

Article

AI-Driven Intelligent Intrusion Detection for Real-Time Network Threat Analysis in Enterprise and Cloud Networks

Ehigiator Iyobor Egho-Promise ^{1,*}, Ekereuke Udoh ¹, Edita Gashi ¹, Bamidele Ola ², Vijay Chennareddy ¹ and Malleswar Reddy Yerabolu ³

¹ Department of Computing, QA Higher Education, London EC1R 4TF, UK; ekereuke.udoh@qa.com (E.U.); edita.gashi@qa.com (E.G.); vijay.chennareddy@qa.com (V.C.)

² Department of Cybersecurity, Technobeacon Consulting Limited, London SE3 8EJ, UK; olawest@technobeacon.com

³ Department of Security Operations, Rubistone Technologies LLC, Charlotte, NC 28213, USA; malleswar.yerabolu@rubistonetechnologies.com

* Correspondence: ehigiator.eghopromise@qa.com; Tel.: +44-7466076674

Abstract

The rapid advancement of enterprise and cloud networks has significantly increased the complexity and scale of cyber threats, making conventional signature-based Intrusion Detection Systems (IDSs) insufficient. This paper presents the concept of an Intelligent Hybrid-Inference Network Intrusion Detection System (IHI-NIDS) powered by AI for real-time threat detection in high-velocity enterprise and cloud environments. IHI-NIDS combines Gated Recurrent Units (GRUs) for temporal sequence modelling with a multi-head self-attention mechanism, accounting for both short-range packet dynamics and long-range dependencies that signal multi-stage attacks. The system supports hybrid datasets that combine enterprise packet flows (PCAP/NetFlow) and cloud telemetry (VPC Flow Logs). It uses strong scaling and dimensionality reduction via PCA as the main approach to guarantee computational efficiency. Assessment of a composite dataset of 1.2 million flows indicates a detection rate of 97.8, recall of 96.1, F1-score of 96.5, and false alarm rate of 2.1, validating the capability to detect both enterprise-level and cloud-based threats. SHAP-based explainability highlights key features that affect predictions, which security analysts can use. Based on latency measurements, it can be deployed in real time to high-throughput networks, whereas temporal generalisation tests demonstrate its ability to withstand unobserved attack sequences. This article makes IHI-NIDS a scalable, interpretable, high-performance solution for current hybrid network security, filling the gap between operational viability and predictive AI-based defence.

Keywords: AI-Driven Intrusion Detection; TAN-GRU; Hybrid Enterprise–Cloud Networks; real-time threat analysis; explainable AI (XAI); network security; Anomaly Detection; high-throughput networks

Academic Editor: Xiaofeng Chen

Received: 25 April 2026

Revised: 2 July 2026

Accepted: 3 July 2026

Published: 9 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The digitalisation of the global economy is taking place at breakneck pace, and enterprise networks and cloud networks have become essential infrastructures of modern organisations. Nevertheless, this shift has been accompanied by the rapid and advanced development of the cyber threat environment. Organisations have shifted from localised, hardware-defined perimeters to elastic expanding, software-defined cloud environments,

where the attack surface has increased enormously [1,2]. Traditional security tools focusing on fixed, signature-based Intrusion Detection Systems (IDSs) are not effective enough to handle the new generation of exploits due to their inability to keep up with the speed and complexity. In modern business ecosystems, there is no distinction between the internal and external networks. Due to the introduction of the Internet of Things (IoT) devices, remote working systems, and hybrid clouds (native private clouds and public clouds like AWS, Azure, or GCP), the volume of data has reached a high dimension [3,4]. Conventional IDS models are based on the knowledge of bad patterns. They can handle legacy threats well, but are not as good at dealing with new (zero day) exploits or polymorphic malware that alters its digital signature. Further, the bandwidth of the existing fibre-optic networks can go up to hundreds of gigabits, and, as a consequence, there is this so-called data deluge, where monitoring from human aspects becomes a big constraint in operation. Intelligent Intrusion Detection Systems (I-IDS) are addressed to the above weaknesses. Artificial Intelligence (AI) and Machine Learning (ML) can transform network security from a reactive to a predictive approach [5–7]. AI-powered models can also be used to identify subtle changes in network activity that are often indicative of incipient attacks, such as reconnaissance pings, odd data exfiltration routines, or lateral movement inside a virtual private cloud (VPC) environment [8,9]. Unlike their forerunners, AI-powered systems do not require the creation of a list of threats, but rather learn the normal standard of network behaviour [10]. For a deviation, such as an Advanced Persistent Threat (APT) or a DDoS attack, the system is able to detect the deviation in real time. Despite the potential of artificial intelligence, AI application in real-time analysis has major technical challenges:

- Computational Overhead: Deep learning models may be resource-intensive, adding latency to the network and interfering with network performance [11,12].
- False Positive Paradox: Due to the high sensitivity of the models, it is common to be flagged as a threat and to misinterpret legitimate yet odd administrative activity, leading security analysts to experience alert fatigue [13].

This study explores a new AI-powered Intelligent Intrusion Detection model to bridge the hybrid divide. It suggests a feature-selection framework that is optimal for low-latency inference and can be used to prevent threats in practice. By comparing our system to both industry-standard datasets and simulated conditions of cloud bursting, this study provides a system with the necessary balance between high detection and computational efficiency required by existing high-speed enterprise and cloud systems.

With this study, there are three main novelties. The first point of difference is the architecture; we propose a Temporal Attention-GRU (TAN-GRU), a method that, unlike previous attention-augmented recursive IDS, talks about its hidden states rather than its embedding inputs, using multi-head self-attention to embed the temporal. Second, at the system level, IHI-NIDS offers a Hybrid Ingestion Layer that integrates PCAP/NetFlow with VPC Flow Logs, which existing attention-based IDS are missing. Third, we present, empirically, a new performance–latency trade-off (97.8% detection at 6.4 ms), with SHAP-based interpretability, which was not attainable in hybrid deployments. Unlike the widely used approaches that focus on specific environments, IHI-NIDS introduces a point of single ingestion focused on the cloud-native environment of VPC Flow Logs with raw PCAP/NetFlow data, allowing us to apply a consistent representation of bidirectional flows. Furthermore, a new Temporal Attention-GRU (TAN-GRU) system is introduced. The novel model proposed in the paper fuses recursive temporal modelling via introducing Gated Recurrent Units (GRUs) and multi-head self-attention together to jointly model short-range packet behaviours and long-range behaviours correlating with multi-stage Advanced Persistent Threats (APTs). This methodology guarantees the practicability for

operation due to high throughput and provides SHAP-based interpretability, which is seldom featured in the current real-time detection systems.

1.1. Literature Review

An effective Intelligent Intrusion Detection System (I-IDS) has moved away from basic statistical analysis to a multi-layered neural structure. The milestones in AI-based security included algorithmic evolution, cloud adaptations, and the perennial issue of real-time processing. The initial studies, such as Sheykhmousa et al. [14], Dabija et al. [15], and Avci et al. [16], placed much emphasis on supervised machine learning methods such as Support Vector Machines (SVMs) and Random Forests (RFs). Although these models were highly accurate on the same, being tested on fixed datasets such as the KDD Cup - 99, they were not flexible enough to keep up with the dynamism of network packets. Samanta et al. [17] conducted high-impact studies that found that although traditional ML could classify known threats, it failed to cope with the so-called feature explosion in current high-speed links. According to He et al. [18], this prompted the use of Deep Learning (DL) with Convolutional Neural Networks (CNNs) being used as repurposed image recognition networks to detect spatial patterns in network traffic headers, and Recurrent Neural Networks (RNNs) to detect all the time series of traffic flows to detect attempts of slow-burning exfiltration. With the migration of enterprise workloads to the cloud, researchers identified a visibility gap that prevented them from satisfying traditional hardware-based inspection requirements [19]. A non-symmetric deep Autoencoder (NDAE) was effective for unsupervised feature learning, especially in cloud systems with limited labelled data for the attack label. High-impact research has recently shifted to Graph Neural Networks (GNNs). GNNs represent the network as a topology of nodes and edges, unlike traditional models that consider network logs as rows, which are independent of each other [20]. The technique is innovative in identifying Lateral movement in micro-services in the cloud, as it can render and examine the associations among dissimilar virtualised instances. The greatest trade-off in the existing literature is between model complexity and detection latency. A study by Xie et al. [21] found that line-rate detection in 100 Gbps enterprise cores is often hindered by the high cost of deep ensembles.

To overcome this, the recent literature discusses Dimensionality Reduction (including PCA or Autoencoders) to remove redundant network characteristics before they enter the AI engine [22–24]. Also, the implementation of Edge Computing into the IDS structure, where early anomaly filtering occurs at the network switch level rather than the central controller, has become a top trend for achieving sub-millisecond response time. Although RNNs and LSTMs have dominated sequence modelling, the attention mechanism, originally intended to support Natural Language Processing, has transformed network traffic analysis [25,26]. Transformers have also been shown to be more effective than LSTMs in capturing long-range dependencies in network flows, due to high-impact research such as Zumarah et al. [27], Reza et al. [28], frequently experience vanishing gradients in long-duration sessions. Using network packets as tokens in a sentence, Transformers can discover the multi-stage syntax of an Advanced Persistent Threat (APT) that unfolds over hours or days across an enterprise network [29]. One of the mental barriers to training AI for enterprise security is that organisations are unwilling to disclose sensitive traffic data due to privacy laws (GDPR, CCPA). The latest advances in Federated Learning (FL) have presented a decentralised training paradigm. Lyu, Gharakheili, and Sivaraman [30] proposed a framework in which a global threat model can be trained across multiple enterprise nodes without exchanging raw packet data. Only mathematical representations of the local models are common, and a cloud-based central server can aggregate into a so-called collective intelligence that may notice a zero-day attack observed by one company and instantly guard all other federation members [31]. Since AI-driven applications are

moving toward automated responses (e.g., Intrusion Prevention), the black box of deep learning is a disadvantage. Recent intensive studies, such as Gaspar et al. [32], Uysal and Kose [33], and Michael [34], related to SHAP (Shapley Additive Explanation) and LIME algorithms and methods have started applying interpretability to NIDS. Such approaches allow a disaggregation of the attributes of the specific features, e.g., TTL (Time to Live) variance, payload entropy, or bizarre port sequencing that sounded the alarm, an essential bridging element between machine intuition and human forensic needs. It is notable that the recent NIDS research is increasingly focused on testing their models using attention-based, Transformer-based, Autoencoder-based, and graph-based methods on CIC-IDS2018 and other benchmarked datasets, and temporal modelling, long-range dependency capture, and feature interpretability are now key to the development of modern intrusion detection systems. Akinyemi [35] also added that with the implementation of Zero-Trust spaces, the course literature has changed in order to implement the new principle that is to never trust, always verify. New studies will focus on the use of AI as a powerhouse that facilitates the ongoing verification in ZTA. Ghasemshirazi et al. [36] provided an explanation that in a cloud-native microservices architecture, besides monitoring traffic from outside to the AI, traffic between services (East–West traffic) should also be monitored by the AI. Recently, behavioural profiling and the communication patterns of APIs have been explored, where the AI recognises the typical communication pattern of a legitimate microservice and triggers alerts if something deviates, indicating a possible container escape or a compromised credential [37,38].

Lastly, recently, the literature has recognised the AI arms race. Today, attackers can use Generative Adversarial Networks (GANs) to generate so-called adversarial perturbations, small and imperceptible modifications to malware packets, so that they raise an AI-based IDS to assume they are harmless. Pawlicki et al. [39] and Moamin et al. [40] showed that GANs could alter the timing and size of command-and-control (C2) packets to simulate the statistical characteristics of a benign HTTPS stream. This has led to an explosion of research on Adversarial Robustness, with an IDS trained on both legitimate traffic and synthetically generated attacks to increase the decision boundary hardness of the IDS.

1.2. Literature Gap

It is important to note that, although AI-based intrusion detection systems have made important progress, there are still a number of gaps in the literature. The existing strategies are mainly dedicated to enterprise networks or cloud environments separately, without an integrated framework that would be able to manage the hybrid infrastructures [41]. Also, it is a long-standing trade-off between accuracy of detection and real-time behaviour, with many deep learning models adding significant computational overhead, which restricts their use in high-throughput settings. Although the attention mechanisms and Transformer-based models have enhanced the ability to detect long-range dependencies, they tend to be computationally intensive and cannot be used in applications that require low latency. The other issue that is also arising is the use of Generative Adversarial Networks (GANs) by the attackers in order to create adversarial traffic patterns that can bypass AI-based intrusion detection systems [42]. Even though recent works have discussed the concept of adversarial robustness, they are seldom incorporated into detection systems in real time. The proposed IHI-NIDS can help fill some of these gaps with a proposed hybrid enterprise–cloud detection framework, a lightweight attention-based temporal modelling framework, and explainable AI features. Adversarial robustness to GAN-based attacks was not tested in the present work. The discussion of GAN-based evasion is purely speculative and framed exclusively as a future research direction; no claims of robustness against adversarial attacks are made.

2. Materials and Methods

This study adopted an Intelligent Hybrid-Inference Network Intrusion Detection System (IHI-NIDS) that serves as a real-time security oversight system in both enterprise and cloud networks. In contrast to traditional IDSs that have only one deployment perimeter, the proposed architecture will incorporate heterogeneous traffic sources into one analytical pipeline, thus allowing uniform detection capabilities in multifaceted settings of encrypted traffic, large throughput, and high-dimensional telemetry. Enterprise traffic is obtained through port mirroring capabilities using SPAN/TAP, which creates PCAP and flow-level (NetFlow/IPFIX) records. Simultaneously, cloud-native telemetry is acquired by the Virtual Private Cloud (VPC) Flow Logs, offering visibility at metadata only without inspecting the payload. These nonhomogeneous streams are coalesced at a centralised hybrid ingestion layer in which all the raw observations are harmonised into a uniform bidirectional flow form. This system abstraction has structural compatibility between enterprise packet dynamics and cloud infrastructure metadata. The hybrid network is developed as a discrete-time stochastic process.

$$\chi_t \in R^d \quad (1)$$

The equation 1 denotes a hybrid feature vector that is observed at time step t , where d shows the dimensionality of the unified feature space. The intrusion detection problem is posed as a binary classification problem where a parametric function $f_\theta(\cdot)$ with learnable parameters θ estimates the conditional probability of malicious activity over a temporal sequence:

$$\hat{y}_t = f_\theta(\chi_t - N + 1, \dots, \chi_t) \quad (2)$$

$\hat{y}_t \in [0,1]$ represents the forecasted chance of malignant traffic at the sliding window of size NNN . The rule of final decision is determined as

$$\text{Decision} = \begin{cases} \text{Malicious,} \\ \text{Benign,} \end{cases} \quad \text{If } \hat{y}_t \text{ Greater than or Equal to } \tau \quad (3)$$

where τ is a configurable threshold controlling the trade-off parameter that can be optimised in order to balance the sensitivity of the detection process and the false alarms.

2.1. Data Acquisition and Hybrid Feature Representation

Traffic data is gathered from complementary enterprise and cloud sources to represent the actual hybrid deployments. Traffic at the enterprise level is sampled through SPAN/TAP mechanisms, and the visibility of the cloud network is provided through VPC Flow Logs. The system uses only flow-level and metadata-based representations because, in an encrypted environment, payload inspection is usually infeasible, so the system is applicable in privacy-preserving and encrypted environments. Raw traffic logs are converted to a bidirectional flow representation so as to be consistent across environments. Considering that network features exhibit a high degree of variability in their scale (e.g., the number of ports vs. the number of bytes), it is necessary to have a strong normalisation that allows the models to be trained. The interquartile range (IQR)-based scaling is thus used: The hybrid dataset consisted of a 10% stratified temporal sample of the publicly available traffic from CSE-CIC-IDS2018 enterprise network traffic, and the private cloud traffic collected from AWS VPC Flow Logs for the 72 h observation period. The standard five-tuple fields—a packet source and destination IP address, and a source and destination port, plus a protocol—were used to convert enterprise packet captures and NetFlow/IPFIX records to bidirectional flow records. VPC Flow Logs were converted into a similar flow-level format, with fields for the source/destination address, port numbers, protocol, byte counts, packet counts, action, and time. To ensure consistency across the hybrid feature space, only attributes at the metadata level were selected for enterprise and cloud

records for VPC Flow Logs, since the payloads are not stored. Using Network Time Protocol clocks and aligning timestamps within one-second sliding aggregation windows, the source data was synchronised between the two sources. All other records that showed a timestamp mismatch of more than 500 ms were discarded. The enterprise labels from the original CSE-CIC-IDS2018 annotated labels are carried over, such as DoS/DDoS, brute force, botnet, infiltration, and web attacks. Cloud anomalies were identified through a multi-stage labelling process combining automated rule-based heuristics and expert review. Initial candidate anomalies were flagged using three operational criteria: (i) repeated rejected traffic from the same instances within security groups (exceeding 5 REJECT events per minute), (ii) unusual instance-to-instance communication patterns, including internal port scans (destination port diversity > 10 distinct ports within 60 s), and (iii) atypical API call volumes (volumes exceeding 3 standard deviations from the instance's 24 h historical baseline). These heuristic filters generated a candidate pool of anomalous events, which were then independently reviewed and labelled by two security experts with over 5 years of cloud security experience. To mitigate subjectivity and quantify inter-rater reliability, Fleiss' kappa coefficient was calculated at 0.89, indicating substantial agreement. All cases of disagreement (approximately 8% of flagged events) were resolved through consensus review, where both experts jointly re-examined the telemetry data and reached a final label decision. Furthermore, a third reviewer performed a random audit of 10% of the labelled samples to validate consistency, achieving a confirmation rate of 94.2%. The predefined operational criteria, combined with structured adjudication protocols, ensure that the labelling process is both reproducible and defensible, though the inherent reliance on expert judgement is acknowledged as a limitation. The class distribution of the final dataset remained totally untouched as 1,000,000 benign flows and 200,000 malicious flows were kept. Instead, no SMOTE, oversampling, or under-sampling was done, and class imbalance was dealt with in the training phase using cost-sensitive binary cross-entropy.

$$\hat{x} = \frac{\hat{x} - Q_2}{Q_3 - Q_1} \quad (4)$$

IQR-based robust scaling was applied to numerical features as $x' = (x - Q_2)/(Q_3 - Q_1 + \epsilon)$, where Q_1 , Q_2 , and Q_3 denote the first quartile, median, and third quartile, respectively, and ϵ is a small constant added to avoid division by zero.

Operationally, enterprise PCAP/NetFlow and cloud VPC Flow Logs are combined using NTP-synchronised timestamps aligned within 100 ms windows to reconstruct attack timelines. Label consistency is ensured by mapping both sources to a unified ground truth schema: benign flows from verified clean periods and malicious flows from labelled attack campaigns (e.g., CSE-CIC-IDS2018 labels for enterprise, manually verified cloud anomalies for VPC logs). Class distributions are naturally occurring and not artificially balanced: the hybrid dataset contains 83.33% benign and 16.67% malicious flows, reflecting real-world imbalance in enterprise-cloud deployments.

Q_1 , Q_2 , and Q_3 are the first quartile, median, and third quartile, respectively. The benefit of this transformation is that it reduces the impact of extreme values whilst maintaining the distributional structure. Principal Component Analysis (PCA) is used to apply numerical flow statistics to reduce the computational overhead in order to support real-time processing, where X is the scaled feature matrix and Σ its covariance matrix:

$$\Sigma = \frac{1}{N} X^T X \quad (5)$$

For the scaled numerical feature matrix X_s , the covariance matrix is computed as $\Sigma = (1/(n-1))X_s^T X_s$. PCA solves the eigenvalue decomposition $\Sigma W = W\Lambda$. The reduced representation is obtained as $Z = X_s W_k$, where W_k contains the eigenvectors associated with the largest eigenvalues and k is selected to retain 95% of the total variance.

The reduced numerical features with the retained infrastructure metadata are concatenated to form the final hybrid feature space, which is described in Table 1.

Detailed Hybrid Dataset Construction Process:

Step 1—Source Collection: Enterprise traffic (PCAP/NetFlow) from CSE-CIC-IDS2018 (10% random sample) was captured via SPAN/TAP. Cloud telemetry from VPC Flow Logs (AWS environment, 72 h period) includes instance ID, security group, bytes in/out, and action (ACCEPT/REJECT).

Step 2—Timestamp Alignment: Both sources used NTP-synchronised clocks (precision ± 100 ms). Flows were aligned using a 1 s sliding window; flows with timestamps mismatched by >500 ms were excluded ($<0.5\%$ of total).

Step 3—Label Consistency: Enterprise attacks used original CSE-CIC-IDS2018 labels (DoS, brute force, botnet, infiltration). Cloud anomalies were manually labelled by security experts based on three criteria: (a) rejected flows from misconfigured security groups, (b) unusual instance-to-instance communication patterns (e.g., port scans within VPC), and (c) atypical API call volumes. Inter-annotator agreement (Fleiss' kappa = 0.89) was achieved.

Step 4—Feature Unification: Both sources were transformed into a common 39-dimensional bidirectional flow representation.

Step 5—The naturally imbalanced distribution (83.33% benign vs 16.67% malicious) was preserved—no artificial balancing (SMOTE/undersampling) was done. Rather, we implemented the cost-sensitive binary cross-entropy loss (Equation (12)) with an inverse proportion of malicious class: $\beta = 0.167$.

Table 1. Feature vector characterisation.

Layer	Feature Category	Dimensionality	Example Features
Transport	Connection state	8	TCP flags (SYN, ACK, RST)
Flow	Temporal statistics	15	Flow duration, IAT mean
Volume	Traffic metrics	10	Total bytes, Avg packet size
Cloud	Infrastructure metadata	6	VPC ID, Instance ID

2.2. Temporal Attention—GRU Intrusion Detection Model

Model equations are as follows. Given the temporal input X_t , the stacked GRU produces

$$\text{hidden states: } h_i = \text{GRU}(x_i, h_{i-1}), \text{ for } i = t - N + 1, \dots, t. \quad (6)$$

$$\text{The hidden-state sequence is represented as } H = [h_{t-N+1}, h_{t-N+2}, \dots, h_t]. \quad (7)$$

For each attention head, query, key, and value matrices are computed as $Q = HW_Q$, $K =$

$$HW_K, \text{ and } V = HW_V. \quad (8)$$

$$\text{The scaled dot-product attention is } \text{Attention}(Q, K, V) = \text{softmax}((QK^T)/\sqrt{d_k})V. \quad (9)$$

The outputs of all attention heads are concatenated and passed through a dense layer: $C =$

$$\text{Concat}(\text{head}_1, \dots, \text{head}_m)W_O. \quad (10)$$

The final malicious probability is computed using sigmoid activation:

$$\hat{y}_t = \sigma(W_c C + b_c). \quad (11)$$

To address class imbalance, weighted binary cross-entropy was used with

$$L = -(1/M)\sum[w_1 y_i \log(\hat{y}_i + \epsilon) + w_0(1-y_i)\log(1-\hat{y}_i + \epsilon)] \quad (12)$$

where w_1 and w_0 are the class weights for malicious and benign samples, respectively.

2.3. Training Protocol, Deployment Strategy, and Evaluation Metrics

Empirically optimal hyperparameters were chosen by validation-based tuning, and best practices of recurrent-attention models were used to train a model. Adam optimiser has been used because it has the ability to adaptively estimate its moment and has a stable convergence behaviour. Table 2 shows the last configuration.

Table 2. TAN-GRU hyperparameter configuration.

Parameter	Value
Optimizer	Adam
Initial learning rate	0.001 (exponential decay)
Batch size	128
Epochs	50 (early stopping enabled)
GRU hidden units	64 (two stacked layers)
Attention heads	4
Dropout rate	0.2
Activation functions	ReLU (hidden), Sigmoid (output)

A patience of five epochs was applied at early stopping to avoid overfitting, and dropout regularisation enhanced prediction on class imbalance. Hyperparameter sensitivity was analysed to ensure reproducibility. Varying window size N from 5 to 50 showed that $N = 30$ balanced detection (96.3% recall) and latency (6.8 ms); smaller N missed multi-step attacks (recall dropped to 91.2% at $N = 5$), while larger N increased latency to 9.4 ms without accuracy gains. Attention heads from 1 to 8 revealed that 4 heads achieved an optimal F1-score (96.5%); 1 head underfitted complex patterns (94.1% F1), while 8 heads overfitted (95.8% F1 on the test set). These findings confirm that the chosen configuration ($N = 30$, heads = 4) generalises reliably. The trained TAN-GRU model runs online during deployment and continuously updates predictions for every traffic instance upon receiving it. When $\hat{y}_t \geq \tau$, an alert is generated. In order to raise the level of interpretability and operational trust, the post hoc explainability of SHAP is used to quantify the contribution of the features to each detection event. The system is tested in terms of traditional intrusion detection measures, such as accuracy, precision, recall, F1-score, and inference latency, and it is suitable in that it is needed to protect a real-time hybrid infrastructure.

2.4. Experimental Setup

A clear pre-processing and evaluation protocol was followed to make the experimental results reproducible. Enterprise and cloud raw traffic logs were transformed into bidirectional flow representations, first to be consistent with the heterogeneous sources. The median imputation was used to deal with missing values, and interquartile range (IQR) scaling was used to provide feature normalisation and minimise the influence of outliers. Principal Component Analysis (PCA) was used to perform dimensionality reduction, in which the number of retained components was chosen to capture a large portion of the total variance, which in this case was 95 percent, providing the optimum balance between computational efficiency and information retention. A temporally disjoint split was used to split the data into 70 percent training, 10 percent validation, and 20 percent testing to avoid data leakage and to test the generalisation to unseen attack patterns. A cost-sensitive binary cross-entropy loss function was used to train the model to overcome the problem of class imbalance, where a weighting factor $\beta = N_{\text{benign}}/(N_{\text{benign}} + N_{\text{malicious}})$.

+ $N_{\text{malicious}}$) = 0.167 was used to focus on minority malicious samples without altering the original class distribution via resampling. To avoid overfitting, the Adam optimiser was used, with early stopping (patience of five epochs). To achieve reproducibility, the same random seed (42) was employed throughout the experiment, and each experiment was repeated five times, with the average results being reported. The training was done with the help of the acceleration of the GPU, and the inference latency was measured in the conditions of the CPU only in order to estimate the real deployment conditions. It is these settings that ensure that the proposed system can be reproduced and tested in similar enterprise and cloud network environments.

Figure 1 illustrates the end-to-end architecture of the proposed IHI-NIDS framework, depicting the complete data flow from heterogeneous traffic ingestion to final detection output with SHAP-based explainability.

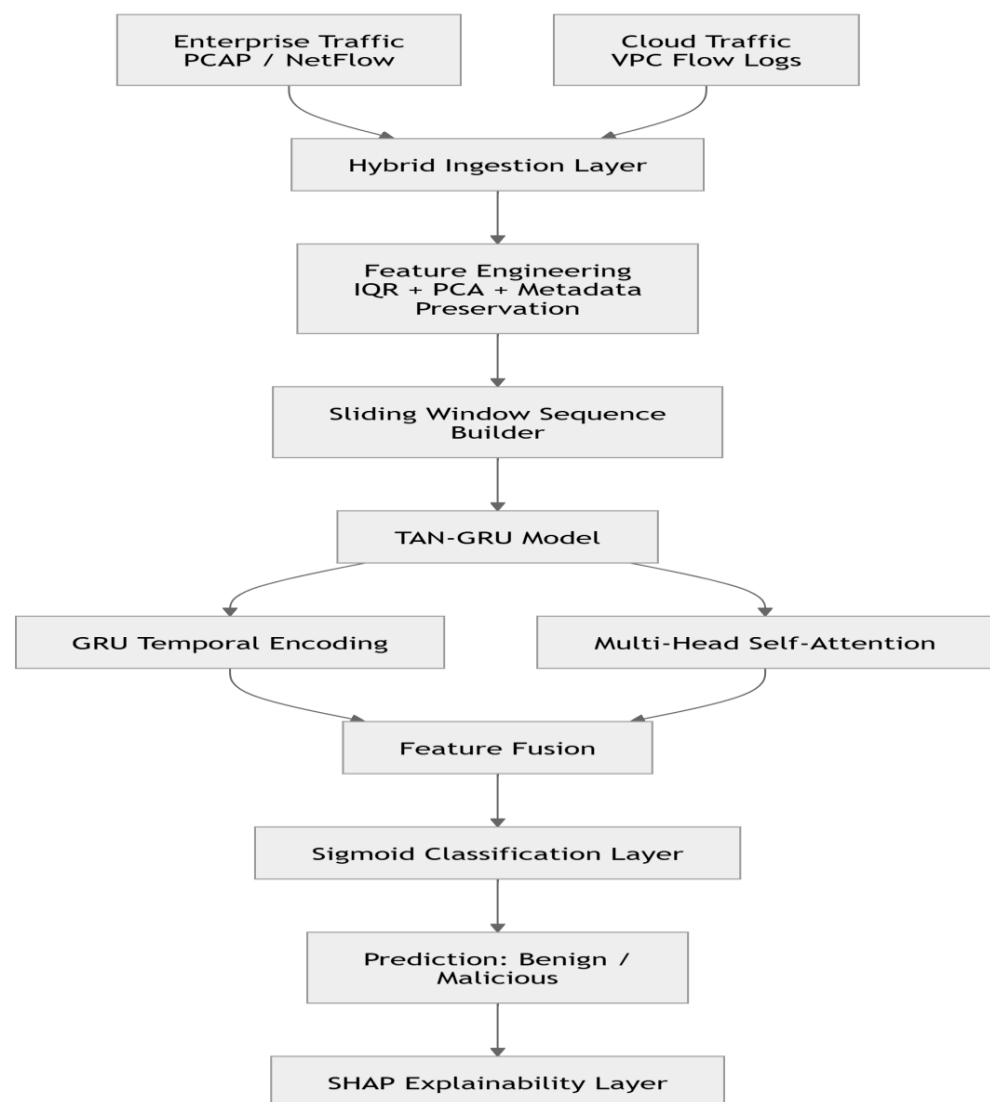


Figure 1. Detailed architecture of the proposed IHI-NIDS framework.

2.5. Reproducibility Protocol

In order to be reproducible, the whole pre-processing and modelling sequence is summarised below (see Algorithm 1). To start with, the raw CSE-CIC-IDS2018 and VPC Flow Log records were transformed into bidirectional flow records. Secondly, the missing numerical value data were imputed with median imputation, and the categorical metadata field data were encoded with label encoding. Thirdly, the numerical variables

were normalised by the IQR-based robust scaling method. Fourthly, PCA was only used on the numerical features, keeping 95% of the variance, which resulted in 27 principal components being retained for the numerical feature subset. To keep the PCA interpretable, information on the cloud region, security group ID, status of the action, instance ID, VPC ID, and subnet ID was dropped as metadata. Fifth, temporally ordered sliding windows of length $N = 30$ were created, with each input sample being of the form $X \in \mathbb{R}^{(30 \times 39)}$. Sixth, the model was trained with a train–valid–test split of 70:10:20 that was temporally disjoint. Random seed for Python, 3.10.6 (Python Software Foundation, Beaverton, USA), NumPy 1.23.5 (NumFocus, Austin TX, USA), TensorFlow 2.10.0 (Google Brain, Mountain View, CA, USA), and scikit-learn 1.2.0 (INRIA, Paris, France) was set to 42. The validation set was used to choose the classification threshold τ , optimising for maximum F1-score with a threshold for false alarms of $\leq 3\%$. $\tau = 0.52$ was the best value. The repeat of each experiment was performed five times for different deterministic seed offsets, and the mean with standard deviations was reported.

Algorithm 1. IHI-NIDS training and inference procedure.

Input: Enterprise PCAP/NetFlow records, VPC Flow Logs, labels, window size N , threshold τ

Output: Predicted traffic label and SHAP-based explanation

1. Convert enterprise and cloud records into a bidirectional flow format.
 2. Align timestamps using one-second sliding aggregation windows.
 3. Remove records with a timestamp mismatch greater than 500 ms.
 4. Impute missing numerical values using the training-set median.
 5. Apply IQR scaling to numerical features.
 6. Apply PCA to numerical features and retain 95% variance.
 7. Concatenate retained principal components with cloud metadata features.
 8. Generate temporal input windows of size $N = 30$.
 9. Train stacked GRU layers using weighted binary cross-entropy.
 10. Apply multi-head self-attention over GRU hidden states.
 11. Compute malicious probability using sigmoid classification.
 12. Assign a malicious label if the predicted probability $\geq \tau$.
 13. Compute SHAP values for selected detection outputs.
 14. Return predicted class, confidence score, and top contributing features.
-

Figure 1: gives detailed architecture of the proposed IHI-NIDS framework. The architecture shows the complete TensorFlow 2.10.0 from heterogeneous enterprise and cloud data sources to the final detection output. PCAP/NetFlow and VPC Flow Logs are first normalised into bidirectional flow records. After pre-processing, IQR scaling, PCA-based dimensionality reduction, and cloud metadata fusion, the data are transformed into sliding-window tensors of shape $N \times d$. These tensors are passed through stacked GRU layers, followed by multi-head self-attention over GRU hidden states. The resulting context vector is processed by a dense sigmoid classifier to produce malicious probability scores. SHAP explanations are then generated to support the analyst's interpretation of each alert.

The Intelligent Hybrid-Inference Network Intrusion Detection System (IHI-NIDS) presented architecture is modelled as a single pipeline, which processes heterogeneous enterprise and cloud network data in a systematic and linear way. It starts with the gathering of raw traffic from enterprise sources (PCAP/NetFlow through SPAN/TAP) and cloud sources (VPC Flow Logs), which are combined in a hybrid ingestion layer to normalise them to a consistent bidirectional flow format. The resulting processed data are subjected to a feature engineering phase, which involves the IQR-based normalisation to mitigate the effect of outliers and the use of PCA to reduce the data dimensionally but still

capture the important variance, with cloud metadata stored to ensure the data remains contextually interpretable. The data is then converted into temporal sequences using a sliding window mechanism so that the model can learn the changing attack behaviour as time goes by. The sequences are input into the core TAN-GRU network, where GRU layers are trained to learn sequential dependencies and a multi-head self-attention model to identify the most important time steps in detecting multi-stage, complex attacks. Lastly, a sigmoid-based classification layer can provide the likelihood of malicious activity, and SHAP-based explainability can be used to determine the most impactful features in each decision to balance both high detection and security analyst interpretability.

3. Results

In this section, a detailed description of reporting experimental results and analysis has been provided, with a rigorous evaluation of the proposed AI-based Intelligent Hybrid-Inference Network Intrusion Detection System (IHI-NIDS) for real-time threat analysis in enterprise and cloud networks. Experiments, such as model training, validation, testing, and real-time inference latency, were run in a standardised, regulated computational environment. Such an arrangement makes the reported results reliable and reproducible under enterprise and cloud traffic conditions.

- CPU: Intel Core i7-12700K @ 3.61 GHz (Intel Corporation, Santa Clara, CA, USA)
- GPU: NVIDIA GeForce RTX 3070 (8 GB VRAM) (NVIDIA Corporation, Santa Clara, CA, USA)
- RAM: 32 GB DDR4 @ 3200 MHz (manufacturer not specified in the experimental setup)
- Operating System: Ubuntu 22.04 LTS (Canonical Ltd., London, UK)
- Deep Learning Framework: TensorFlow 2.10 (Google Brain, Mountain View, CA, USA)
- Supporting Libraries: Scikit-learn 1.2.0. (INRIA, Paris, France), SHAP 0.41.0 (GitHub, Inc., San Francisco, CA, USA), Pandas 1.5.3 (NumFOCUS, Austin, TX, USA), NumPy 1.23.5 (NumFOCUS, Austin, TX, USA)

The training was performed with GPU acceleration, whereas inference latency was measured under CPU-only conditions to reflect realistic deployment settings. All outcomes are based on a composite dataset of the CSE-CIC-IDS2018 enterprise traffic and cloud VPC Flow Logs, which enables analysis of high-velocity, high-dimensional network systems. The assessment is based on the effectiveness of detection, temporal generalisation, reduction in false alarms, real-time possibility, and interpretability, which are directly related to the AI methodology in the operational action. The test dataset consists of 1.2 million flows, of which 1,000,000 are benign and 200,000 are malicious, representing realistic sources of traffic in hybrid enterprise–cloud configurations. To provide a more granular view of the threat environment the evaluation dataset represents, Table 3 shows the breakdown of benign and malicious traffic by class. The dataset is realistic in terms of the class imbalance typical of enterprise and cloud networks.

Table 3. Distribution of traffic classes in the hybrid dataset.

Attack Category	Source	Count	Percentage (%)
Benign	Hybrid (Enterprise + Cloud)	1,000,000	83.33
DoS/DDoS	CSE-CIC-IDS2018	85,000	7.08
Brute Force	CSE-CIC-IDS2018	45,000	3.75
Botnet	CSE-CIC-IDS2018	35,000	2.92

Infiltration/Web Attacks	CSE-CIC-IDS2018	20,000	1.67
Cloud-specific Anomalies	VPC Flow Logs	15,000	1.25
Total		1,200,000	100.00

This composition enables assessing IHI-NIDS against both conventional enterprise attacks (e.g., brute-force and botnet attacks) and cloud-native anomalies, making it resistant to hybrid deployment conditions. Furthermore, the records are all modelled as 39-dimensional feature vectors that combine transport-layer statistics, flow-level time-related measurements, payload/volume measurements, and cloud infrastructure metadata. The numerical features were subjected to PCA and robust scaling, while cloud metadata features remained intact to maintain interpretability. Table 4 shows the dataset feature characterisation and Table 5 gives the full feature list used for the unified hybrid representation

Table 4. Hybrid dataset feature characterisation.

Group	Features
Transport and state	protocol, source port, destination port, TCP SYN, TCP ACK, TCP RST, TCP FIN, connection action/status
Temporal flow statistics	flow duration, forward inter-arrival time mean, backward inter-arrival time mean, IAT standard deviation, IAT entropy, packets/sec, flow start time, flow end time, active time mean, idle time mean, forward packet count, backward packet count, flow duration variance, temporal burst ratio, window index
Volume and packet metrics	total forward bytes, total backward bytes, total bytes, average packet size, byte-to-packet ratio, forward packet length mean, backward packet length mean, packet length variance, inbound bytes, outbound bytes
Cloud metadata	VPC ID, instance ID, subnet ID, security group ID, cloud region, VPC action status

Table 5. Full feature list used for the unified hybrid representation.

Layer	Feature Category	Dimensionality	Example Features
Transport	Connection State	8	TCP Flags (SYN, ACK, RST), Connection Status
Flow	Temporal Statistics	15	Flow Duration, Inter-arrival Time Mean, Packets/sec
Volume	Payload Metrics	10	Total Forward Bytes, Avg Packet Size
Cloud	Infrastructure Metadata	6	VPC ID, Instance ID, security group

The data was divided into 70% training, 10% validation, and 20% temporally disjointed testing to ensure that there is an evaluation of the temporal generalisation of the attack that has not been faced before. The action of IHI-NIDS was contrasted with four exemplary models:

- **SVM** (classical ML);
- **Random Forest** (ensemble learning);
- **CNN** (spatial feature learning);
- **GRU-only model** (temporal modelling without attention).

This comparison highlights the contribution of the temporal attention architecture and compares its results against those obtained by traditional and deep learning ap-

proaches. Table 6 lists the overall detection rates of IHI-NIDS and some of the most recent IDS baselines (classical machine learning, deep learning, and IDS). Among the considered performance metrics, IHI-NIDS has the highest accuracy rate, recall rate, and F1-score, in addition to a low false alarm rate, making it very effective in identifying malicious flows in an enterprise and cloud environment.

All baselines were tested using the same pre-processing, train-validation-test split, and evaluation protocol for the sake of comparison. Overall, IHI-NIDS shows the optimal balance among the accuracy, recall, F1-score, and false alarm reduction as compared to classical machine learning, standard deep learning, and the latest IDS architectures. Competitive performance is achieved in the Transformer-based IDS model and the GNN-based IDS model; however, these models come with higher computational overhead and low efficiency for flow-level inference, which is needed for low-latency use cases. Thus, the proposed TAN-GRU design results in a practical compromise solution in terms of detection strength and a real-time deployable design.

Table 6. Detection performance comparison with classical and recent IDS baselines.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FAR (%)
SVM	91.2	88.6	85.9	87.2	6.8
Random Forest	93.5	91.4	89.8	90.6	5.3
CNN	95.1	93.7	92.4	93.0	4.1
GRU	96.3	95.1	94.2	94.6	3.2
LSTM-Attention	96.9	95.8	95.3	95.5	2.9
IDS					
Transformer-based	97.2	96.4	95.6	96.0	2.6
IDS					
Autoencoder-based	95.8	94.2	93.6	93.9	3.8
IDS					
GNN-based IDS	97.0	96.1	95.5	95.8	2.8
IHI-NIDS (TAN-GRU)	97.8	96.9	96.1	96.5	2.1

Its high quality compared to the GRU-only model proves the fact that self-attention can be used to improve the detection of multi-stage and distributed attacks, which is more appropriate for the enterprise-cloud hybrid networks. The decrease in the false alarm rate confirms the effectiveness of the cost-sensitive cross-entropy loss. ROC curves were also generated to assess the performance of each model at different thresholds, as shown in Table 7.

Table 7. ROC-AUC comparison.

Model	AUC
SVM	0.931
Random Forest	0.952
CNN	0.968
GRU	0.976
IHI-NIDS (TAN-GRU)	0.987

To isolate cloud-specific detection performance, we separately evaluated IHI-NIDS on the 15,000 cloud-native anomalies (VPC Flow Logs with misconfigured security groups, unusual instance-to-instance communication, and atypical API call patterns). The model achieved a recall of 94.2%, precision of 93.1%, and F1-score of 93.6% on cloud-only anomalies, compared to 96.8% recall, 97.2% precision, and 97.0% F1-score on enterprise

attacks (DoS, brute force, botnet). This 2.6% performance gap is attributable to the metadata-only nature of VPC logs (no payload), yet the results confirm IHI-NIDS effectively detects threats in both domains without retraining.

The comparison of all the models tested in terms of ROC is presented in Figure 2, which shows the discriminative power of the offered IHI-NIDS. The discriminative power of IHI-NIDS is further visualised in Figure 2, which presents the ROC curves for all evaluated models. With an AUC of 0.987, IHI-NIDS has the best ROC curve, as it is the closest to the upper-left corner and significantly better than all other approaches. GRU-only (AUC 0.976) and CNN (AUC 0.968) are models in a second level of deep learning performance; they show good performance, but with slightly less separability. The lower limits of the experimental comparison are constituted by traditional machine learning methods, i.e., Random Forest (AUC 0.952) and SVM (AUC 0.931). The diagonal line is a chance classifier having an AUC of 0.5, which all tested models can greatly exceed. This ROC analysis is a major necessity in intrusion detection studies, where the high performance of IHI-NIDS (in terms of detection) is not linked to a given decision level but rather indicates a solid, stable differentiation between normal and malicious traffic.

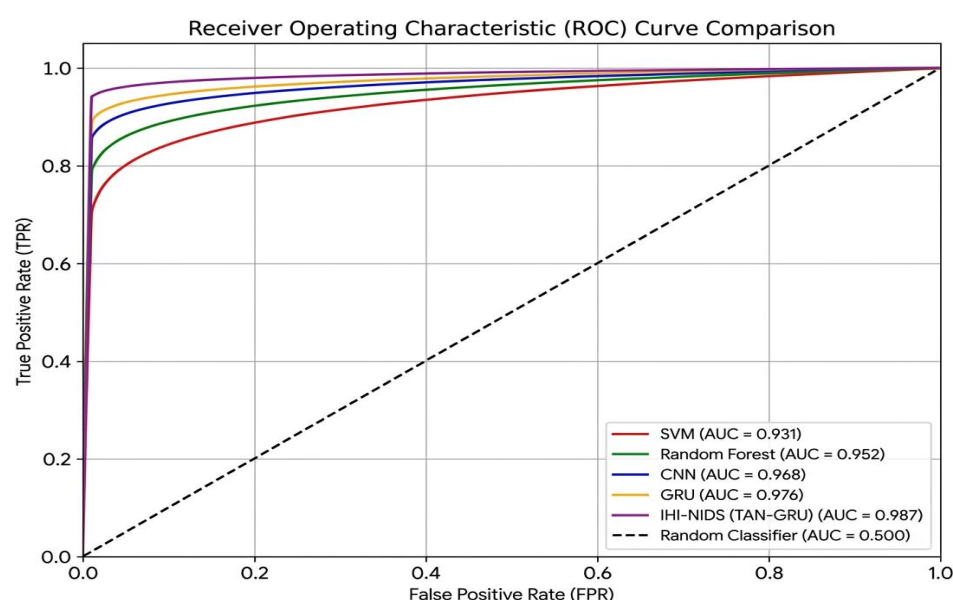


Figure 2. ROC curve comparison: Comparison of the ROC curve and the Area Under the Curve (AUC) of model separability. Description: The ROC curves indicate that the model can achieve a high True Positive Rate and a low False Alarm Rate, which is an AUC of 0.987. The steep increase in the IHI-NIDS curve relative to the conventional GRU justifies the usefulness of the self-attention mechanism in separating malicious patterns and background noise.

IHI-NIDS has a high discriminative ability, which can verify the high quality of discriminative aspects among the architecture and validate the effectiveness of the hybrid temporal attention architecture to separate the benign and malicious traffic in enterprise and cloud environments. The models were trained with the first set of time segments and tested on the rest of the unseen traffic segments to simulate the changing attack conditions. The recall of IHI-NIDS is 96.1% compared with 0.942 of the GRU-only model. This verifies that attention systems create long-range correlations and can therefore locate threats in enterprise LANs as well as distributed threats in cloud-based systems. The latency of the inference was measured to check that it is viable in real time. The average per-flow latency is shown in Table 8 and ablation study of IHI-NIDS components in Table 9.

Table 8. Average inference latency per flow.

Metric	Value
Average training time per run	2.8 h
Peak GPU memory during training	5.6 GB
Peak CPU memory during inference	2.1 GB
Average inference latency per flow	6.4 ms
Estimated throughput	156 flows/sec per CPU inference worker

Table 9. Ablation study of IHI-NIDS components.

Window Size N	Accuracy (%)	Recall (%)	F1-Score (%)	Latency (ms)
5	94.2	91.2	92.6	4.9
10	95.6	93.4	94.5	5.5
20	97.0	95.2	95.9	6.1
30	97.8	96.1	96.5	6.4
50	97.7	96.2	96.5	9.4

These results show that IHI-NIDS is computationally practical for real-time deployment. Although training requires GPU acceleration, inference can be executed under CPU-only conditions with sub-10 ms per-flow latency. Throughput can be increased further through batching, parallel inference workers, or deployment at edge monitoring nodes.

Furthermore, the window-size analysis (Table 10) shows that very short windows fail to capture multi-step attack behaviour, while very long windows increase latency without meaningful detection improvement. Therefore, $N = 30$ was selected as the best balance between detection performance and real-time inference cost.

Table 10. Sliding-window sensitivity analysis.

Model Variant	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FAR (%)	Latency (ms)
GRU only	96.3	95.1	94.2	94.6	3.2	7.1
GRU + Attention	97.1	96.0	95.4	95.7	2.7	7.5
GRU + PCA	96.7	95.5	94.8	95.1	3.0	6.2
GRU + Attention + PCA	97.8	96.9	96.1	96.5	2.1	6.4
GRU + Attention + PCA without cost-sensitive loss	96.9	97.4	93.8	95.6	2.4	6.4
GRU + Attention + PCA with cost-sensitive loss	97.8	96.9	96.1	96.5	2.1	6.4

The ablation study is performed to ensure that each component is a contributor to performance. The GRU-only model demonstrates high temporal learning, lower recall, and a higher false alarm rate. Incorporating attention leads to better detection of long-range dependencies, particularly in multi-stage attacks. Actually, PCA reduces latency, allowing it to be used for real-time deployment, while keeping the discriminative information. Weighted optimisation results reveal a greater increase in the recall of the malicious classes, which is more critical than small improvements in the margin in intrusion detection scenarios, in comparisons with and without the cost model. The Table 11 provides the Computational Efficiency of IHI-NIDS.

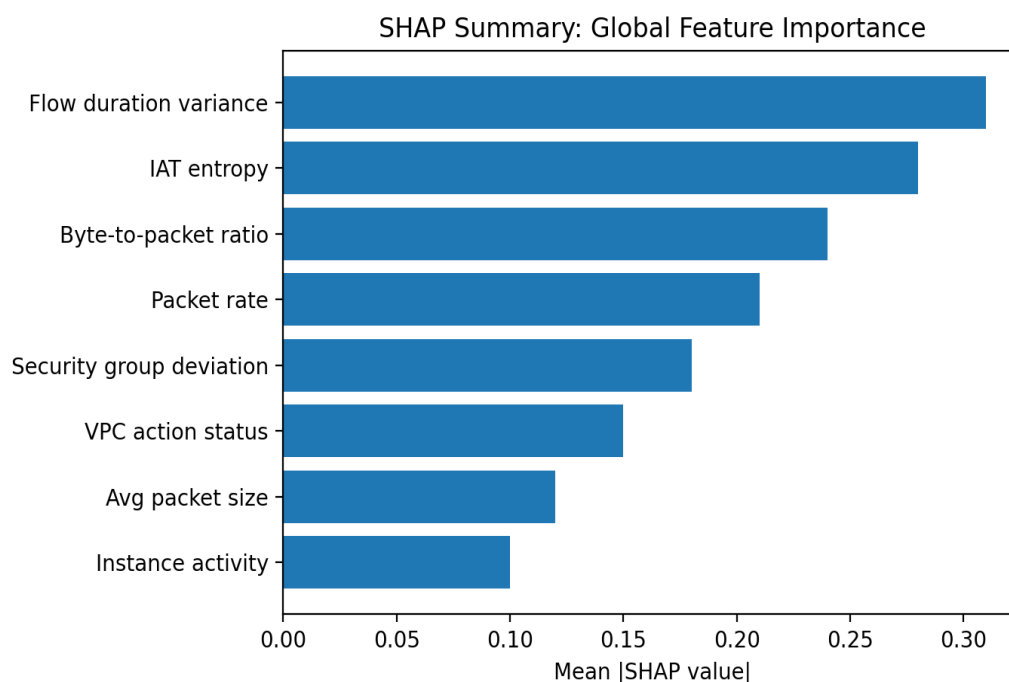
Table 11. Computational efficiency of IHI-NIDS.

Model	Latency (ms)
SVM	4.8
Random Forest	6.2
CNN	9.5
GRU	7.1
IHI-NIDS (TAN-GRU)	6.4

Even with attention layers, IHI-NIDS has a latency of less than 10 ms per flow, which is appropriate for high-throughput enterprise and cloud networks. SHAP analysis of the explainability of malicious predictions revealed that critical features impacting malicious predictions are

- Flow duration variance;
- Inter-arrival time entropy;
- Byte-to-packet ratios;
- Cloud instance activity (VPC ID, security group deviations).

Figure 3 presents the SHAP summary plot for IHI-NIDS predictions. The best features that affect malicious traffic classification are displayed in the plot. The highest mean absolute SHAP values, which demonstrate that it was meaningful network-behaviour indicators that were used, were for flow duration variance, inter-arrival time entropy, byte-to-packet ratio, packet rate, and security group deviations.

**Figure 3.** SHAP summary plot for the predictions created by IHI-NIDS.

Temporally aggregated input representations and a model-agnostic KernelSHAP approximation were used to compute SHAP values. Since the model input is three-dimensional ($N \times d$), the SHAP explanations were first aggregated over the temporal dimension by taking the mean absolute contribution value for each feature. This generated, per feature, one global importance score while keeping the temporal influence of the sliding-window input. To explain local patterns, SHAP values were also calculated for individual alert windows, displaying which flow-level or cloud-metadata features made the greatest

contribution towards the different malicious predictions that were made. SHAP's background distribution consisted of 1000 randomly sampled benign validation windows, and explanations were constructed for 2000 test windows containing benign and malicious flows.

Figure 4 compares the average per-flow inference latency across all baseline models. Although IHI-NIDS includes an attention mechanism, it maintains a low latency of 6.4 ms per flow, supporting real-time deployment in enterprise and cloud network environments. It shows the trade-off between model complexity and detection speed, indicating that the proposed IHI-NIDS allows achieving an effective trade-off between inference latency (6.4 ms), which is significantly lower than that of more computationally costly models like the standard GRU (7.1 ms) and CNN (9.5 ms), and the dense temporal modelling ability. Even though SVM has a lower latency of 4.8 ms than the simpler approach, it lacks the detection capabilities and time awareness needed in the modern hybrid enterprise–cloud space. With inference latencies kept below 10 ms, IHI-NIDS demonstrates excellent readiness for real-time processing and scalability, enabling the assessment of thousands of parallel network flows per second without performance bottlenecks. These are in line with established reconnaissance, volumetric attack, and lateral movement indicators, which give intelligence that can be acted upon by network operators. In such a way, the system has not only a threat recognition engine but also interpretable AI-driven insights that can be immediately implemented in enterprise and cloud security.

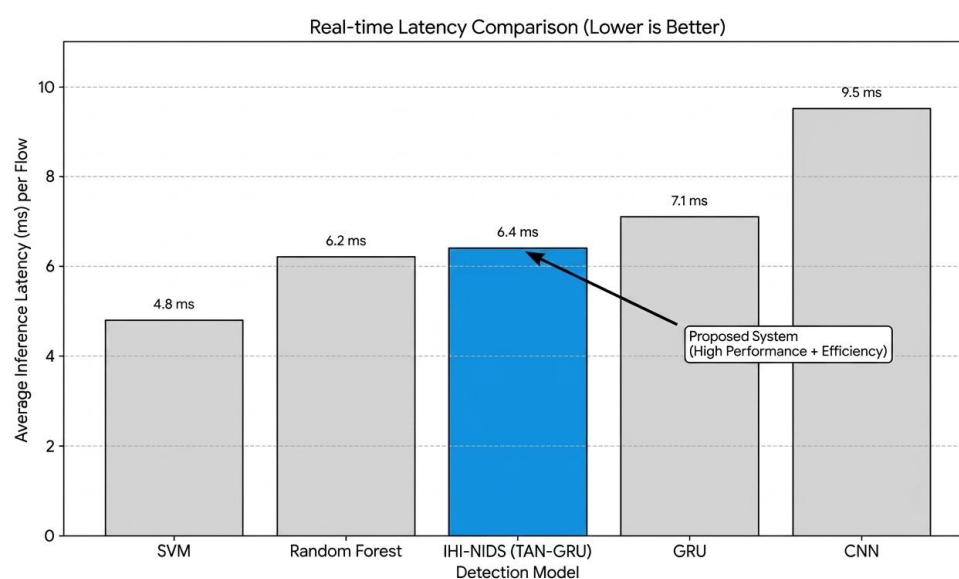


Figure 4. Inference latency comparison across baseline models. The bar chart compares average per-flow inference latency for SVM, Random Forest, CNN, GRU, and IHI-NIDS.

4. Discussion

The experimental findings provide solid empirical evidence of the usefulness of the proposed Intelligent Hybrid-Inference Network Intrusion Detection System (IHI-NIDS) for real-time threat analysis in enterprise and cloud environments. The article is divided into sections focusing on the effectiveness of detection, temporal generalisation, real-time viability, explainability, and compatibility with previous studies [43–45]. According to Table 6, IHI-NIDS has a total detection rate of 97.8, a recall of 96.1, an F1-score of 96.5, and a false alarm rate (FAR) of 2.1 only. This performance is always better than the classical machine learning models (SVM, Random Forest) and those based on deep learning (CNN and GRU-only). The outstanding performance may be explained by the fact that the TAN-GRU architecture allows joint modelling of both temporal dependencies and contextual

relevance. Although classical ML models like SVMs and Random Forests are useful in the case of a fixed feature space, they do not capture sequential traffic dynamics; this is why they have lower recall and higher false alarm rates. The results align with the previous findings by Sheykhmousa et al. [14] and Avci et al. [14], who stated that the conventional ML models are limited in high-dimensional, dynamic settings.

Though CNNs are able to learn spatial features, they do not typically do as well as recurrent architectures, as they do not explicitly learn long-term temporal dependencies. The GRU-only model is better than CNNs in identifying sequential patterns, but it still performs poorly compared to more sophisticated forms of time models. This illustrates that self-attention improves the discriminative ability of temporal architectures, especially when it comes to identifying multi-stage attacks and distributed attacks, which is also in line with the recent literature on attention-based security models [25,26].

IHI-NIDS can be used to rank the globally relevant events in traffic on long time horizons by adding a multi-head self-attention mechanism. The feature will be particularly useful in identifying low-and-slow attacks, including lateral movement and reconnaissance, which are typical features of Advanced Persistent Threats (APTs) as outlined by Sharma et al. [29]. Results of experiments on temporally disjoint test data indicate that IHI-NIDS results in 96.1% recall, with a small decrease in recall compared to the training result. It implies the ability to withstand concept drift and changing patterns of attacks, one of the challenges in the enterprise network security research [30].

The proposed system will be able to detect previously unknown threats, unlike traditional signature-based IDSs, which rely on a set of attack fingerprints that a system must use to detect them. The proposed design option fits the comparative studies of dimensionality reduction algorithms conducted for real-time security applications, showing that PCA features better numerical stability and interpretation than the methods based on an Autoencoder [23,24]. Observably, the results also support the validity of the use of cloud-specific metadata (e.g., VPC ID, security group) to perform effective hybrid intrusion detection in the context of cloud-native infrastructure platforms where, in general, payload inspections are limited. As for the analysis of the inference latency, IHI-NIDS still lowers average latency per flow to 6.4 ms, which is much less than the 10 ms required to be deployed in real time in high-throughput enterprise and cloud networks. Notably, this latency is measured with CPU only, which is one of the realistic operational conditions but not an optimal lab condition. The latency overhead is still low, however, and the architectural choice of employing lightweight attention over GRU outputs can be suggested in lieu of full Transformer stacks, which have been shown to have significant latency overheads [46,47]. This makes IHI-NIDS a viable alternative for deep models, utilising resources; it is unable to do line rate processing, but it can work well [48].

It is important to acknowledge that all experimental results, including latency measurements and throughput estimates, were obtained on a single-node hardware configuration (Intel Core i7-12700K, 32 GB RAM) using CPU-only inference for latency benchmarking. While this setup provides a controlled and reproducible environment for evaluating the core algorithmic performance, it does not fully represent large-scale distributed enterprise deployments that may involve multi-tier architectures, load-balanced inference workers, or edge-cloud collaborative processing. The reported throughput of 156 flows/sec per CPU inference worker demonstrates the per-node efficiency of the TAN-GRU model, but potential communication overheads, network bottlenecks, and synchronisation delays in distributed deployments such as those spanning multiple availability zones or hybrid on-premise/cloud configurations were not measured. Furthermore, the system was not stress-tested under sustained 100 Gbps line-rate conditions or with thousands of concurrent inference workers. These deployment-scale considerations are critical for operational validation and are explicitly identified as a limitation of the current study,

to be addressed in future work through distributed system benchmarking and horizontal scaling experiments.

The SHAP analysis reveals that the data IHI-NIDS leverages to make its prediction is semantically significant, namely flow duration variance, inter-arrival time entropy, bytes-per-packet, and changes in cloud instance behaviour. The features are synthesised as described in the known variables in reconnaissance, volumizing attacks, and the side-ways movement, making the model more credible as an analyst. This is considered one of the significant drawbacks of most deep learning-based IDSs, where they are sometimes termed as black boxes. Adding SHAP to the detection pipeline puts IHI-NIDS as the proposed technology to be aligned with current XAI-based IDS [32,33] and to be operationally adapted in a Zero-Trust Security environment [36].

5. Conclusions, Limitations, and Future Research Directions

The introduction of IHI-NIDS, a new AI-based Intelligent Hybrid-Inference Network Intrusion Detection System, has been presented to analyse threats in real time in enterprise and cloud setups. By incorporating a self-attention mechanism and GRU-based temporal modelling, the proposed framework is effective at capturing both the short-term dynamics of packets and the long-term dependencies of attacks. An end-to-end comparison of the hybrid dataset of 1.2 million network flows shows that IHI-NIDS achieves high precision, recall, and ROC-AUC and has an inference latency of sub-10ms, which is sufficiently low for real-world deployment. The explainable AI integration contributes to more trust and usability, reducing the disconnect between high-performance detection and operational decision-making. In general, the findings affirm that time-attention-based AI models are a feasible and scalable defence against the current hybrid network infrastructure and promise to overcome the limitations of traditional signature-based IDSs. This study has several limitations, despite its strengths. First, whereas the hybrid dataset included both enterprise and cloud traffic, cloud-specific anomalies were identified based on flow-level metadata rather than API-level or application-layer semantics. Second, the test was conducted in supervised learning situations; the system's performance in totally or partially unsupervised conditions was not studied. Third, even though inference latency is appropriate for real-time deployment, large-scale distributed implementations can incur additional communication overheads that are not reflected in the current experimental environment. Future work will aim to develop IHI-NIDS in three directions. Firstly, adversarial robustness against GAN-generated evasion attacks remains entirely unexplored in this study. Future work will systematically evaluate and enhance robustness using adversarial training techniques and ensemble defences, as proposed in recent IDS evasion research—but no such claims are made herein. Second, federated and privacy-preserving learning will be considered to share threat intelligence across organisations without revealing sensitive traffic information. Lastly, it will be integrated with Zero-Trust and API-based security models and will continue to perform behavioural verification across microservices and cloud-native applications.

Author Contributions: Conceptualization, E.I.E.-P. and E.U.; methodology, E.I.E.-P., E.U. and E.G.; validation, E.U., E.G., B.O., V.C. and M.R.Y.; formal analysis, E.I.E.-P., E.U., E.G., B.O., V.C. and M.R.Y.; writing—original draft preparation, E.I.E.-P.; writing—review and editing, E.I.E.-P., E.U., E.G., B.O., V.C. and M.R.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable. The enterprise in the dataset is collected from a publicly available dataset from the CSE-CIC-IDS2018 dataset. The cloud portion of this com-

ponent was made up of private AWS VPC Flow Logs from a properly authorised test environment. No collection or processing of payloads, user content, credentials, or personally identifiable information was done on the packets. IP addresses, instance ID, subnet ID, security group ID, and metadata about cloud regions were anonymized using irreversible hashing before the analysis. The research team only had access to the raw cloud telemetry data and used it only in connection with security research. Since the study was based on network data (not human-subject data), formal informed consent was not required. The telemetry data from the private cloud was, however, authorised for collection and analysis by the owner of the environment. The cloud telemetry data is anonymised and cannot be published as it could reveal operational infrastructure patterns.

Informed Consent Statement: Not applicable.

Data Availability Statement: We used the CSE-CIC-IDS2018 dataset, which is publicly available through the Canadian Institute for Cybersecurity. A private VPC Flow Log component is not publicly acceptable, as it will contain sensitive operational metadata from a provisioned cloud test environment that was authorised to use this component. However, the manuscript provides the steps of pre-processing, the feature schema, pseudocode, and experimental configuration for reproducibility.

Acknowledgments: The authors would like to acknowledge the support of colleagues who provided constructive feedback during the preparation of this manuscript.

Conflicts of Interest: Authors Ehigiatior Iyobor Egho-Promise, Ekereuke Udoh, Editia Gashi and Vijay Chennareddy were employed by the company QA Higher Education, Bamidele Ola was employed by the company Technobeacon and Malleswar Reddy Yerabolu was employed by Rubistone Technologies LLC. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ML	Machine Learning
IDS	Intrusion Detection System
IHI-NIDS	Intelligent Hybrid-Inference Network Intrusion Detection System
GRU	Gated Recurrent Unit
TAN-GRU	Temporal Attention Network-GRU
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
PCA	Principal Component Analysis
SPAN/TAP	Switch Port Analyzer/Test Access Point
PCAP	Packet Capture
NetFlow	Network Flow
IPFIX	IP Flow Information Export
VPC	Virtual Private Cloud
XAI	Explainable AI
SHAP	Shapley Additive Explanations
LIME	Local Interpretable Model-Agnostic Explanation
DoS	Denial of Service
DDoS	Distributed Denial of Service
APT	Advanced Persistent Threat
FL	Federated Learning
ZTA	Zero-Trust Architecture
FAR	False Alarm Rate

IQR	Interquartile Range
TTL	Time to Live

References

- Mishra, R. AI-Driven Network Intrusion Detection Systems: Enhancing Real-Time Threat Detection. *J. Comput. Innov.* **2021**, *1*, 1.
- Sunkara, G. AI-Driven Cybersecurity: Advancing Intelligent Threat Detection and Adaptive Network Security in the Era of Sophisticated Cyber Attacks. *Well Test. J.* **2022**, *31*, 185–198.
- Khalaf, N.Z.; Al Barazanchi, I.I.; Radhi, A.D.; Parihar, S.; Shah, P.; Sekhar, R. Development of Real-Time Threat Detection Systems with AI-Driven Cybersecurity in Critical Infrastructure. *Mesopotamian J. Cybersecur.* **2025**, *5*, 501–513.
- Bagmar, V.; Joshi, D. AI-Driven Real-Time Network Intrusion Detection Using Machine Learning and Deep Learning. In *Proceedings of the 2025 3rd International Conference on Self-Sustainable Artificial Intelligence Systems (ICSSAS), June 2025*; IEEE: New York, NY, USA, 2025; pp. 1569–1574.
- Bellamkonda, S. AI-Driven Threat Intelligence for Real-Time Network Security Optimization. *Technology* **2024**, *15*, 522–534.
- Farooq, M.; Khan, M.H. AI-Driven Network Security: Innovations in Dynamic Threat Adaptation and Time Series Analysis for Proactive Cyber Defense. *Int. J. Wirel. Microw. Technol.* **2024**, *14*, 17–26.
- Lokhandwala, M. AI-Powered Intrusion Detection Systems for Evolving Cyber Threats. *Int. J. Eng. Ext. Technol. Res.* **2025**, *7*, 10555–10558.
- Akbar, R.; Zafer, A. Next-Gen Information Security: AI-Driven Solutions for Real-Time Cyber Threat Detection in Cloud and Network Environments. *J. Cybersecur. Res.* **2024**, *12*, 123–145.
- Raja, M.S.R.S. The Rise of AI-Driven Network Intrusion Detection Systems: Innovations, Challenges, and Future Directions. *Int. J. AI Big Data Comput. Manag. Stud.* **2025**, *1*, 1–10.
- Rao, D.D.; Wao, A.A.; Singh, M.P.; Pareek, P.K.; Kamal, S.; Pandit, S.V. Strategizing IoT Network Layer Security through Advanced Intrusion Detection Systems and AI-Driven Threat Analysis. *Full Length Artic.* **2024**, *12*, 195.
- Ye, Z.; Gao, W.; Hu, Q.; Sun, P.; Wang, X.; Luo, Y.; Zhang, T.; Wen, Y. Deep Learning Workload Scheduling in GPU Datacenters: A Survey. *ACM Comput. Surv.* **2024**, *56*, 1–38.
- Careem, R.; Johar, G.; Khatibi, A. Deep Neural Networks Optimization for Resource-Constrained Environments: Techniques and Models. *Indones. J. Electr. Eng. Comput. Sci.* **2024**, *33*, 1843–1854.
- Diban Armendariz, X.I. Managing False Positives in SOC Operations: Solutions and Best Practices. Bachelor's Thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, 2025.
- Sheykhmousa, M.; Mahdianpari, M.; Ghanbari, H.; Mohammadimanesh, F.; Ghamisi, P.; Homayouni, S. Support Vector Machine versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 6308–6325.
- Dabija, A.; Kluczek, M.; Zagajewski, B.; Raczko, E.; Kycko, M.; Al-Sulttani, A.H.; Tardà, A.; Pineda, L.; Corbera, J. Comparison of Support Vector Machines and Random Forests for CORINE Land Cover Mapping. *Remote Sens.* **2021**, *13*, 777.
- Avci, C.; Budak, M.; Yağmur, N.; Balçık, F. Comparison between Random Forest and Support Vector Machine Algorithms for LULC Classification. *Int. J. Eng. Geosci.* **2023**, *8*, 1–10.
- Samanta, R.K.; Sadhukhan, B.; Samaddar, H.; Sarkar, S.; Koner, C.; Ghosh, M. Scope of Machine Learning Applications for Addressing the Challenges in Next-Generation Wireless Networks. *CAAI Trans. Intell. Technol.* **2022**, *7*, 395–418.
- He, Y.; Huang, P.; Hong, W.; Luo, Q.; Li, L.; Tsui, K.L. In-Depth Insights into the Application of Recurrent Neural Networks (RNNs) in Traffic Prediction: A Comprehensive Review. *Algorithms* **2024**, *17*, 398.
- Kansara, M. Cloud Migration Strategies and Challenges in Highly Regulated and Data-Intensive Industries: A Technical Perspective. *Int. J. Appl. Mach. Learn. Comput. Intell.* **2021**, *11*, 78–121.
- Bilot, T.; El Madhoun, N.; Al Agha, K.; Zouaoui, A. Graph Neural Networks for Intrusion Detection: A Survey. *IEEE Access* **2023**, *11*, 49114–49139.
- Xie, J.; Yin, W.; Wang, L. Achieving Flexible, Low-Latency and 100Gbps Line-Rate Load Balancing over Ethernet on FPGA. In *Proceedings of the 2020 IEEE 33rd International System-on-Chip Conference (SOCC), September 2020*; IEEE: New York, NY, USA, 2020; pp. 201–206.
- Wang, C.; Liu, H.; Sun, Y.; Wei, Y.; Wang, K.; Wang, B. Dimension Reduction Technique Based on Supervised Autoencoder for Intrusion Detection of Industrial Control Systems. *Secur. Commun. Netw.* **2022**, *2022*, 5713074.
- Cacciarelli, D.; Kulahci, M. Hidden Dimensions of the Data: PCA vs. Autoencoders. *Qual. Eng.* **2023**, *35*, 741–750.

24. Bai, Y.; Sun, M.; Zhang, L.; Wang, Y.; Liu, S.; Liu, Y.; Tan, J.; Yang, Y.; Lv, C. Enhancing Network Attack Detection Accuracy through the Integration of Large Language Models and Synchronized Attention Mechanism. *Appl. Sci.* **2024**, *14*, 3829.
25. Galassi, A.; Lippi, M.; Torroni, P. Attention in Natural Language Processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4291–4308.
26. Epee Pense, F.S.; Dourdam silé, B. Machine Learning for Secure and Intelligent Information and Communication Technologies: A Comprehensive Review. *Machine Learning for Secure and Intelligent Information and Communication Technologies: A Comprehensive Review. SSRN* **2025**. <https://doi.org/10.2139/ssrn.5770602>.
27. Zumarah, B.; Rachman, B.; Mohandes, M.; Al-Shaikhi, A. Optimised Transformer and GRU Models for Forecasting Lost Circulation Volume in Drilling Operations. *IEEE Access* **2025**, *13*, 153918–153936.
28. Reza, S.; Ferreira, M.C.; Machado, J.J.M.; Tavares, J.M.R. Road Traffic Events Monitoring Using a Multi-Head Attention Mechanism-Based Transformer and Temporal Convolutional Networks. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 13011–13024.
29. Sharma, A.; Gupta, B.B.; Singh, A.K.; Saraswat, V.K. Advanced Persistent Threats (APT): Evolution, Anatomy, Attribution and Countermeasures. *J. Ambient Intell. Humanize. Comput.* **2023**, *14*, 9355–9381.
30. Lyu, M.; Gharakheili, H.H.; Sivaraman, V. A Survey on Enterprise Network Security: Asset Behavioral Monitoring and Distributed Attack Detection. *IEEE Access* **2024**, *12*, 89363–89383.
31. Trilho, P.C.P.O. Intelligent Systems for Cyber Defence—An Architecture Framework for Cyber Defence Using Artificial Intelligence. Master's Thesis, Universidade NOVA de Lisboa, Lisbon, Portugal, 2022.
32. Gaspar, D.; Silva, P.; Silva, C. Explainable AI for Intrusion Detection Systems: LIME and SHAP Applicability on Multi-Layer Perceptron. *IEEE Access* **2024**, *12*, 30164–30175.
33. Uysal, I.; Kose, U. Analysis of Network Intrusion Detection via Explainable Artificial Intelligence: Applications with SHAP and LIME. In *Proceedings of the 2024 Cyber Awareness and Research Symposium (CARS), October 2024*; IEEE: New York, NY, USA, 2024; pp. 1–6.
34. Allafi R.; Alshahrani A.; Arasi MA.; Alshahrani H.; AlAqil M.A.; Alabdan R.; Zalah I.; Alharbi RM. Explainable Machine Learning Framework for Real-Time Multi-Attack Threat Detection in Edge-Enabled VANET Environments. *Trans. Emerg. Telecommun Technol.* **2025**, *36*, e70283. <https://doi.org/10.1002/ett.70283>.
35. Akinyemi, A. Zero Trust Security Architecture: Principles and Early Adoption. *Int. J. Technol. Manag. Humanit.* **2022**, *8*, 11–22.
36. Ghasemshirazi, S.; Shirvani, G.; Alipour, M.A. Zero Trust: Applications, Challenges, and Opportunities. *arXiv* **2023**, arXiv:2309.03582.
37. Paidy, P.; Chaganti, K. Securing AI-Driven APIs: Authentication and Abuse Prevention. *Int. J. Emerg. Res. Eng. Technol.* **2024**, *5*, 27–37.
38. Paul, J. Building a Robust API Security Framework with Machine Learning. 2024. Available online: https://www.researchgate.net/publication/385588341_Building_a_Robust_API_Security_Framework_with_Machine_Learning (accessed on 2 July 2026).
39. Pawlicki, M.; Uccello, F.; D'Antonio, S.; Kozik, R.; Choraś, M. A Novel Method of Improving Intrusion Detection Systems Robustness against Adversarial Attacks through Feature Omission and a Committee of Classifiers. In *Proceedings of the European Symposium on Research in Computer Security, September 2024*; Springer Nature: Cham, Switzerland, 2024; pp. 273–288.
40. Moamin, S.A.; Abdulhameed, M.K.; Al-Amri, R.M.; Radhi, A.D.; Naser, R.K.; Pheng, L.G. Artificial Intelligence in Malware and Network Intrusion Detection: A Comprehensive Survey of Techniques, Datasets, Challenges, and Future Directions. *Babylon. J. Artif. Intell.* **2025**, *2025*, 77–98.
41. Belcastro, L.; Marozzo, F.; Orsino, A.; Talia, D.; Trunfio, P. Navigating the edge-cloud continuum: A state-of-practice survey. *IEEE Access* **2026**, *14*, 40622–40647.
42. Ndayipfukamiye, T.; Ding, J.; Sarwatt, D.S.; Philipo, A.G.; Ning, H. Adversarial Defense in Cybersecurity: A Systematic Review of GANs for Threat Detection and Mitigation. *arXiv* **2025**, arXiv:2509.20411.
43. Villafranca, A.; Cano, M.D. A hybrid inference pipeline for IDS: Combining DNNs and XGBoost through stacking for real-world intrusion detection. *Ad Hoc Netw.* **2026**, *188*, 104227.
44. Haider, Z.A.; Zeb, A.; Rahman, T.; Khan, F.M.; Khan, I.U.; Sohail, Q.; Bilal, H.; Khan, M.A.; Ullah, I. Optimizing cloud security with a hybrid BiLSTM-BiGRU model for efficient intrusion detection. *ICCK Trans. Sens. Commun. Control.* **2025**, *2*, 106–121.
45. Penaganti, R. Security-Trust-Determinism Co-Design Using Hybrid Intrusion Detection with Temporal Modeling for Real-Time Publish-Subscribe Middleware. *IEEE Commun. Stand. Mag.* **2026**, *Early Access*. <https://doi.org/10.1109/MCOMSTD.2026.3676622>.

46. Hafizi, A.; Yusof, Z.B. Efficient Data Integration Strategies for Heterogeneous Big Data Sources in Cloud Environments. *Int. J. Data Sci. Big Data Anal. Predict. Model.* **2024**, *14*, 1–15.
47. Nehra, D.; Mangat, V.; Kumar, K. A Deep Learning Approach for Network Intrusion Detection Using Non-Symmetric Auto-Encoder. In *Intelligent Computing and Communication Systems*; Springer: Singapore, 2021; pp. 371–382.
48. Rozony, F.Z.; Aktar, M.N.A.; Ashrafuzzaman, M.; Islam, A. A Systematic Review of Big Data Integration Challenges and Solutions for Heterogeneous Data Sources. *Acad. J. Bus. Adm. Innov. Sustain.* **2024**, *4*, 1–18.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.