Syst. 324 (2025) 113769, https://doi.org/10.1016/j.knosys.2025.113769.
[52] M. Sarhan, S. Layeghy, M. Gallagher, M. Portmann, From zero-shot machine learning to zero-day attack detection, Int. J. Inf. Secur. 22 (4) (Aug. 2023) 947–959, https://doi.org/10.1007/s10207-023-00676-0.
[53] K. Berahmand, F. Daneshfar, E.S. Salehi, Y. Li, Y. Xu, Autoencoders and their applications in machine learning: a survey, Artif. Intell. Rev. 57 (2) (Feb. 2024), https://doi.org/10.1007/s10462-023-10662-6.
[54] B.A. Manjunatha, K.A. Shastry, E. Naresh, P.K. Pareek, K.T. Reddy, A network intrusion detection framework on sparse deep denoising auto-encoder for dimensionality reduction, Soft. comput. 28 (5) (Mar. 2024) 4503–4517, https://doi.org/10.1007/s00500-023-09408-x.
[55] N. Latif, W. Ma, H.B. Ahmad, Advancements in securing federated learning with IDS: a comprehensive review of neural networks and feature engineering techniques for malicious client detection, Artif. Intell. Rev. 58 (3) (Mar. 2025), https://doi.org/10.1007/s10462-024-11082-w.
[56] A.A. Wani, Comprehensive review of dimensionality reduction algorithms: challenges, limitations, and innovative solutions, PeerJ. Comput. Sci. 11 (Jul. 2025) e3025, https://doi.org/10.7717/peerj-cs.3025.
[57] T.S. Lakshmi, M. Govindarajan, A. Srinivasulu, Embedding and Siamese deep neural network-based malware detection in Internet of Things, Int. J. Pervas. Comput. Commun. 21 (1) (Jan. 2025) 14–25, https://doi.org/10.1108/IJPCC-06-2022-0236.
[58] W. Dai, X. Li, W. Ji, S. He, Network intrusion detection method based on CNN-BiLSTM-attention model, IEEe Access. 12 (2024) 53099–53111, https://doi.org/10.1109/ACCESS.2024.3384528.
[59] Y. Li, G. Guo, J. Shi, R. Yang, S. Shen, Q. Li, J. Luo, A versatile framework for attributed network clustering via K-nearest neighbor augmentation, The VLDB Journal 33 (6) (2024) 1913–1943.
[60] T.B. Ogunseyi, G. Thiyagarajan, An explainable LSTM-based intrusion detection system optimized by Firefly algorithm for IoT networks, Sensors 25 (7) (Apr. 2025), https://doi.org/10.3390/s25072288.
[61] S. Subudhi, S. Panigrahi, Application of OPTICS and ensemble learning for database intrusion detection, J. King Saud Univ. - Comput. Inf. Sci. 34 (3) (Mar. 2022) 972–981, https://doi.org/10.1016/j.jksuci.2019.05.001.
[62] P. Artioli, A. Maci, A. Magrì, A comprehensive investigation of clustering algorithms for user and entity behavior analytics, Front. Big. Data 7 (2024), https://doi.org/10.3389/fdata.2024.1375818.