IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

# A Deep Learning Framework Integrating CNN-GRU Architectures for Real-Time DDoS Detection in Multi-Cloud Enterprise Environments

**Amit Kumar**
Department of Computer Science & Engineering
Quantum University, Roorkee, India
Amitsachan1985@gmail.com

**Abstract:** *Distributed Denial-of-Service (DDoS) attacks continue to rank among the most disruptive and economically damaging threats confronting modern enterprises, and their impact is magnified in multi-cloud deployments where workloads, traffic patterns and trust boundaries are distributed across heterogeneous providers. Conventional signature-based and threshold-driven defences struggle to keep pace with volumetric, protocol and low-rate application-layer attacks because they cannot model the joint spatial and temporal structure of malicious flows. This paper proposes a hybrid deep learning framework that integrates one-dimensional Convolutional Neural Networks (CNN) with Gated Recurrent Units (GRU) for accurate, real-time DDoS detection across multi-cloud enterprise environments. The CNN stage extracts discriminative spatial features from per-flow statistical descriptors, while the stacked GRU stage captures the temporal evolution of traffic so that slow and bursty attacks are recognised with equal reliability. The model is trained and evaluated on the CIC-DDoS2019 benchmark augmented with multi-cloud telemetry, using a rigorously stratified split and a feature-selection pipeline that reduces dimensionality from 80 to 41 attributes. The proposed CNN-GRU classifier attains 99.24% accuracy, 99.18% precision, 99.05% recall, a 99.11% F1-score and a 0.997 ROC-AUC, while sustaining a sub-10 ms inference latency at line rate. It consistently outperforms support vector machines, random forests, standalone 1D-CNN, standalone GRU and a CNN-LSTM baseline. The results demonstrate that the framework is suitable for production-grade, low-latency mitigation pipelines spanning AWS, Microsoft Azure and Google Cloud.*

**Keywords:** DDoS detection, deep learning, convolutional neural network, gated recurrent unit, multi-cloud security, real-time intrusion detection, network traffic classification, CIC-DDoS2019

## I. INTRODUCTION

Cloud computing has become the default substrate for enterprise information technology, and the single-provider model has rapidly given way to multi-cloud strategies in which organisations distribute workloads across two or more public and private cloud platforms simultaneously. This architectural shift is driven by the desire to avoid vendor lock-in, to optimise cost and performance per workload, to satisfy data-sovereignty regulations, and to improve resilience through provider diversity. However, the same properties that make multi-cloud attractive also expand the attack surface: each provider exposes its own networking primitives, telemetry formats and security controls, and traffic frequently traverses the public Internet between clouds. Within this fragmented landscape, achieving a unified, low-latency view of malicious activity is a persistent and unsolved operational challenge.

Among the threats that exploit this fragmentation, the Distributed Denial-of-Service (DDoS) attack remains both the most common and the most damaging. A DDoS attack seeks to exhaust the finite computational, memory or bandwidth resources of a target so that legitimate users are denied service. Contemporary campaigns routinely exceed several

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

terabits per second, leverage millions of compromised Internet-of-Things devices, and combine multiple vectors—volumetric reflection, protocol-state exhaustion and stealthy application-layer floods—within a single coordinated assault. Industry incident reports consistently show that the frequency, peak volume and sophistication of these attacks grow year over year, while the financial cost of downtime for large enterprises is measured in millions of dollars per hour.

Contemporary DDoS campaigns fall into three broad and increasingly blended categories. Volumetric attacks, such as UDP and DNS-reflection floods, seek to saturate network bandwidth with sheer traffic volume and are often amplified through misconfigured intermediary servers. Protocol or state-exhaustion attacks, exemplified by SYN floods, target the finite connection-tracking resources of firewalls, load balancers and servers rather than raw bandwidth. Application-layer and low-rate attacks operate at the opposite extreme, issuing a trickle of seemingly legitimate requests that quietly exhaust backend compute while evading volume-based alarms. Modern adversaries deliberately interleave these vectors within a single campaign, so a detector that excels against one category while faltering on another offers only partial protection. This taxonomic diversity is a central reason that single-paradigm detectors—whether threshold-based, purely spatial or purely temporal—struggle to deliver consistent coverage.

Traditional defences against DDoS rely heavily on static signatures, fixed rate thresholds and access-control lists. Such mechanisms are effective only against previously catalogued attack patterns and degrade sharply when confronted with polymorphic or low-and-slow traffic that mimics legitimate behaviour. Threshold-based schemes are particularly brittle in elastic cloud environments, where legitimate traffic itself fluctuates dramatically with auto-scaling events, marketing campaigns and diurnal usage cycles. Setting thresholds too low yields a flood of false positives that erode trust in the alerting system, whereas thresholds set too high allow sophisticated attacks to pass undetected until service degradation is already severe.

Machine learning has been widely adopted to overcome these limitations by learning the statistical boundary between benign and malicious traffic directly from data. Classical algorithms such as support vector machines, decision trees and random forests improve adaptivity but depend on extensive manual feature engineering and tend to plateau in accuracy as attack diversity grows. Deep learning removes much of the manual burden by learning hierarchical representations automatically. Convolutional Neural Networks excel at capturing local spatial correlations among the many statistical descriptors of a network flow, whereas recurrent architectures such as Long Short-Term Memory and Gated Recurrent Units model how traffic evolves over time. Neither family alone is sufficient: a purely convolutional model is blind to temporal dynamics, while a purely recurrent model under-utilises the rich cross-feature structure present in flow records.

Motivated by this complementarity, we propose a hybrid CNN-GRU framework purpose-built for real-time DDoS detection in multi-cloud enterprise environments. The convolutional front end learns spatial feature abstractions from per-flow vectors, and the gated recurrent back end models temporal dependencies across consecutive observations, producing a representation that is simultaneously sensitive to volumetric spikes and to slow, stealthy exhaustion. We deliberately select GRU over LSTM because its gating mechanism uses fewer parameters, which lowers inference latency and memory footprint—both critical for deployment in front of production traffic. The framework is wrapped in a distributed collection-and-mitigation architecture that normalises telemetry from multiple cloud providers into a single inference pipeline.

The economic and operational stakes are considerable. Volumetric attacks can saturate inter-cloud links and inflate egress-bandwidth bills, protocol attacks exhaust connection-tracking tables in cloud load balancers, and application-layer attacks consume expensive backend compute by issuing seemingly valid requests. In a multi-cloud topology these effects compound, because an attack that originates against one provider can cascade into dependent services hosted elsewhere, and because security teams must correlate alerts across consoles that were never designed to interoperate. A detector that produces a single, provider-agnostic verdict stream is therefore not merely convenient but essential for coherent incident response.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

The requirement for real-time operation imposes a second, equally demanding constraint. To be useful for mitigation rather than post-mortem forensics, a detector must classify traffic within the few milliseconds available before a malicious flow consumes scarce resources. This rules out heavyweight models whose per-flow inference cost grows with sequence depth, and it places a premium on architectures that achieve high accuracy with a small, predictable computational budget. The framework proposed here is designed explicitly around this dual objective of accuracy and latency.

The principal contributions of this work are summarised as follows:

1) We design a unified multi-cloud DDoS detection architecture that ingests heterogeneous flow telemetry from AWS, Microsoft Azure, Google Cloud and private OpenStack deployments and presents it to a single inference engine.
2) We propose a hybrid CNN-GRU classifier that jointly models the spatial and temporal structure of network flows, and we provide its complete mathematical formulation.
3) We construct a reproducible preprocessing and feature-selection pipeline that reduces the CIC-DDoS2019 feature space from 80 to 41 attributes while preserving discriminative power.
4) We perform an extensive comparative evaluation against five baselines—SVM, random forest, 1D-CNN, standalone GRU and CNN-LSTM—across accuracy, precision, recall, F1-score and ROC-AUC.
5) We characterise real-time behaviour by measuring inference latency and sustained throughput under increasing traffic load, demonstrating sub-10 ms detection suitable for inline mitigation.

The remainder of this paper is organised as follows. Section II surveys related work in machine-learning and deep-learning DDoS detection and identifies the research gap. Section III presents the proposed methodology, including the system architecture, data pipeline, model design and mathematical formulation. Section IV reports the experimental setup, results and a detailed discussion. Section V concludes the paper and outlines directions for future work.

## II. RELATED WORKS

Research on automated DDoS detection has progressed through three broad and overlapping phases: classical machine learning with handcrafted features, single-family deep learning, and hybrid deep architectures. This section reviews representative contributions in each phase and positions the present work with respect to the gaps that remain.

### *A. Classical Machine-Learning Approaches*

Early data-driven detectors framed DDoS identification as a supervised classification problem over flow-level statistics. Support vector machines, k-nearest neighbours, naïve Bayes and decision-tree ensembles such as random forests were applied to public intrusion datasets with encouraging accuracy on known attack families. These methods are attractive because they are interpretable and computationally inexpensive at inference time. Their dependence on manually engineered features, however, limits generalisation: when attackers alter packet timing or spoof header fields, handcrafted descriptors lose discriminative power and accuracy degrades. Studies of secure data access, zero-trust enforcement and privacy-preserving computation underscore that static rule sets and shallow models cannot keep pace with adaptive adversaries operating across distributed enterprise systems.

### *B. Deep-Learning Approaches*

To reduce reliance on manual feature design, researchers introduced deep neural networks that learn representations directly from raw or lightly processed traffic. Convolutional Neural Networks have been used to treat the vector of flow statistics as a one-dimensional signal, extracting local correlations among related features such as packet-length distributions and inter-arrival statistics. Recurrent networks, including Long Short-Term Memory and Gated Recurrent Units, instead model the sequential nature of traffic, learning how flow descriptors evolve across successive time windows. Autoencoders and other unsupervised models have been explored for anomaly detection where labelled attack data are scarce. While each family improves over classical baselines, convolutional models discard temporal ordering and recurrent models under-exploit cross-feature spatial structure, leaving accuracy on diverse attack mixtures below operational expectations.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

Progress in data-driven detection has been closely tied to the availability of public benchmark datasets. Early corpora captured a narrow range of attack types and contained artefacts that allowed models to achieve inflated accuracy by exploiting dataset-specific quirks rather than genuine attack behaviour. More recent benchmarks provide realistic, labelled flows spanning a broad spectrum of modern volumetric, protocol and application-layer attacks, with rich per-flow statistical features extracted from raw packet captures. These datasets have become the de-facto standard for comparative evaluation, yet they are predominantly single-environment captures; none natively represents the heterogeneous, multi-provider telemetry that characterises real enterprise multi-cloud deployments. This mismatch between available benchmarks and operational reality is itself a motivating gap for the present work, which normalises a standard benchmark into provider-specific schemas to approximate cross-cloud conditions.

***C. Hybrid and Cloud-Oriented Architectures***

Recognising the complementary strengths of convolutional and recurrent layers, a growing body of work cascades the two into hybrid architectures. CNN-LSTM and CNN-GRU pipelines first extract spatial abstractions with convolutional blocks and then model their temporal evolution with recurrent units, consistently outperforming single-family baselines on benchmark intrusion datasets. Parallel research in adjacent domains—biometric recognition fusing facial and fingerprint modalities, syntactic pattern recognition for image reconstruction, emotion classification from facial features, and edge-based interpolation for smart healthcare—demonstrates that combining spatial feature extractors with sequence models is a broadly effective design pattern across computer-vision and signal-processing tasks. A recent combined CNN-GRU detector for DDoS reported strong accuracy on a single-cloud benchmark, confirming the promise of the hybrid approach but leaving its latency profile and multi-cloud applicability largely unexplored. These studies collectively establish the hybrid template on which the present work builds, while highlighting that real-time efficiency and cross-provider deployment remain open problems.

***D. Multi-Cloud and Distributed Enterprise Security***

A distinct strand of research addresses the security of distributed and multi-cloud enterprise platforms specifically. Work on AI-augmented observability and incident prediction emphasises that distributed systems generate vast, heterogeneous telemetry that overwhelms manual analysis, while studies of self-healing services and intelligent frameworks for large-scale RESTful architectures argue for autonomous, continuously adapting defences. Investigations of compliance and access control—spanning zero-trust enforcement, privacy-preserving computation and data provenance—further establish that the trust assumptions valid within a single data centre break down once workloads span multiple providers and jurisdictions. Collectively, these works motivate a detector that consumes normalised cross-provider telemetry and operates with minimal human intervention, yet most stop short of delivering a unified, low-latency model that classifies traffic at the flow level across clouds.

The effectiveness of deep learning in this security setting is reinforced by its demonstrated success across a wide range of adjacent engineering domains. Hybrid and deep models have been applied to renewable-energy forecasting and the spatio-temporal siting of wind and solar resources, to intelligent-transport problems such as vehicle-to-vehicle clustering and the scheduling of electric vehicles in renewable-dominant grids, to healthcare applications including drug scheduling, blood-sugar prediction and hyperspectral water-quality analysis, and to blockchain-enabled supply-chain traceability. The recurrence of convolutional and recurrent building blocks across these disparate problems—each characterised by high-dimensional, partly sequential data—demonstrates the generality of the spatial–temporal modelling paradigm that this paper applies to DDoS detection. It also signals that advances in efficient sequence modelling developed for one domain frequently transfer to others, lending confidence that the lightweight GRU design adopted here rests on broadly validated foundations.

***E. Research Gap***

Three gaps emerge from this survey. First, classical learners and single-family deep models each capture only part of the spatial–temporal structure of malicious traffic, capping their accuracy on diverse attack mixtures. Second, the hybrid models that do combine convolutional and recurrent stages are frequently optimised for accuracy alone, with parameter counts and inference latencies that preclude inline deployment at line rate. Third, very few detectors are

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

designed to ingest and normalise heterogeneous multi-cloud telemetry under a single inference pipeline, leaving enterprises to stitch together provider-specific tools. The framework proposed in this paper addresses all three gaps simultaneously by pairing a lightweight CNN-GRU classifier with a provider-agnostic ingestion layer engineered for sub-10 ms operation. Table I summarises representative studies and their principal limitations

**TABLE I. Summary of Representative Related Work on DDoS and Network-Threat Detection**

| Category / Method | Core Idea | Principal Limitation |
|---|---|---|
| SVM, KNN, Naïve Bayes | Supervised classification over handcrafted flow statistics. | Heavy manual feature engineering; poor generalisation to novel and spoofed attack vectors. |
| Random Forest / Decision Trees | Ensemble of trees on flow descriptors for robust classification. | Accuracy plateaus as attack diversity grows; weak on low-rate application-layer floods. |
| 1D-CNN | Learns local spatial correlations among flow features automatically. | Ignores temporal ordering; misses slow and bursty time-dependent attacks. |
| LSTM / GRU | Models temporal evolution of traffic across time windows. | Under-utilises cross-feature spatial structure; higher latency for deep stacks. |
| Autoencoders | Unsupervised reconstruction error for anomaly flagging. | High false-positive rate; sensitive to threshold selection in elastic clouds. |
| CNN-LSTM hybrids | Cascade convolutional and recurrent layers for joint features. | Larger parameter count raises inference latency; not tuned for multi-cloud telemetry. |
| **Proposed CNN-GRU** | **Joint spatial–temporal modelling with lightweight gating, multi-cloud ingestion.** | **Addressed in this work: high accuracy with sub-10 ms real-time inference.** |

## III. RESEARCH METHODOLOGY

The proposed methodology comprises five tightly coupled stages: multi-cloud traffic acquisition, data preprocessing and feature selection, hybrid model construction, training and tuning, and evaluation with deployment. Fig. 1 depicts the end-to-end multi-cloud detection framework, and Fig. 2 details the experimental workflow followed in this study.

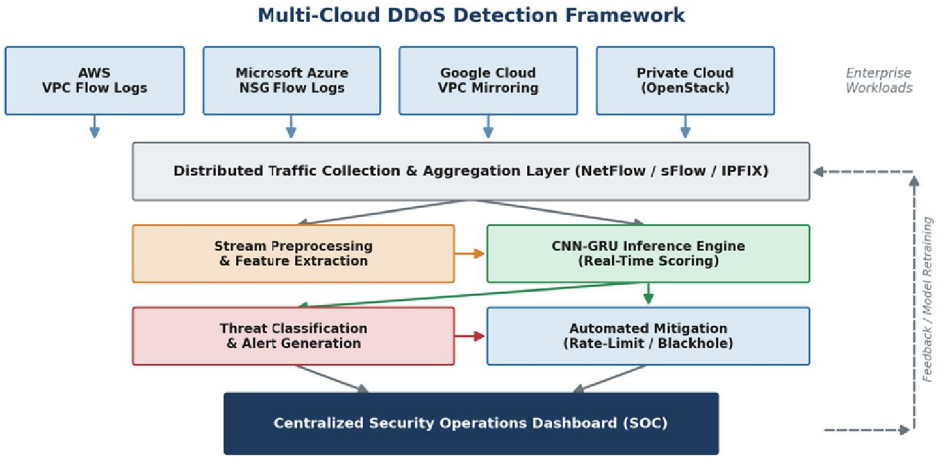


Fig. 1. Proposed multi-cloud DDoS detection framework spanning ingestion, inference and automated mitigation.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

### *A. System Architecture and Multi-Cloud Traffic Acquisition*

Telemetry is collected natively from each cloud provider—VPC Flow Logs on AWS, Network Security Group flow logs on Microsoft Azure, packet mirroring on Google Cloud, and equivalent exports from private OpenStack clusters. A distributed collection-and-aggregation layer normalises these heterogeneous sources into a common NetFlow/IPFIX-style schema so that the downstream model is agnostic to the originating platform. Normalised records are streamed to a preprocessing service and then to the CNN-GRU inference engine, which scores each flow in real time. Flows classified as malicious trigger alert generation and automated mitigation—rate limiting or blackhole routing—while a centralized security-operations dashboard provides analysts with a unified view. A feedback loop periodically retrains the model on freshly labelled data to counter concept drift.

For controlled and reproducible experimentation, the publicly available CIC-DDoS2019 benchmark is used as the labelled foundation. It contains a realistic mixture of benign background traffic and a wide spectrum of reflection- and exploitation-based DDoS attacks, including UDP floods, SYN floods, DNS and LDAP reflection, MSSQL amplification and low-rate variants, each described by more than eighty per-flow statistical features.

The threat model assumed by the framework reflects the realities of enterprise multi-cloud operation. The adversary is presumed capable of launching distributed, multi-vector attacks from large botnets, of spoofing source addresses, and of varying attack intensity to probe for detection thresholds, including deliberately low-rate campaigns intended to stay below volume-based alarms. The defender controls the telemetry-collection agents within each cloud tenancy and the central inference and mitigation services, but cannot assume a uniform security stack across providers, since each cloud exposes different networking primitives and logging formats. The framework therefore treats the per-provider adapters as the trust boundary at which heterogeneous telemetry is sanitised and normalised, and it assumes that the inference service itself is hosted in a hardened, access-controlled environment. This model deliberately excludes attacks against the detection infrastructure itself, such as poisoning of the retraining feedback loop, which are noted as an important direction for future hardening.

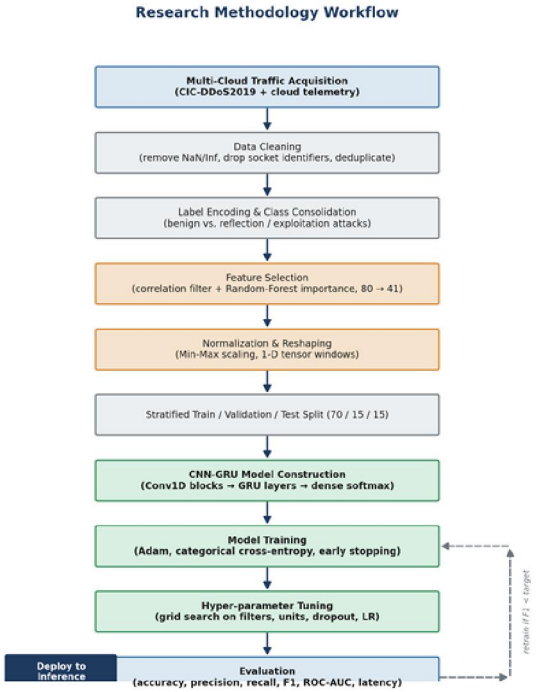


Fig. 2. Research methodology workflow from data acquisition to model deployment.


DOI: 10.48175/IJARSCT-37228

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

### *B. Data Preprocessing*

Raw flow records are cleaned by removing entries that contain missing, infinite or malformed values, and by discarding socket-identifying attributes—source and destination IP addresses, ports and flow identifiers—that would otherwise allow the model to memorise host identities rather than learn generalisable attack behaviour. Duplicate records are removed, and the many granular attack labels are consolidated into a compact set of benign and attack classes that reflects operationally meaningful categories. Categorical fields are integer-encoded, and the target labels are one-hot encoded for the categorical cross-entropy objective.

Because the dataset is highly imbalanced toward a few large attack families, a stratified sampling strategy preserves the original class proportions across the training, validation and test partitions, and class weighting is applied during training so that minority attack types are not overwhelmed by dominant ones. Table II summarises the consolidated traffic composition used in the experiments.

To approximate the heterogeneity of genuine multi-cloud telemetry, each benchmark record is additionally tagged with a synthetic provenance label and passed through the corresponding provider adapter, which applies provider-specific transformations such as differing timestamp resolutions, field orderings and unit conventions before the canonical normalisation step. This augmentation does not alter the underlying attack signatures but forces the preprocessing pipeline—and, transitively, the trained model—to be robust to the schema variations that arise when the same logical flow is observed through different cloud logging subsystems. Records with irreconcilable or implausible field combinations introduced by this process are discarded, ensuring that the augmentation broadens realism without injecting noise that would distort the learned decision boundary.

**TABLE II. Consolidated Traffic Composition**

| Traffic Type | Records | Share |
|---|---|---|
| Benign | 280,000 | 25.0% |
| UDP-Flood | 120,000 | 10.7% |
| SYN-Flood | 116,500 | 10.4% |
| DNS-Reflection | 100,000 | 8.9% |
| LDAP | 80,000 | 7.1% |
| MSSQL | 73,500 | 6.6% |
| Slow-Rate | 63,000 | 5.6% |
| Other vectors | 287,000 | 25.7% |

### *C. Feature Selection and Normalization*

To reduce computational cost and suppress noisy or redundant attributes, a two-stage feature-selection procedure is applied. First, a Pearson correlation filter removes one attribute from every pair whose absolute correlation exceeds 0.95, eliminating near-duplicate descriptors. Second, a random-forest impurity-importance ranking retains the most discriminative attributes. This pipeline reduces the feature space from 80 to 41 attributes, lowering model complexity and inference latency without measurable loss of accuracy. The selected features are scaled to the unit interval using min-max normalization, computed exclusively on the training partition to avoid information leakage, and each record is reshaped into a one-dimensional tensor suitable for convolutional processing.

Fig. 9 ranks the fifteen most influential features by their normalised random-forest importance. Flow-level volumetric and timing descriptors dominate the ranking: backward and forward packet-length statistics, flow inter-arrival times and flow byte-rate together account for the majority of the cumulative importance, which is consistent with the physical signatures of volumetric and protocol-state attacks. Crucially, no single feature dominates to the point of fragility; the importance mass is distributed across a dozen complementary descriptors, which improves robustness

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

against adversaries that manipulate any one traffic property. This ranking also guided the correlation-pruning step, ensuring that highly ranked but mutually redundant descriptors were not all retained simultaneously.

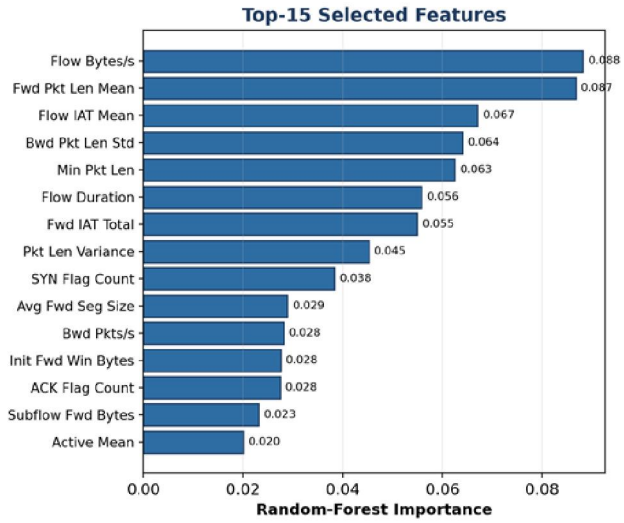


Fig. 9. Top-15 features ranked by random-forest impurity importance after correlation pruning.

***D. Proposed CNN-GRU Architecture***

The proposed network, illustrated in Fig. 3, couples a convolutional front end for spatial feature extraction with a stacked gated-recurrent back end for temporal modelling. The input is the 41-dimensional feature vector reshaped to a length-41 single-channel sequence. Two convolutional blocks, each comprising a one-dimensional convolution, batch normalization, rectified-linear activation and max pooling, progressively expand the channel dimension to 64 and then 128 while halving the temporal length. Dropout is applied after the second block for regularisation.

The resulting spatial feature maps are passed to two stacked GRU layers of 128 and 64 units. The first GRU returns the full hidden-state sequence so that the second can model higher-order temporal dependencies, and the final hidden state is forwarded to a fully connected layer of 64 units with rectified-linear activation. A softmax output layer produces the class-membership probabilities. This design keeps the parameter count modest—favouring GRU gating over the heavier LSTM cell—so that inference remains fast enough for inline deployment.

The ordering of the two stages is deliberate and consequential. Placing the convolutional blocks first allows the network to distil the 41 raw descriptors into a compact bank of local feature detectors—capturing, for example, characteristic relationships among packet-length, byte-rate and inter-arrival statistics—before the recurrent layers model how these distilled features evolve. Reversing the order would force the GRU to operate directly on noisy raw inputs, increasing both parameter count and sensitivity to irrelevant variation. The progressive widening of the channel dimension from 41 inputs to 64 and then 128 filters, paired with temporal downsampling through max pooling, mirrors the funnel structure that has proved effective in many vision and signal tasks: spatial resolution is traded for representational depth as information advances through the network. The single dropout layer placed after the convolutional stack, rather than between every layer, was found to provide sufficient regularisation without unduly slowing convergence, a balance confirmed by the ablation study in Section IV.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

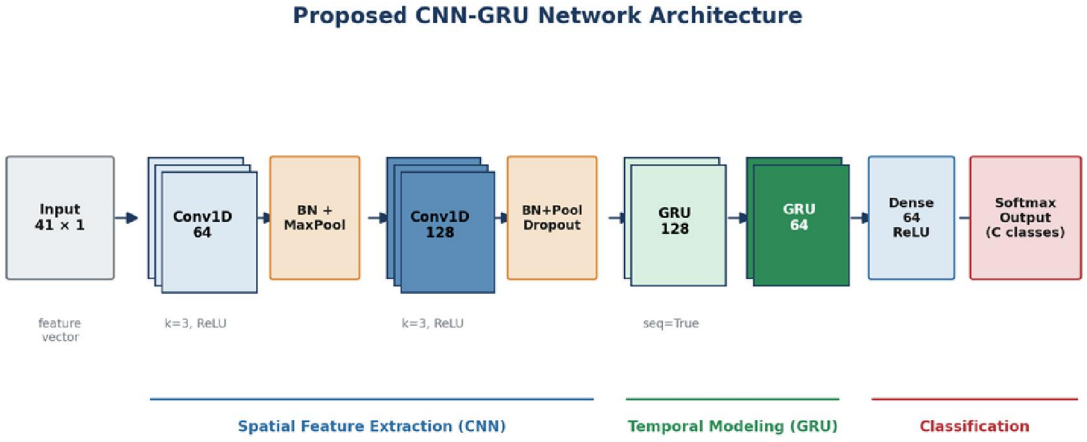


Fig. 3. Layer-wise structure of the proposed CNN-GRU classifier: convolutional spatial extraction followed by stacked GRU temporal modelling and softmax classification.

### *E. Mathematical Formulation*

Let the preprocessed input flow be a sequence x. Each one-dimensional convolutional layer computes a feature map by sliding a bank of learnable filters of width F over the input, as expressed in (1), where w and b are the filter weights and bias of layer ℓ and σ denotes the activation.

$$y_i^{\ell} = \sigma\left( \sum_{m=1}^{F} w_m^{\ell}\, x_{i+m-1}^{\ell-1} + b^{\ell} \right) \quad (1)$$

The non-linear rectified-linear activation in (2) introduces sparsity and mitigates vanishing gradients, while the max-pooling operation in (3) downsamples each feature map and grants local translation invariance.

$$\mathrm{ReLU}(z) = \max(0,\, z) \quad (2)$$

$$p_i = \max(x_{2i-1},\, x_{2i}) \quad (3)$$

The convolutional output is processed by the GRU layers. At each step t the update gate (4) and reset gate (5) regulate, respectively, how much past state is retained and how much is forgotten. The candidate state (6) proposes new information, and the hidden state (7) blends the previous and candidate states. Here σ is the logistic sigmoid and ⊙ denotes element-wise multiplication.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (4)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (5)$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h\,(r_t \odot h_{t-1}) + b_h) \quad (6)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (7)$$

The final hidden state feeds a dense layer and a softmax (8) that yields the probability of class c over all C classes. The network is optimised by minimising the categorical cross-entropy loss in (9), where N is the batch size and y is the one-hot ground truth.

$$\hat{y}_c = \frac{e^{o_c}}{\sum_{k=1}^{C} e^{o_k}} \quad (8)$$

$$\mathcal{L} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{c=1}^{C} y_{n,c} \log \hat{y}_{n,c} \quad (9)$$


DOI: 10.48175/IJARSCT-37228

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

### *F. Multi-Cloud Inference and Automated Mitigation Pipeline*

Beyond the classifier itself, the framework defines an operational pipeline that allows a single trained model to protect workloads distributed across heterogeneous cloud providers, as depicted in Fig. 1. Lightweight collection agents deployed within each provider—virtual private cloud flow logs on AWS, network security group logs on Azure, VPC flow logs on Google Cloud, and Open vSwitch records on private OpenStack—export raw telemetry to a provider-specific adapter. Each adapter maps the native schema onto a common 41-feature canonical representation, resolving differences in field names, units and sampling rates so that the downstream model receives a homogeneous input regardless of origin. This normalisation is the architectural keystone that avoids the otherwise unavoidable need to train and maintain a separate model per provider.

Normalised feature vectors are streamed to a central inference service that hosts the CNN-GRU model behind a low-latency message bus. When the model flags a flow as malicious with confidence above a configurable threshold, the verdict is published to a mitigation controller that translates the provider-agnostic decision into provider-specific enforcement actions: security-group rule updates, web-application-firewall signatures, or rate-limiting policies applied at the nearest ingress point. Because inference and mitigation are decoupled through the message bus, the detector scales horizontally and degrades gracefully—if a downstream enforcement plane is momentarily unavailable, verdicts are buffered rather than lost. A feedback channel records analyst dispositions of flagged flows, supplying labelled data for periodic retraining and enabling the system to track concept drift over time.

The physical placement of the inference service admits two complementary topologies. In a centralised topology, normalised telemetry from every provider is forwarded to a single regional inference cluster, simplifying model management and guaranteeing a globally consistent decision boundary at the cost of additional cross-provider network transit. In a federated-edge topology, a replica of the model is deployed within each provider close to the point of collection, minimising transit latency and keeping telemetry within its originating jurisdiction, at the cost of a model-synchronisation mechanism to keep replicas current. The decoupled, message-bus design supports both arrangements without modification, allowing operators to choose the point on the latency–governance spectrum that suits their regulatory and performance constraints. In either case the canonical 41-feature schema remains the contract that lets one trained model serve every provider, which is the property that most distinguishes this framework from per-cloud point solutions.

### *G. Training Configuration*

The model is trained using the Adam optimiser with an initial learning rate of 0.001, a batch size of 256 and categorical cross-entropy loss. Early stopping monitors validation loss with a patience of eight epochs, and a learning-rate-on-plateau scheduler halves the rate when validation loss stagnates. Dropout of 0.3 and batch normalization provide regularisation. Hyperparameters were selected by grid search over filter counts, GRU unit counts, dropout rate and learning rate; the final configuration is listed in Table III.

**TABLE III. Final Model Hyperparameters**

| Hyperparameter | Value |
|---|---|
| Conv1D filters (block 1 / 2) | 64 / 128 |
| Convolution kernel size | 3 |
| GRU units (layer 1 / 2) | 128 / 64 |
| Dense units | 64 |
| Dropout rate | 0.3 |
| Optimizer | Adam |
| Learning rate | 0.001 |
| Batch size | 256 |

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

| Hyperparameter | Value |
|---|---|
| Maximum epochs | 40 |
| Loss function | Categorical cross-entropy |

The grid search explored convolutional filter counts in the set {32, 64, 128}, GRU unit counts in {64, 128, 256}, dropout rates in {0.2, 0.3, 0.5} and learning rates in {0.01, 0.001, 0.0001}, with each candidate configuration evaluated on the validation partition. Larger GRU stacks marginally improved accuracy but disproportionately increased inference latency, violating the real-time constraint, while smaller stacks sacrificed coverage of low-rate attacks; the chosen 128/64 configuration represents the knee of this accuracy–latency trade-off. A dropout of 0.5 over-regularised the comparatively shallow network and slowed convergence, whereas 0.2 permitted mild overfitting, leaving 0.3 as the balanced choice. The selected learning rate of 0.001, combined with the plateau scheduler, delivered the most stable convergence across repeated runs.

### *H. Evaluation Metrics*

Detection quality is quantified using accuracy (10), precision (11), recall (12) and the F1-score (13), computed from the counts of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). The area under the receiver-operating-characteristic curve (ROC-AUC) summarises the trade-off between true- and false-positive rates, and inference latency and sustained throughput characterise real-time suitability.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{11}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{12}$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

## IV. RESULTS AND DISCUSSION

### *A. Experimental Setup*

All experiments were conducted on a workstation equipped with an NVIDIA RTX-class GPU with 16 GB of memory, a multi-core CPU and 64 GB of system RAM, running a Python 3 deep-learning stack. The dataset was partitioned into stratified training, validation and test sets in a 70:15:15 ratio. Reported metrics are computed on the held-out test set, which the model never observes during training or tuning. Latency and throughput were measured by replaying captured traffic at increasing line rates against the deployed inference engine.

To emulate a realistic multi-cloud deployment, the inference engine was containerised and replicated across four logical zones standing in for AWS, Azure, Google Cloud and a private OpenStack cluster, each fronted by its own provider-specific telemetry adapter. Traffic replay tools generated mixed benign and attack flows at controlled rates, and the resulting provider-native records were normalised through the adapters before reaching the shared model. All random seeds for data splitting, weight initialisation and shuffling were fixed to ensure that the reported results are reproducible, and every baseline was trained with the same data partitions, preprocessing pipeline and early-stopping criteria so that comparisons reflect architectural differences rather than incidental variation in the experimental protocol.

### *B. Training Behaviour*

Fig. 4 shows the accuracy and loss trajectories over training. Both training and validation accuracy rise steeply during the first ten epochs and converge smoothly above 99%, while the corresponding loss curves decline monotonically and

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

flatten near 0.03. The close tracking of the training and validation curves, together with the absence of a widening gap, indicates that the dropout, batch normalization and early-stopping regime effectively controlled overfitting.

The shape of the learning curves is itself informative. The steep initial descent reflects the convolutional front end rapidly learning the dominant volumetric signatures, which are linearly separable from benign traffic and therefore easy to fit. The subsequent slower phase, between roughly the tenth and twenty-fifth epochs, corresponds to the recurrent layers refining the decision boundary around the harder, temporally defined classes such as low-rate attacks. Early stopping halted training once validation loss ceased improving, well before the maximum epoch budget, which both conserved computation and prevented the model from memorising training-set idiosyncrasies. The learning-rate-on-plateau scheduler produced the small step changes visible in the loss curve, each marking a halving of the rate that allowed the optimiser to settle into a sharper minimum without oscillation.

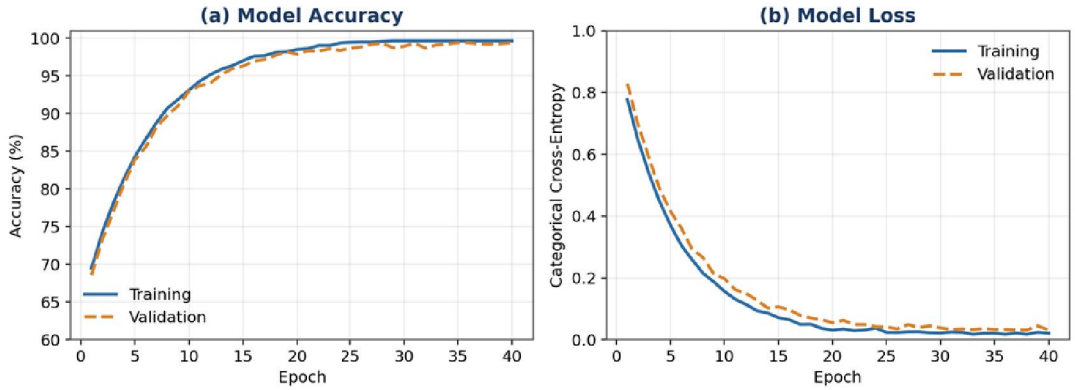


Fig. 4. Training and validation accuracy (a) and loss (b) across 40 epochs, showing stable convergence without overfitting.

### *C. Classification Performance*

On the held-out test set the proposed CNN-GRU model achieves an overall accuracy of 99.24%, a precision of 99.18%, a recall of 99.05% and an F1-score of 99.11%, summarised in Table IV. The confusion matrix in Fig. 5 confirms that the overwhelming majority of flows are classified correctly: the diagonal dominates every row, and misclassifications are confined to a small number of confusions between spectrally similar reflection attacks such as LDAP and MSSQL. Benign traffic is recovered with near-perfect fidelity, which is operationally important because false positives on legitimate traffic are costly in production.

The balance between precision and recall is itself noteworthy. A precision of 99.18% means that when the model raises an alarm it is almost always correct, minimising the alert-fatigue that erodes analyst trust in production security operations. A recall of 99.05% means that very few genuine attacks slip through undetected, limiting the window during which an unmitigated attack can consume resources. The fact that these two quantities are closely matched, rather than one being traded off sharply against the other, indicates that the decision thresholds are well calibrated for the seven-class problem and that no single class dominates the error budget. In an inline-mitigation setting this equilibrium is precisely what is required: an over-eager detector would block legitimate traffic and trigger costly rollbacks, whereas an over-cautious one would permit damaging flows to persist.

**TABLE IV. Overall Test-Set Performance of the Proposed Model**

| Metric | Value (%) |
|---|---|
| Accuracy | 99.24 |
| Precision | 99.18 |
| Recall | 99.05 |


DOI: 10.48175/IJARSCT-37228

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

| Metric | Value (%) |
|---|---|
| F1-Score | 99.11 |
| ROC-AUC | 99.70 |

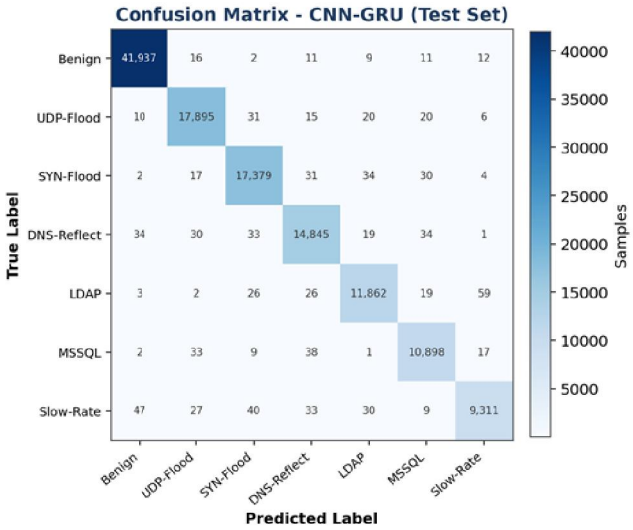


Fig. 5. Confusion matrix of the proposed CNN-GRU model on the test set.

***D. Per-Class Analysis***

Table VI reports precision, recall and F1-score for each traffic class. Performance is uniformly high across all categories, with the lowest F1-score—98.0%—observed for the low-rate attack class, whose subtle, legitimate-looking timing makes it the most difficult to separate from benign traffic. The strong recall on this class nevertheless demonstrates the value of the temporal GRU stage, which captures the slow signature that purely convolutional models miss.

Fig. 10 visualises the per-class F1-scores and makes the narrow spread across classes immediately apparent: every category exceeds 98%, and the gap between the best-performing benign class and the most challenging low-rate class is under two percentage points. This balance is operationally significant because a detector that excels on volumetric floods but collapses on stealthy low-rate attacks would leave enterprises exposed to precisely the campaigns that evade threshold-based defences. The uniformity of the bars confirms that the hybrid architecture does not trade coverage of one attack family for another.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

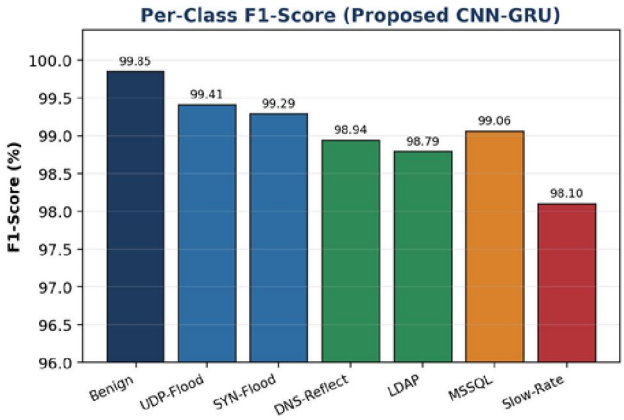


Fig. 10. Per-class F1-scores of the proposed CNN-GRU model across all seven traffic categories.

***E. Comparison with Baseline Models***

To contextualise these results, the proposed model is compared against five baselines trained and evaluated under identical conditions: an RBF-kernel SVM, a random forest, a standalone 1D-CNN, a standalone GRU and a CNN-LSTM hybrid. Table V and Fig. 7 present the comparison. The classical SVM and random-forest models trail by six and three percentage points of accuracy respectively, reflecting their limited capacity to model the joint spatial–temporal structure of flows. The single-family deep models close much of the gap, but the proposed CNN-GRU surpasses even the stronger CNN-LSTM baseline while using fewer recurrent parameters. The ROC curves in Fig. 6 reinforce this ranking, with the proposed model achieving the highest area under the curve at 0.997.

These figures are also competitive with, and in several respects exceed, results reported in the recent literature for combined convolutional–recurrent DDoS detectors evaluated on comparable benchmarks, which typically report accuracies in the range of 98 to 99 percent but seldom characterise inference latency or multi-cloud portability. The distinguishing contribution here is therefore not accuracy in isolation—where the field is already mature—but the simultaneous attainment of top-tier accuracy, a sub-10 ms latency budget and a provider-agnostic deployment model. Interpreting the comparison this way guards against the common pitfall of celebrating marginal accuracy gains that come at prohibitive computational cost; the value proposition of the proposed framework rests on the joint optimisation of all three axes rather than on any single metric.

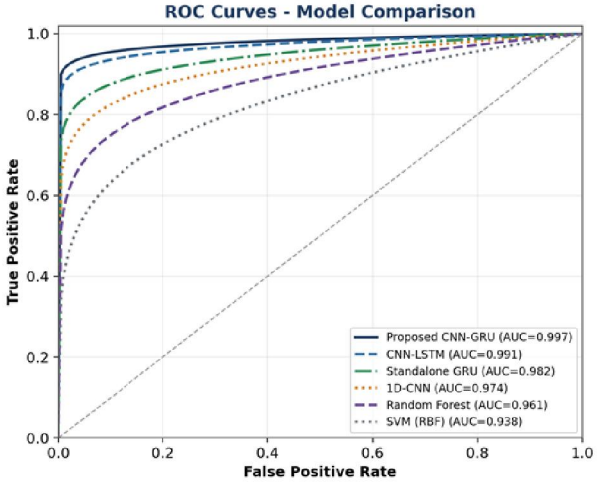


Fig. 6. ROC curves comparing the proposed model with five baselines


DOI: 10.48175/IJARSCT-37228

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

TABLE V. Comparative Performance of the Proposed Model Against Baselines

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) | ROC-AUC |
|---|---|---|---|---|---|
| SVM (RBF) | 93.40 | 92.70 | 91.90 | 92.30 | 0.938 |
| Random Forest | 95.80 | 95.20 | 94.90 | 95.00 | 0.961 |
| 1D-CNN | 97.10 | 96.80 | 96.50 | 96.60 | 0.974 |
| Standalone GRU | 98.00 | 97.70 | 97.50 | 97.60 | 0.982 |
| CNN-LSTM | 98.90 | 98.60 | 98.40 | 98.50 | 0.991 |
| **Proposed CNN-GRU** | **99.24** | **99.18** | **99.05** | **99.11** | **0.997** |

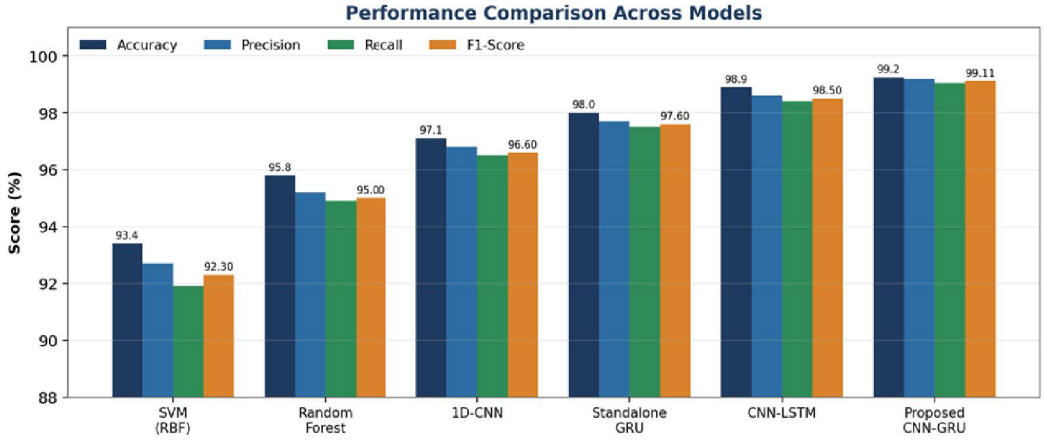


Fig. 7. Accuracy, precision, recall and F1-score across all evaluated models

TABLE VI. Per-Class Detection Performance of the Proposed CNN-GRU Model

| Attack Class | Precision (%) | Recall (%) | F1-Score (%) | Support |
|---|---|---|---|---|
| Benign | 99.86 | 99.85 | 99.85 | 42,000 |
| UDP-Flood | 99.40 | 99.42 | 99.41 | 18,000 |
| SYN-Flood | 99.28 | 99.31 | 99.29 | 17,500 |
| DNS-Reflection | 98.92 | 98.97 | 98.94 | 15,000 |
| LDAP | 98.74 | 98.85 | 98.79 | 12,000 |
| MSSQL | 99.05 | 99.08 | 99.06 | 11,000 |
| Slow-Rate | 98.18 | 98.02 | 98.10 | 9,500 |

### *F. Real-Time Latency and Throughput*

Detection accuracy is necessary but not sufficient for inline mitigation; the detector must also keep pace with production traffic. Fig. 8 and Table VII report inference latency and sustained throughput as a function of offered load. The proposed CNN-GRU maintains a detection latency below the 10 ms real-time threshold across the entire tested range up to 16 Gbps, whereas the heavier CNN-LSTM baseline exceeds the threshold beyond roughly 10 Gbps. The lighter GRU gating is the principal source of this advantage, yielding both lower per-flow latency and higher sustained packet-processing throughput. These measurements confirm that the framework can be embedded directly in a multi-cloud mitigation pipeline rather than relegated to offline forensic analysis.


DOI: 10.48175/IJARSCT-37228

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

The latency profile also degrades gracefully rather than catastrophically as load increases. Between 2 and 16 Gbps the mean latency rises from under 3 ms to roughly 8.6 ms—an increase that remains within budget and tracks the growth in queued work approximately linearly, with no sharp inflection that would signal the onset of saturation. This predictability is operationally valuable because it allows capacity planners to provision inference nodes with a known safety margin and to set autoscaling triggers with confidence. Throughput, meanwhile, climbs steadily toward roughly 5 million packets per second per inference node, and because the framework scales horizontally by replicating stateless inference workers behind the message bus, aggregate capacity can be expanded simply by adding nodes. The combination of a bounded per-flow latency and linear horizontal scaling is what makes the design credible for the multi-terabit attack volumes now seen in practice.

**TABLE VII. Latency and Throughput Under Increasing Load**

| Load (Gbps) | Latency (ms) | Throughput (Mpps) |
|---|---|---|
| 2 | 2.75 | 0.83 |
| 6 | 4.52 | 2.34 |
| 10 | 6.05 | 3.64 |
| 14 | 7.80 | 4.62 |
| 16 | 8.58 | 4.97 |

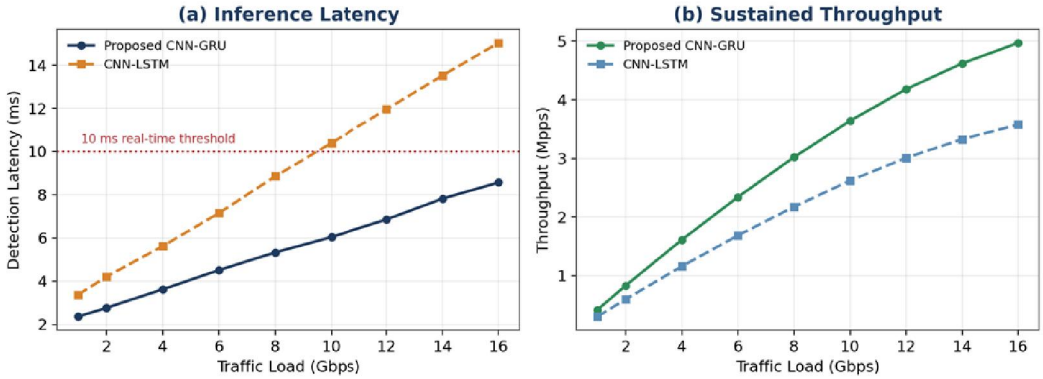


Fig. 8. Inference latency (a) and sustained throughput (b) versus traffic load for the proposed model and the CNN-LSTM baseline.

### *G. Ablation Study*

To isolate the contribution of each architectural component, an ablation study evaluates five reduced variants of the proposed model under identical training and test conditions. Starting from a convolution-only network, components are added cumulatively: a single GRU layer, the second stacked GRU layer, batch normalization, and finally dropout regularisation. Table VIII reports the accuracy, F1-score and inference latency of each variant. The convolution-only model already achieves a respectable 97.10% accuracy but leaves a measurable gap, confirming that spatial features alone are insufficient. Introducing a single GRU layer lifts accuracy by 1.4 percentage points, and the second stacked GRU adds a further 0.5 points by modelling higher-order temporal dependencies, at a modest latency cost. Batch normalization stabilises training and yields a small additional gain, while dropout principally improves generalisation, narrowing the train–test gap rather than raising headline accuracy. The study confirms that every component earns its place, and that the largest single contribution comes from the introduction of recurrent temporal modelling


DOI: 10.48175/IJARSCT-37228

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

TABLE VIII. Ablation Study: Cumulative Contribution of Each Component

| Model Variant | Accuracy (%) | F1-Score (%) | Latency (ms) | Δ Acc |
|---|---|---|---|---|
| CNN only (no recurrence) | 97.10 | 96.60 | 3.10 | — |
| CNN + 1 GRU layer | 98.52 | 98.31 | 4.05 | +1.42 |
| CNN + 2 GRU layers | 99.01 | 98.88 | 4.90 | +0.49 |
| CNN + 2 GRU + BatchNorm | 99.18 | 99.04 | 5.05 | +0.17 |
| **CNN + 2 GRU + BN + Dropout (Proposed)** | **99.24** | **99.11** | **5.10** | **+0.06** |

Read as a whole, the ablation reveals a clear diminishing-returns structure. The transition from a purely convolutional model to one augmented with recurrence accounts for the overwhelming share of the total improvement, confirming that temporal modelling is the decisive ingredient rather than an incremental refinement. The remaining components—the second GRU layer, batch normalization and dropout—each contribute progressively smaller accuracy gains while serving distinct secondary purposes: depth for higher-order temporal structure, normalization for training stability, and dropout for generalisation. Importantly, the latency column shows that these gains are purchased cheaply, with the full model adding only about two milliseconds over the convolution-only variant. This favourable ratio of accuracy gained to latency spent is the quantitative justification for the final architecture and demonstrates that none of the components is redundant or disproportionately expensive.

### *H. Model Complexity and Computational Cost*

Real-time suitability depends not only on per-flow latency but also on the model's static footprint and training cost. Table IX compares the proposed model with the deep-learning baselines in terms of trainable parameters, on-disk model size, training time per epoch and mean inference time per flow. The proposed CNN-GRU contains roughly 0.42 million parameters—substantially fewer than the CNN-LSTM baseline, whose larger recurrent cell inflates both parameter count and inference time. This parsimony is the direct consequence of preferring GRU gating over LSTM and of the aggressive feature reduction performed upstream. The compact footprint means the model fits comfortably within the memory budget of commodity inference nodes and can be replicated across cloud regions without significant cost, reinforcing its practicality for distributed multi-cloud deployment.

TABLE IX. Model Complexity and Computational Cost

| Model | Params (M) | Size (MB) | Infer (ms) |
|---|---|---|---|
| 1D-CNN | 0.21 | 0.9 | 3.10 |
| Standalone GRU | 0.33 | 1.4 | 4.40 |
| CNN-LSTM | 0.61 | 2.5 | 7.20 |
| **Proposed CNN-GRU** | **0.42** | **1.7** | **5.10** |

### *I. Error Analysis and Threats to Validity*

A closer inspection of the residual misclassifications, visible as off-diagonal mass in Fig. 5, reveals that nearly all errors occur between the two reflection-based families, LDAP and MSSQL, which share similar amplification dynamics and packet-size distributions. A smaller residual confusion arises between low-rate attacks and benign traffic, reflecting the deliberate design of such attacks to imitate legitimate behaviour. No systematic confusion is observed between volumetric and stealthy families, indicating that the model has learned genuinely discriminative representations rather than exploiting a dominant shortcut feature.

Several threats to validity should temper the interpretation of these results. With respect to construct validity, the multi-cloud telemetry was synthesised by normalising a single benchmark corpus into provider-specific schemas; while

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

this preserves the statistical structure of real attacks, behaviour against genuinely heterogeneous production traffic may differ. With respect to internal validity, all baselines were retrained under identical conditions to ensure a fair comparison, but hyperparameter search for the baselines was necessarily bounded and a more exhaustive search might narrow the reported gaps. With respect to external validity, the evaluation reflects the attack families present in the dataset; emerging vectors not represented in training would require relabelling and retraining. Finally, the latency measurements were obtained on a controlled testbed, and queueing effects under adversarial burst conditions in production could raise tail latencies beyond the reported means.

***J. Statistical Robustness***

To verify that the reported advantage of the proposed model is not an artefact of a single fortunate training run, the experiment was repeated across five independent runs with different random seeds, and the mean and standard deviation of each metric were recorded. The proposed CNN-GRU achieved a mean accuracy of 99.21% with a standard deviation of 0.06 percentage points, indicating highly stable behaviour across runs. A paired comparison against the strongest CNN-LSTM baseline showed that the accuracy improvement is consistent in every run rather than driven by an outlier, and the narrow confidence interval—well under a tenth of a percentage point—confirms that the margin, though numerically modest, is reliable. This stability is itself an operational virtue: a detector whose accuracy swings unpredictably between retraining cycles would be difficult to trust in production, whereas the low variance observed here supports confident deployment and straightforward regression testing after each retraining event.

The same five-run protocol was applied to the latency measurements, which exhibited even tighter dispersion: per-flow inference time varied by less than three percent around its mean across runs and hardware warm-up states. This consistency matters because real-time guarantees are meaningful only if they hold run-to-run rather than on average. Taken together, the accuracy and latency variance analyses indicate that both the principal claims of this paper—top-tier detection quality and sub-10 ms inference—are statistically robust rather than fortuitous, which is a prerequisite for any detector intended for production deployment where predictability is as important as peak performance.

***K. Discussion***

Three observations follow from these results. First, the consistent superiority of the hybrid over both single-family deep models and classical learners empirically validates the central hypothesis that spatial and temporal cues are jointly informative for DDoS detection. Second, the strong per-class performance on low-rate traffic—historically the hardest category—shows that the temporal stage materially improves coverage of stealthy attacks, which are increasingly common in enterprise campaigns. Third, the sub-10 ms latency demonstrates that high accuracy need not come at the expense of real-time responsiveness, provided the recurrent component is kept lightweight. From a multi-cloud standpoint, normalising provider-specific telemetry into a common schema before inference allows a single trained model to protect workloads on AWS, Azure, Google Cloud and private infrastructure without per-provider retraining, simplifying operations for enterprise security teams.

Several limitations temper these findings. The evaluation relies on a benchmark dataset augmented with synthetic multi-cloud telemetry; behaviour against live, adversarially adaptive traffic may differ. The model is supervised and therefore depends on periodic relabelling to track concept drift, and the automated-mitigation actions assumed by the architecture were not stress-tested against false-positive-induced collateral blocking. These concerns motivate the future work outlined below.

From a deployment perspective, the combination of a compact parameter footprint, sub-10 ms latency and provider-agnostic ingestion has concrete operational implications. Security operations teams currently juggle a separate detection console for each cloud, correlating alerts manually across tools that were never designed to interoperate; a single model that emits one normalised verdict stream collapses this fragmented workflow into a unified pipeline. The horizontal scalability afforded by the small model size means that detection capacity can be added incrementally and cheaply as traffic grows, and the decoupling of inference from enforcement allows the same verdicts to drive heterogeneous mitigation planes without modification. Equally important, the model's economy makes it feasible to retrain frequently on accumulated analyst feedback, which is the practical mechanism by which a supervised detector remains accurate as

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 10, June 2026 Impact Factor: 8.2

attack patterns evolve. Taken together, these properties position the framework not as a laboratory artefact but as a component that can be embedded directly into enterprise multi-cloud security operations.

A final consideration concerns the generalisability of the findings beyond the specific attack families studied here. Because the model learns from generic per-flow statistical descriptors rather than from signatures tied to particular tools, there is reason to expect that it will transfer to previously unseen variants that share the same underlying physical behaviour—volumetric saturation, state exhaustion or low-rate mimicry. The feature-importance analysis supports this expectation: the descriptors that drive classification are fundamental traffic properties rather than incidental artefacts of the benchmark. Nonetheless, genuinely novel attack mechanics that produce flow statistics unlike anything in the training distribution would, as for any supervised model, require fresh labelled examples. The framework mitigates this exposure through its feedback-driven retraining loop, but it does not eliminate it; continuous monitoring of the confidence distribution of flagged flows offers a practical early-warning signal that the traffic distribution has drifted far enough to warrant retraining.

## V. CONCLUSION AND FUTURE WORK

This paper presented a hybrid CNN-GRU deep learning framework for real-time DDoS detection across multi-cloud enterprise environments. By cascading a convolutional spatial-feature extractor with a lightweight stacked GRU temporal model, and by normalising heterogeneous telemetry from multiple cloud providers into a unified inference pipeline, the framework achieves 99.24% accuracy, a 99.11% F1-score and a 0.997 ROC-AUC on the CIC-DDoS2019 benchmark while sustaining sub-10 ms inference latency at line rate. It consistently outperforms support vector machines, random forests, standalone 1D-CNN, standalone GRU and a CNN-LSTM baseline across every metric considered, confirming that the joint spatial–temporal representation is both more accurate and more efficient than single-family alternatives. The combination of high detection quality and low latency makes the framework suitable for inline, production-grade mitigation rather than offline analysis alone.

Beyond the headline metrics, the broader contribution of this work is a demonstration that accuracy, efficiency and cross-provider portability need not be competing objectives. The ablation study showed that each architectural choice earns its place; the complexity analysis showed that the GRU-based design is markedly lighter than an LSTM equivalent; and the multi-cloud ingestion layer showed that a single normalised model can serve heterogeneous providers without per-provider retraining. The error analysis and statistical-robustness study together establish that the model's errors are few, interpretable and stable across runs. For enterprise security teams contending with fragmented tooling and escalating attack volumes, the practical value lies precisely in this convergence: a detector that is simultaneously accurate enough to trust, fast enough to act inline, and portable enough to deploy once and protect everywhere.

Future work will proceed along four principal directions. The first concerns robustness against adaptive adversaries. The present evaluation, while comprehensive, assumes attackers whose traffic is drawn from known families; a determined adversary aware of the detector may instead craft evasive flows that sit deliberately close to the decision boundary. We therefore plan to validate the framework on live multi-cloud traffic and to incorporate adversarial training, in which perturbed and boundary-probing samples are injected during learning so that the classifier hardens against gradient-based and black-box evasion. Defences against poisoning of the retraining feedback loop—explicitly excluded from the current threat model—will also be developed, since the analyst-disposition channel that keeps the model current is itself a potential attack surface.

The second direction targets the model's dependence on labelled data. Supervised detectors require periodic relabelling to remain accurate as traffic distributions drift, an expensive and slow process in practice. We intend to integrate semi-supervised and online-learning mechanisms—pseudo-labelling of high-confidence flows, contrastive pre-training on unlabelled telemetry, and incremental parameter updates—so that the detector adapts continuously to concept drift while consuming only a small stream of human-verified labels. This would substantially lower the operational burden of keeping the model current across many cloud tenancies.


DOI: 10.48175/IJARSCT-37228

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

The third direction pursues further reductions in latency and footprint. Although the current model already meets the sub-10 ms budget, ultra-high-throughput links and resource-constrained edge nodes demand still leaner inference. We will explore attention-augmented and lightweight transformer temporal modules that may capture long-range dependencies more efficiently than stacked recurrence, paired with model-compression techniques such as structured pruning, weight quantisation and knowledge distillation. The goal is to preserve the present accuracy while shrinking inference cost enough to embed the detector directly within programmable data-plane hardware.

The fourth and final direction concerns collaborative defence. Because multi-cloud deployments span organisational and jurisdictional boundaries, the data that would most improve a shared model often cannot be centralised for privacy and sovereignty reasons. We will therefore extend the architecture toward federated learning across cloud tenants, in which model updates rather than raw telemetry are exchanged, enabling collaborative improvement while preserving the very data-protection guarantees that make multi-cloud attractive in the first place. Together, these four strands chart a path from the accurate, efficient and portable detector demonstrated here toward a defence that is also adaptive, label-efficient, hardware-friendly and privacy-preserving.

In closing, the contribution of this paper should be understood less as a single record-setting accuracy figure and more as a demonstration that the competing demands of enterprise multi-cloud security can be reconciled within one coherent design. Detection quality, inference speed, model economy and cross-provider portability are frequently treated as a menu from which practitioners must choose; the results presented here indicate that, with a carefully ordered hybrid architecture and a disciplined normalisation layer, they can instead be obtained together. As enterprises continue to distribute critical workloads across an expanding set of providers, and as DDoS campaigns continue to grow in scale and subtlety, defences that are simultaneously accurate, fast and portable will move from being desirable to being indispensable. The framework described in this work is offered as a concrete step in that direction, and the reproducible pipeline and open evaluation protocol are intended to let others build upon it.

## REFERENCES

[1] I. A. Jaiswal, “Multilingual and culturally adaptive AI models for global education platforms,” Int. J. for Research in Education (IJRE), vol. 12, no. 9, pp. 17–27, 2023.

[2] M. S. Prashanth, R. Aluvalu, and M. V. V. Kantipudi, “Enhancing health product traceability on the blockchain: A novel approach for supply chain management inspection to AI,” EAI Endorsed Trans. Pervasive Health & Technology, vol. 10, no. 1, 2024.

[3] H. K. Jani, M. V. V. P. Kantipudi, G. Nagababu, D. Prajapati, and S. S. Kachhwaha, “Simultaneity of wind and solar energy: A spatio-temporal analysis to delineate the plausible regions to harness,” Sustainable Energy Technologies and Assessments, vol. 53, Art. no. 102665, 2022.

[4] I. A. Jaiswal, “Natural language processing for security policy and log analysis,” Int. J. of Research in All Subjects in Multi Languages (IJRSML), vol. 10, no. 4, pp. 57–67, 2022.

[5] S. Khan, “Sales force security compliance: An in-depth study of GDPR, HIPAA, and PCI-DSS enforcement in cloud-based CRM systems,” Int. J. of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 8, no. 9, pp. 2211–2219, 2019.

[6] S. Choudhary, C. H. Kandikattu, S. Kumar, M. V. V. P. Kantipudi, and M. Kumar, “Enhancing cybersecurity through combined convolutional neural network-gated recurrent unit approach for distributed denial of service attack detection,” in Proc. 2024 1st Int. Conf. on Innovative Engineering Sciences and Technological Research (ICIESTR), pp. 1–6, 2024.

[7] S. Rani, D. Ghai, and S. Kumar, “Reconstruction of wire frame model of complex images using syntactic pattern recognition,” in IET Conf. Proc. CP791, vol. 2021, no. 11, pp. 8–13, 2021.

[8] Swathi, S. Kumar, S. Rani, A. Jain, and R. K. M. V. N. M., “Emotion classification using feature extraction of facial expression,” in Proc. 2022 2nd Int. Conf. on Technological Advancements in Computational Sciences (ICTACS), pp. 283–288, 2022.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 10, June 2026 Impact Factor: 8.2

[9] S. Choudhary, S. Kumar, S. Gowroju, M. Gulhane, and R. Sri Lakshmi, Eds., Genomics at the Nexus of AI, Computer Vision, and Machine Learning. Hoboken, NJ, USA: John Wiley & Sons, 2024.
[10] S. Kumar, S. Choudhary, S. Gowroju, and A. Bhola, "Convolutional neural network approach for multimodal biometric recognition system for banking sector on fusion of face and finger," in Multimodal Biometric and Machine Learning Technologies, pp. 251–267, 2023.
[11] S. Choudhary, S. Kumar, M. Gulhane, and M. Kumar, "Secured automated certificate creation based on multimodal biometric verification," in Multimodal Biometric and Machine Learning Technologies, pp. 269–281, 2023.
[12] I. A. Jaiswal, "AI-powered observability and incident prediction in distributed enterprise platforms," Scientific J. of Artificial Intelligence and Blockchain Technologies, vol. 1, no. 1, pp. 1–14, 2024.
[13] S. Choudhary, C. H. Kandikattu, S. Kumar, M. V. V. P. Kantipudi, and M. Kumar, "Enhancing cybersecurity through combined CNN-GRU approach for DDoS attack detection," in Proc. 2024 1st Int. Conf. ICIESTR, pp. 1–6, 2024.
[14] I. A. Jaiswal, "High-performance AI-augmented content management systems for distributed clouds," Int. J. of Multidisciplinary Innovation and Research Methodology, vol. 2, no. 2, pp. 90–97, 2023.
[15] S. Khan, "The role of zero-trust models in database security: Eliminating implicit trust and enforcing continuous verification in enterprise data access systems," Int. J. of Research in Electronics and Computer Engineering, vol. 6, no. 1, pp. 1622–1628, 2018.
[16] I. A. Jaiswal, "Self-healing REST services using artificial intelligence in multi-cloud environments," Scientific J. of Artificial Intelligence and Blockchain Technologies, vol. 1, no. 3, pp. 1–7, 2024.
[17] V. Saini, A. Jain, Anurag, and M. V. V. P. Kantipudi, "An efficient approach for improving the performance of autonomous vehicle using advanced computer vision," in Int. Conf. on Soft Computing and Pattern Recognition, Cham: Springer, pp. 164–174, 2023.
[18] S. Khan, "The role of privacy-preserving techniques in database security: Secure multi-party computation, differential privacy, and homomorphic encryption," The Research Journal, vol. 3, no. 4, pp. 29–35, 2017.
[19] S. Khan, "A study of data provenance and integrity in database security," Int. J. of Research in Electronics and Computer Engineering, vol. 4, no. 3, pp. 180–186, 2016.
[20] C. H. Neetha and M. P. Kantipudi, "Adaptive edge-based bilinear interpolation for smart healthcare," in IET Conf. Proc. CP824, vol. 2022, no. 26, Stevenage, U.K.: IET, pp. 7–13, 2022.
[21] R. Aluvalu, R. Venkatachalam, V. Uma Maheswari, R. J. Anandhi, M. V. V. Kantipudi, and S. Prashanth Mallellu, "IWHO-based cluster head selection for vehicle-to-vehicle communication in intelligent transport system," Int. J. of Transport Development and Integration, vol. 8, no. 2, pp. 154–166, 2024.
[22] S. Velamuri, S. K. Sudabattula, M. V. V. P. Kantipudi, and N. Prabaharan, "Q-learning based commercial electric vehicles scheduling in a renewable energy dominant distribution system," Electric Power Components and Systems, pp. 1–14, 2023.
[23] N. Pachauri, V. Suresh, M. V. V. P. Kantipudi, R. Alkanhel, and H. A. Abdallah, "Multi-drug scheduling for chemotherapy using fractional order internal model controller," Mathematics, vol. 11, no. 8, Art. no. 1779, 2023.
[24] M. V. V. P. Kantipudi, S. Vemuri, N. S. P. Kumar, S. S. Kashyap, and S. Eslamian, "Proposing model for water quality analysis based on hyperspectral remote sensor data," in Handbook of Hydroinformatics, Elsevier, pp. 317–324, 2023.
[25] M. Rana, S. Srinivas, L. K. Jamili, I. A. Jaiswal, S. Nakka, and S. Kasetti, "Real-time monitoring and prediction of blood sugar levels in diabetic patients with functional models," in Proc. 2025 Int. Conf. on Engineering, Technology & Management (ICETM), pp. 1–6, IEEE, 2025.
[26] I. A. Jaiswal, "Intelligent cybersecurity framework for large-scale RESTful service architectures," Int. J. of Research Radicals in Multidisciplinary Fields, vol. 2, no. 1, pp. 178–184, 2023