International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Edge-Aware GATv2-BiLSTM Framework for Spatiotemporal Intrusion Detection in Autonomous UAV Command-and-Control Networks

Jury Osama Balgoon[1], Mohammed Omar Baz[2]

[1] Department of Computer Engineering, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia

Submitted: 01/07/2026 Revised: 15/08/2026 Accepted: 23/08/2026

Abstract: Securing Unmanned Aerial Vehicle (UAV) networks against complex cyber threats is critical for ensuring the operational safety of autonomous command-and-control systems. Traditional intrusion detection systems (IDS) often treat traffic flows as independent records, limiting their ability to capture both relational communication dependencies and temporal attack evolution. For that purpose, this paper presents a hybrid framework that combines the ability of Graph Neural Networks (GNNs) to represent complex network structures with the power of Bidirectional Long Short-Term Memory (BiLSTM) units for learning sequential communication patterns. The proposed hybrid model architecture consists of two main parts: a deep graph encoder that encodes raw UAV communication flows as high-dimensional structural representations through edge-aware Graph Attention Network v2 (GATv2) layers, and a BiLSTM sequential framework that links spatial embeddings to a 10-step temporal window. The framework learns spatiotemporal representations that model cyber-threat evolution in both forward and backward temporal contexts. The framework was evaluated on the simulated UAVIDS-2025 ad hoc network dataset and the real T-ITS cyber-physical UAV Wi-Fi dataset. On UAVIDS-2025, the proposed GNN-BiLSTM + XGBoost pipeline achieved 98.05% accuracy and 98.18% F1-score, outperforming the strongest reported tabular XGBoost baseline by 2.05% and 2.18% points, respectively. On the T-ITS cyber dataset, the AE + GNN + BiLSTM pipeline achieved 99.16% multi-class accuracy and 99.57% weighted F1-score under the reported split. These results indicate that jointly modeling graph-based communication structure and temporal dependencies can improve UAV intrusion detection performance across simulated and real cyber-physical UAV communication settings.

Keywords: *Graph Neural Networks (GNN), Bidirectional LSTM, UAV Cybersecurity, Spatiotemporal Learning, and Intrusion Detection Systems.*

1. Introduction

Unmanned Aerial Vehicles (UAVs) are no longer stand-alone aerial platforms, but have emerged as networked cyber-physical systems used in surveillance, logistics, disaster response, traffic monitoring, precision agriculture, and military applications. These applications rely on wireless command-and-control (C2) links, on-board sensing, autonomous decision modules, and inter-UAV communication for telemetry, control, payload data, and mission status. This connectivity brings more flexibility for operation, but it also introduces ways to attack a UAV network, which can result in violations of confidentiality, integrity, availability, and flight safety [1]. The open wireless operating medium, together with the limited computation, storage, and energy of UAVs, makes it easier to launch denial-of-service, replay, spoofing, false data injection, jamming, hijacking, and route manipulation attacks [2].

An Intrusion Detection System (IDS) is thus a crucial security layer when autonomous UAV flight is considered. In UAV environments, traditional signature-based IDS systems are hard to keep up to date due to changing attack signatures, topology changes, and the resemblance between specific attacks and legitimate traffic. This has been improved by the use of machine learning and deep learning algorithms that learn discriminative representations from traffic, sensor, and cyber-physical features. Previous studies have explored ML-based IDSs for UAV networks [1] and recurrent models such as LSTM and BiLSTM for temporal dependencies in drone-network traffic [3]. Cyber-physical UAV datasets also indicate that integrating cyber and physical evidence can enhance detection capability in the presence of DoS, replay, evil twin, and false data injection attacks [2].

However, many current UAV IDS strategies continue to consider traffic records as independent feature vectors [4]. This assumption is a limitation for UAV C2 systems, as malicious behavior can appear not only in individual packets or flows, but also in the relationships between UAVs, ground stations, access points, and communication paths. By nature, network traffic is relational, and Graph Neural Networks (GNNs) are capable of learning from such structures by propagating information among neighboring entities and encoding node and edge interactions [5] [6]. Therefore, Graph Attention Network v2 (GATv2), which extends standard graph attention with dynamic attention, is appropriate for heterogeneous UAV communication patterns [7].

The second drawback of pure graph-based detection is that a static graph snapshot cannot fully reflect the evolution of attacks. UAV attacks can be highly incremental, while BiLSTM, when used alone, cannot directly model communication topology. This encourages a hybrid spatiotemporal IDS that learns graph-structured relations and short temporal dynamics simultaneously.

This study addresses three research questions:

RQ1: Does replacing sequential-only learning with edge-aware graph modeling benefit UAV intrusion detection?

RQ2: Does combining edge-aware graph modeling with BiLSTM temporal modeling and optimized tree-based/deep classification heads (such as XGBoost) improve detection performance compared with graph-only and sequence-only baselines?
RQ3: Can the proposed scheme be generalized to simulated traffic of UAVs in ad hoc networks and actual UAV Wi-Fi cyber-physical traffic?
This paper has four main contributions. First, it considers an edge-aware graph construction strategy, which maintains the connectivity and dependency of distributed UAV entities. Second, it constructs a hybrid GATv2-BiLSTM architecture augmented with Autoencoders (AE) and XGBoost meta-classifiers to perform spatiotemporal intrusion detection. Third, it provides a modular open framework where graph encoding, dimensionality reduction, and classification heads can be tuned to data that contains differently structured graphs. Finally, it confirms the proposed framework over two different heterogeneous UAV cybersecurity datasets covering simulated UAV network attacks and real cyber-physical UAV communication attacks.
We have structured the rest of the paper as follows. Section 2 presents related work in the field of UAV intrusion detection and graph-based learning. The proposed methodology along with datasets, preprocessing, graph construction and model architecture are given in Section 3. In Section 4 the experimental results are reported and discussed. Finally, the paper is concluded in Section 5 where the future research directions are given.

2. Literature Review

Recent studies around UAV intrusion detection have focused on conventional machine learning models, autoencoder-based feature extraction, recurrent learning, attention mechanisms, and UAV-specific datasets. Nevertheless, current methods still rely mainly on tabular records or temporal sequences, while the relational structure of UAV communication networks remains weakly represented. In this section, 5 closely related studies were reviewed in terms of their relevance to the proposed spatiotemporal graph-based IDS.
Zeng et al. [8] presented UAVIDS-2025, an intrusion detection benchmark dataset for UAV-specific aerial ad hoc networks. The dataset contains 122,171 flow-level records, 22 features, normal traffic, and four network-layer attacks: blackhole, flooding, Sybil, and wormhole. Several classical and deep learning models were evaluated, where XGBoost achieved the best accuracy and F1-score of 96%. This work is important because it provides a UAV-specific IDS dataset; however, the evaluated models treat flows as feature vectors or sequences and do not explicitly use the graph topology of UAV entities.
Vuong et al. [9] proposed an autoencoder-based intrusion detection framework for UAV communication using a real UAV cyber dataset. The autoencoder maps cyber features into low-dimensional latent representations, which are then used by conventional machine learning classifiers. The Decision Tree achieved the highest F1-score of 96.72% in binary classification, while MLP achieved 84.09% in multi-class classification. However, the method remains a two-stage tabular learning process and does not capture graph relationships or attack evolution over time.
Ramadan et al. [10] proposed a deep learning framework for Internet of Drones intrusion detection using LSTM-RNN modules at the drone and base-station levels. The framework was tested on several standard network intrusion datasets and achieved competitive accuracy of approximately 95% to 97% in some settings. This work is relevant because it considers temporal modeling in distributed drone networks. However, the evaluation is not based on UAV-specific traffic, and the model does not represent the connections between UAVs, ground stations, and communication links.
Kou et al. [11] proposed an intrusion detection model for drone communication in a software-defined networking environment. Their model combines a deep autoencoder, CNN, and attention mechanism for feature reduction, pattern extraction, and feature weighting. The model achieved 99.7% accuracy in binary classification and 95.5% accuracy in multi-class classification on the InSDN dataset. However, the dataset is SDN-focused and does not include physical or ad hoc UAV communication scenarios for autonomous UAV C2 networks.
Aldossary et al. [12] proposed a Cross-Layer Convolutional Attention Network (CLCAN) for intrusion detection in drone-to-drone and drone-to-base-station communications. The model uses cross-layer feature learning, contextual attention, and dynamic feature fusion to detect DDoS attacks, malware infection, and communication anomalies. Reported results show 98.4% accuracy, 98.7% recall, 98.1% F1-score, and 0.991 AUC. Although the method performs well and uses attention, it still does not model UAV entities and communication flows as dynamic edge-aware graphs.
The reviewed studies show significant progress in UAV and drone-network intrusion detection through UAV-specific datasets, autoencoders, recurrent models, and attention mechanisms. However, communication records are still commonly modeled as separate feature vectors or sequential samples, which limits their ability to represent relational UAV communication structures. Even UAVIDS-2025 was mainly evaluated using classical ML and standard deep-learning models without considering the graph structure of UAV communication flows.
Another gap is joint spatial-temporal attack behavior modeling. Autoencoders, LSTM-based models, and attention mechanisms have been used, but they are often implemented separately or without structural UAV topology. To fill these gaps, the proposed work uses edge-aware graph representations, GATv2 layers for spatial communication dependencies, and BiLSTM-based temporal modeling to capture attack evolution. This allows the IDS to model both who communicates with whom and how malicious behavior evolves over time.
Table 1 summarizes the key related studies and their main limitations.

3. Methodology

3.1. Methodology Overview

The presented approach follows a pipeline-based model and operates at the level of network flows. The raw traffic data is initially converted into a relational graph that can be used for message passing. Then, graph attention layers learn contextual node representations, while short temporal correlations are accounted for through a bidirectional recurrent block processing short sliding windows of edge representations. The framework was tested on two datasets with different feature spaces and topologies: UAVIDS-2025 [8], which contains simulated multi-UAV ad hoc traffic, and T-ITS [2], which contains real UAV Wi-Fi cyber-physical traffic.
The proposed intrusion detection framework is reported in Fig. 1.

Table 1. Summary of Related Work

Study	Study Focus	Dataset	Main Method	Main Strength	Main Limitation
[8]	UAV-specific IDS benchmark	UAVIDS-2025	Ten ML/DL models, including XGBoost and LSTM	Provides a dedicated UAV network IDS dataset with four attack classes	Does not exploit graph topology explicitly
[9]	AE-based UAV cyber IDS	Actual UAV cyber dataset	Autoencoder + DT/RF/KNN/MLP/SVM	Improves compact feature extraction over SHAP-based baselines	No graph learning or temporal edge modeling
[10]	IoD deep IDS	KDDCup99, NSL-KDD, UNSW-NB15, Kyoto, CICIDS2017, TON-IoT	Distributed LSTM-RNN	Models temporal patterns at drone and base-station levels	Uses general IDS datasets rather than UAV-specific traffic
[11]	Drone SDN IDS	InSDN	DAE + CNN + attention	High accuracy and fast convergence	Dataset is SDN-focused, not UAV-specific ad hoc C2 traffic
[12]	Cross-layer drone IDS	Drone communication dataset	CLCAN with convolutional attention	Strong cross-layer detection performance	Does not model UAV flows as edge-aware dynamic graphs

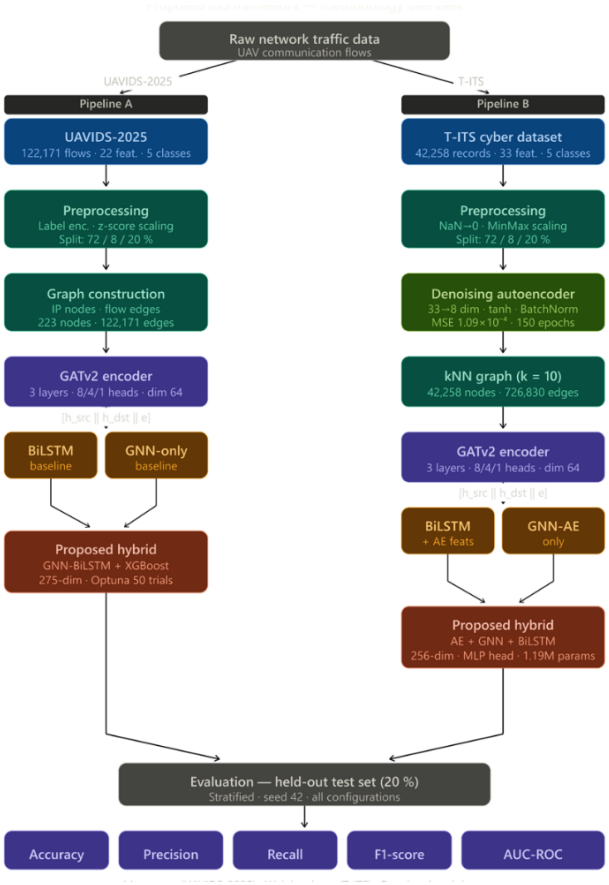

Fig. 1. System methodology

The diagram depicts two parallel pipelines executing the same evaluation protocol. Pipeline A takes the UAVIDS-2025 data and follows a graph path, where nodes are IP addresses and edges are network flows. Pipeline B passes the T-ITS cyber data through an autoencoder-mediated path and builds a k-nearest neighbors graph in the latent space. Both pipelines share preprocessing, stratified train-validation-test splitting, and a 3-layer GATv2 graph encoder with 8, 4, and 1 attention heads and a hidden width of 64.

The pipelines also have different classification heads. Pipeline A passes the 275-dimensional concatenation of the source embedding, destination embedding, and edge features to an Optuna-tuned XGBoost model selected after 50 trials. The multi-layer perceptron head of Pipeline B, with ~1.19M trainable parameters, takes the 256-dimensional BiLSTM output as input.

To isolate the contribution of an individual architectural component, two ablation pipelines are presented: one is a BiLSTM-only baseline, and the other is a graph-only baseline. Both pipelines lead to the same evaluation step on the held-out test set which is 20 per cent of each data set, and the results are reported in 5 measurements including accuracy, precision, recall, F1, and AUC-ROC.

3.2. Datasets

3.2.1. UAVIDS-2025 Dataset

UAVIDS-2025 [8] was obtained using the wireless network simulator NS-3.24 based on IEEE 802.11ac wireless protocol. The Ad hoc On-Demand Distance Vector routing protocol and a BOIDswarm mobility model, emulating cooperative behavior of small UAV fleets, were implemented in the simulator. The final set in the corpus has 122,171 flow records, which are categorized in five classes such as Normal Traffic, Blackhole Attack, Wormhole Attack, Sybil Attack, and Flooding Attack. There are 22 features coded in each record that are divided into three groups. The connection metadata includes source and destination addresses, source and destination ports, the protocol ID, and the flow duration. Traffic-volume metrics include the number of packets and bytes transmitted and received along with rates. Performance indicators provide end to end delay, jitter, throughput, a measurement of the average number of hops the packet traverses and the packet drop ratio. The dataset captures the network behavior for the small UAV ad hoc networks under benign as well as adversarial situations and thus is ideal for evaluating intrusion detection methods based on graphs. The class distribution of the UAVIDS-2025 is reported in Fig. 2.

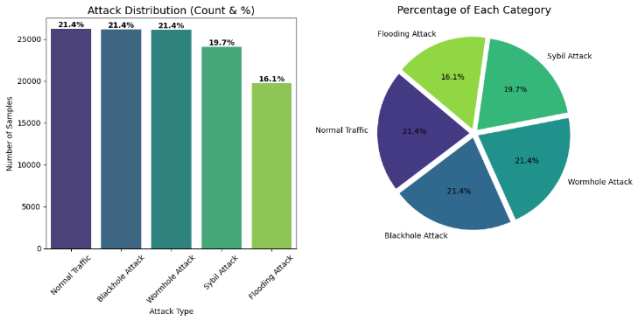

Fig. 2. Class distribution of the UAVIDS-2025 dataset.

Normal Traffic, Blackhole Attack, and Wormhole Attack each account for approximately 21% of the corpus and are each about

21% of the corpus approximately. The percentages of records with "Sybil Attack" and "Flooding Attack" are 19.7% and 16.1%, respectively. While not perfectly balanced, the five classes are moderately balanced in comparison to many public network traffic datasets. The different types of attack categories make it possible to assess the detection result for the volumetric attacks (Flooding) as well as the stealthy attacks caused by the manipulation of routing in a wireless ad hoc network (Blackhole and Wormhole attacks).

Table 2. Characteristics of the datasets used to assess the model.

Learning Task	Dataset	Class	Dimension	Train		Test		Class Ratio (%)
				Entities	Size	Entities	Size	
Multi-class Edge Classification	UAVIDS-2025	Normal Traffic	22	223	18,844	223	5,234	21.4
		Blackhole Attack	22	223	18,799	223	5,222	21.4
		Wormhole Attack	22	223	18,782	223	5,217	21.4
		Sybil Attack	22	223	17,335	223	4,815	19.7
		Flooding Attack	22	223	14,202	223	3,945	16.1
	T-ITS	Benign	33	42,258	6,786	42,258	1,885	22.3
		DoS Attack	33	42,258	8,403	42,258	2,334	27.6
		Replay	33	42,258	8,644	42,258	2,401	28.4
		Evil Twin	33	42,258	4,092	42,258	1,137	13.4
		FDI	33	42,258	2,500	42,258	695	8.2

3.2.2. T-ITS Cyber Dataset

The T-ITS dataset [2] consists of actual cyber physical intrusion benchmark collected from operational UAV communications operating following the IEEE 802.11 Wi-Fi standard. There are 42258 records in the corpus, which are divided into 5 classes: Benign, DoS Attack, Replay, Evil Twin, False Data Injection (FDI). There are 39 columns in the original feature space. Six of the fields, namely the frame number, the wlan.bssid field, the raw timestamp, the receiver address field, the transmitter address and the fields that were not part of the Wireshark KCl template were discarded, resulting in 33 numerical attributes available for downstream processing. The following fields are retained from the packet: length, time delta, frames protocols, several WLAN specific fields, and a number of physical layer indicators. The class distribution for the T-ITS data is presented in the Fig. 3.

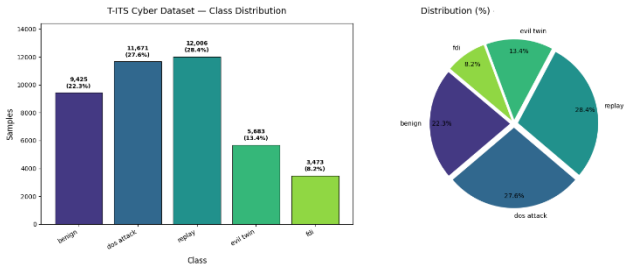

Fig. 3. Class distribution of the T-ITS cyber dataset.

The two classes, Replay and DoS Attack, together account for 56% of the records, whereas Benign takes 22.3%. The proportion of classes Evil Twin (13.4%) and FDI (8.2%) is very low. This imbalance will put more pressure on the optimization process as without class aware weighting, the model will be overfitting to the two major attack types. Additionally, the T-ITS data is observed to come at a different numerical scale for each of the features, which further impelled the utilization of a denoising autoencoder front and a min max rescaler.

3.2.3. Datasets Characteristics

Table 2 summarizes the characteristics of the two datasets as summarized by the same descriptors. The table reports the learning task, dataset, class labels, feature dimensions, training and test sizes, graph entities, and class ratios. UAVIDS-2025 graph combines traffic from 223 different IP addresses and results in comparatively small number of nodes, but rather dense graph. The T-ITS graph consists of one node for each T-ITS record, creating 42,258 nodes linked by a k nearest neighbor graph in the latent space of the autoencoder as described in more detail in Table 2.

3.3. Data Preprocessing and Graph Construction

3.3.1. UAVIDS-2025 Preprocessing

The raw UAVIDS-2025 records were first cleaned and then the label column is encoded as integer index (label encoding).

The Protocol field was modified from a textual string to a number field. Two new columns, src_node_idx and dst_node_idx, were added, as a node look up dictionary from the source/destination IP address. Each distinct IP address is then assigned to a node index by assigning it a contiguous node index in the dictionary, resulting in 223 nodes in total. The data were split into the three sets, train, validation, and test, of 72%, 8%, and 20% respectively, with stratification according to class label ensuring that the class distribution in the splits is similar to that of the whole data set.

A communication graph with a mix of node types was then constructed. Each unique IP address is a node and each network flow is a directed edge. Per node statistics were merged from all incident flows: the in degree, the out degree, the mean of transmitted bytes, the standard deviation of transmitted bytes, the average rate of transmitted and received bytes, the average delay, the average jitter, the average throughput, average received drop rate and the average number of hops. Numerical attributes were z score transformed with only the training partition considered. Edge features are the 19 numerical flow attributes (after the same standardization step). The two objects x and the edge_index are following the PyTorch Geometric Data convention for the graph object, that is, x has shape (223, 12), edge_index has shape (2, 122171), and edge_attr has shape (122171, 19). The class labels are attached at the edge level, thus making it a multi class edge classification problem.

3.3.2. T-ITS Preprocessing

The T-ITS data was cast to numeric values columnwise, non-numeric tokens were converted to NaN values and subsequently replaced with zero. Benign, DoS Attack, Replay, Evil Twin, FDI were simplified to integers indices of 0, 1, 2, 3, 4 respectively. Furthermore, a binary label has been created such that label 1 has been given to any non-Benign record. Stratified sampling was employed to split the dataset in 72% training, 8% validation and 20% test. The feature columns were normalized to the range [0, 1]

using the min max scaler, which was trained on the training partition only.

A denoising autoencoder was added before the graph building step. The autoencoder consists of a symmetrically encoded and decoded network, three hidden layers (encoder/decoder), 64-32-32 hidden, tanh activation, batch normalization, and 10% dropout. The gradient flow through the reconstruction was preserved with a 'residual short cut' path from the input layer to the bottleneck. The bottleneck size was set to N = 8 and this gave a compression ratio of 24.2 per cent. This autoencoder was trained using Adam optimizer for 150 epochs, at an initial learning rate of 1 × 10-3, a batch size of 256 and an early stopping of 25 epochs, based on the validation mean squared error (MSE). The training and validation MSE curves for the 150-epoch budget are presented in Fig. 4. The final validation error is 1.09×10^{-4} which is indicative of the fact that the bottleneck representation is able to capture the dominant variation of the input space.

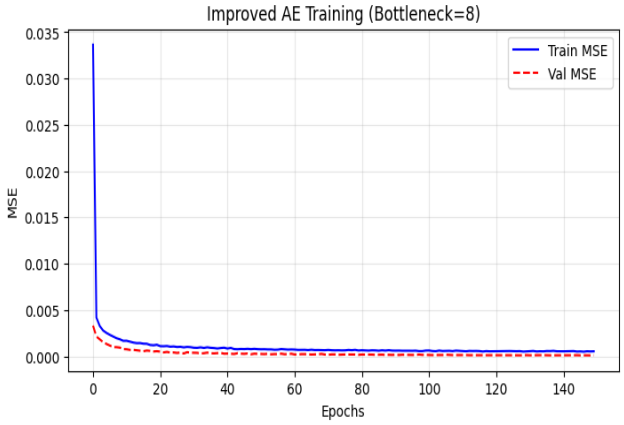

Fig. 4. Training and validation Mean Squared Error of the denoising autoencoder used as a feature compressor for the T-ITS dataset.

After convergence of the autoencoder, each sample was encoded to its latent embedding space using the 8-dimensional latent representation. Then, in the latent space, a 10-neighbour graph was built, using cosine distance as similarity metric. Fitting the k nearest neighbor index was done only on the training latent embeddings, while the validation and test latent embeddings were memory-efficiently queried against the frozen index. The resulting graph has 42,258 nodes and 726,830 training, validation and test directed edges consisting of 608,500 training edges, 33,810 validation edges and 84,520 test edges. The features of nodes are the latent embeddings in dimension 8 and the edge features are the concatenation of latent embeddings of two endpoints with dimension 16. The classification is done at the edge level, taking over the label of the source node as the label for the edge. This construction helps avoid test leakage and separates KNN index fitting from validation and test querying.

3.4. AI Models

This section describes the main components used to construct the pipeline. The graph attention encoder, bidirectional LSTM, denoising autoencoder and the XGBoost head are introduced in turn as well as Focal Loss objective. For each block, a short description is included detailing the operations it contains, and the equations that govern its forward pass.

3.4.1. Graph Attention Network v2

The major building block of the framework is the Graph Attention Network v2 (GATv2) [13] used for feature extraction. It updates an initial node feature matrix to get the contextual embedding by dynamically attending to the node's local neighborhood. For a node i with h_i, and an edge (j, i) of features q_{ij}, e_{ij} is measured as in (1), and the normalized attention weight α_{ij} is measured using a softmax over N(i) as in (2). The new node feature is the weighted sum of neighbor features based on (3) after transformation.

$$e_{ij} = a^T \text{ LeakyReLU}(W [h_i \parallel h_j \parallel q_{ij}]) \quad (1)$$

$$\alpha_{ij} = \exp(e_{ij}) / \Sigma_{k \in N(i)} \exp(e_{ik}) \quad (2)$$

$$h'_I = \sigma(\Sigma_{j \in N(i)} \alpha_{ij} W h_j) \quad (3)$$

In the above equations W is a learnable linear projection, a is a learnable attention vector, ‖ is vector concatenation and σ is a non linear activation function. We design the GATv2 attention as a dynamic manner because the score e_{ij} is dependent on source, destination, and edge feature to bring in more expressive power than the fixed attention in the original GAT formulation [13] and [14]. The architecture consists of three GATv2 layers (8, 4, 1 attention heads and 64 hidden width). In the third layer, a residual projection links the input feature space to the output feature space: to achieve this, a residual is added to the initial component. The multi head case is simply an elaboration of (3) as a sequence of K parallel attention calculations, as in (4).

$$H'_I = \|^K_{k=1} \sigma(\Sigma_{j \in N(i)} \alpha^{(k)}_{ij} W^{(k)} h_j) \quad (4)$$

3.4.2. Bidirectional Long Short-Term Memory

Long Short-Term Memory (LSTM) is a cell with three gates and with a cell state that determines the flow of information between the time steps x_1, x_2, ..., x_T. The input gate i_t, the forget gate f_t and output gate o_t are computed as in (5), the candidate cell content as in (6), new cell state as in (7) and hidden state as in (8) at step t [15].

$$I_t = \sigma(W_i [h_{t-1}, x_t] + b_i) \quad (5a)$$

$$f_t = \sigma(W_f [h_{t-1}, x_t] + b_f) \quad (5b)$$

$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o) \quad (5c)$$

$$\tilde{c}_t = \tanh(W_c [h_{t-1}, x_t] + b_c) \quad (6)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (7)$$

$$h_t = o_t \odot \tanh(c_t) \quad (8)$$

The bidirectional variant involves running two passes (forward and backward) of LSTM simultaneously and concatenates the forward and backward hidden states at each time step (like (9)). The framework is composed of two layer BiLSTM with hidden size of 128 and dropout of 0.3, so that it will have an output size of 256 for each time step. This learnable attention pooling layer maps the sequence into a fixed-length vector that is used to represent the temporal context in the window of 10 edge embeddings [15].

$$H_t = [h^{\rightarrow}{}_t \parallel h^{\leftarrow}{}_t] \quad (9)$$

3.4.3. Denoising Autoencoder

The denoising autoencoder is a low dimensional representation that minimizes the reconstruction error between the input and the denoising autoencoder's decoded output. Let x be an input record, $x \in \mathbb{R}^{33}$. Define the encoder function f_{enc} (θ_e), and the decoder function f_{dec} (θ_d). The bottleneck representation and the reconstruction are calculated as in (10) and (11), and the training criterion is the mean squared error between it and the input (12) [16].

$$Z = f_{enc}(x, \theta_e) \quad (10)$$

$$\hat{X} = f_{dec}(z, \theta_d) \quad (11)$$

$$\mathscr{L}_{AE} = (1/N)\ \Sigma_{i=1}^{N} \|x_i - \hat{x}_i\|^2_2 \quad (12)$$

A small amount of Gaussian noise is injected into the input during training, compelling the network to learn robust latent representations that capture the underlying data manifold rather than memorizing individual input patterns. The encoder has three fully connected layers with hidden widths of 64, 32, and 8, as a bottleneck. The decoder is set up the same. Each hidden layer consists of tanh activation and batch normalization, while there is a residual short cut between the input layer and the bottleneck. The decoder output layer is assigned with sigmoid, as the min max rescaler gives the range [0,1] and the decoder generates a same range from this.

3.4.4. XGBoost Classifier

XGBoost is a scalable implementation of a gradient boosting framework that fits an additive model of decision trees while minimizing a regularized loss function. The sum of the K decision tree outputs f_k belonging to the space of regression trees F gives the predicted score for an input x_i as written in (13). The objective is the sum of a task specific loss l and a regularization penalty Ω over each tree as given in (14) [17].

$$\hat{Y}_i = \Sigma_{k=1}^{K} f_k(x_i),\ f_k \in F \quad (13)$$

$$Obj = \Sigma_{i=1}^{N} l(y_i, \hat{y}_i) + \Sigma_{k=1}^{K} \Omega(f_k) \quad (14)$$

$$\Omega(f) = \gamma T + (1/2)\ \lambda\ \|w\|^2 \quad (15)$$

The penalty in (15) is used to tune the number of leaves T via parameter γ and the L2 norm of the leaf weights w via parameter λ. Each boosting pass, the tree is trained to fit the negative gradient of the loss function with respect to the current prediction of the ensemble of trees, which is an efficient approximation to the functional gradient descent in the space of trees. The softmax objective is selected, with 50 Optuna trials were used for hyperparameter selection, and the multi class softmax objective is used. The search domain spans the number of estimators, the maximum depth of the tree, the learning rate, the subsample ratio, the column subsample ratio, the minimum value of the estimators — the minimum child weight, γ — and the regularization parameters — α and λ .

3.4.5. Focal Loss Objective

The objective function used to train the classification heads is Focal Loss which down weights the easy examples and gives more weight to the hard ones [18]. Let p_t be the probability of a ground truth class that is predicted. It is written in line (16) and is a smoothed version of cross entropy in line (17) with ε as smoothing factor and C as the number of classes.

$$\mathscr{L}_{FL} = -\ \alpha\ (1 - p_t)^{\gamma}\ \log(p_t) \quad (16)$$

$$Y^{ls}_c = (1 - \varepsilon)\ y_c + \varepsilon\ /\ C \quad (17)$$

The focusing parameter γ was chosen to be 2.0 and the smoothing factor ε was chosen to be 0.05. The per class weight α was calculated from the training partition based on the inverse class frequency method that accounts for the remaining imbalance after the stratified split. For the GNN, the BiLSTM and the hybrid head, the same loss formulation was used but for the XGBoost head we used the internal objective, the multi class log loss [19].

3.5. Proposed Hybrid Frameworks

3.5.1. UAVIDS-2025 Pipeline. GNN-BiLSTM + XGBoost

The three layers inside the GATv2, the two-layer BiLSTM with temporal attention, and the XGBoost with Optuna are combined in the pipeline used for UAVIDS-2025. GATv2 encoder: embeddings generated by the GATv2 encoder, along with the projected edge feature, are fused together to create an edge level embedding [h_{src} ‖ h_{dst} ‖ e]. After further reshaping the resulting edge-level embedding to sequences of 10 is passed through the BiLSTM, resulting in a 256-dimensional temporal representation after attention pooling. This is added to the 27-dimensional raw edge feature vector, creating a hybrid feature vector of size 275, which is then fed to the Optuna-tuned XGBoost classifier for predicting five classes. This combination of temporal embedding and the raw flow features maintains low-level traffic statistics that complement those provided by the relational and temporal patterns in the neural component. The number of trainable parameters of the neural component is about 1.27 million. The tuned XGBoost head has 479 trees, 6 maximum depth, and 5.22×10^{-2} learning rate. The full architecture is given as shows in Fig. 5.

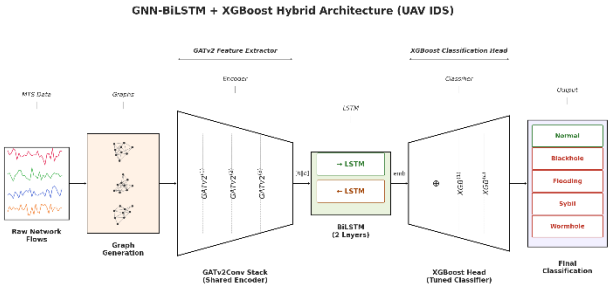

Fig. 5. Architecture of the proposed GNN-BiLSTM + XGBoost hybrid model for the UAVIDS-2025 dataset.

3.5.2. T-ITS Pipeline. AE + GNN + BiLSTM

The T-ITS pipeline prepends a denoising autoencoder to the graph encoder and replaces the XGBoost head with an end-to-end neural head. Every input is compressed to an 8-dimensional vector in the latent space by the autoencoder. Finally, the latent graph is created in the shape of a KNN graph which then is processed by the GATv2 stack – similar to the UAVIDS pipeline. Edge level representation [h_{src} ‖ h_{dst} ‖ e] is reshaped to sequences with a length of 10 and fed into a temporal attention augmented, 2-layer, bidirectional LSTM. Final class scores are modeled by a three-layer multilayer perceptron with widths 256, 128 and 5. This pipeline has a total of 1,188,166 trainable parameters. A complementary direct head without a recurrent block was evaluated as an ablation control. The complete architecture is reported in Fig. 6.

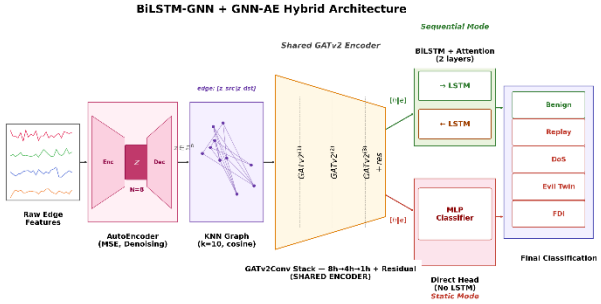

Fig. 6. Architecture of the proposed AE + GNN + BiLSTM hybrid model for the T-ITS dataset.

3.6. Evaluation Metrics

The metrics used to measure the model performance include accuracy, precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC-ROC). The confusion matrix consists of four entries: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), and the four-classification metrics are all calculated from these four entries. The accuracy, precision, recall and F1 score are defined by equations (18 – 21) [20].

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (18)$$

$$\text{Precision} = TP / (TP + FP) \quad (19)$$

$$\text{Recall} = TP / (TP + FN) \quad (20)$$

$$F1 = 2 \cdot (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (21)$$

In the case of multi class problems, macro and weighted averages of these measures are provided. The macro average and the weighted average are the average F1 scores of each class and weighted by the support of each class, respectively, as written in (22) and (23). Macro averaging assigns more penalties to misclassifications in minority classes than does weighted averaging; this is why it is used for the class imbalanced benchmarks [20] and [21].

$$F1_{macro} = (1 / C) \Sigma_{c=1}^{C} F1_c \quad (22)$$

$$F1_{weighted} = \Sigma_{c=1}^{C} (n_c / N) F1_c \quad (23)$$

The AUC-ROC summarizes the threshold independent ranking quality of the predicted scores. It is computed as the area under the curve that maps the False Positive Rate to the True Positive Rate as the decision threshold is varied. The integral form is given by equation (24). For the multi class setting, the macro averaging one versus rest variant is used, which is thus a single scalar summary of the discrimination quality for all class pairs [22].

$$\text{AUC-ROC} = \int_0^1 \text{TPR}(FPR^{-1}(t))\, d(t) \quad (24)$$

4. Experimental Results and Discussion

4.1. Overview of the Experimental Setup

This section presents the experimental assessment conducted on the two datasets mentioned in Section 3, in the context of the proposed IDS framework. The aim of the evaluation is two-fold. The first is to quantify the contribution of each of the architectural components to the classification performance. The second one is to evaluate the generalization ability of the common GATv2 body on two different datasets: varying topology, feature space, and class imbalance profile. This model was trained in three different versions: a pure GNN baseline, a pure recurrent baseline, and the proposed hybrid model. For all models, the stratified hold-out test partition of 20 per cent of each dataset was used.

Of those reported, accuracy, precision, recall, F1 score and AUC-ROC are used. Macro averages were used for UAVIDS-2025, while weighted averages were used for T-ITS because of class imbalance. Averaged were also done on the proposed model accounting for completeness. For cross configuration comparison, the optimiser, learning rate schedule and Focal Loss are set to the same for all experiments in the respective datasets [2] [8].

4.2. Results on the UAVIDS-2025 Dataset

4.2.1. Overall Performance

The results of the three configurations on UAVIDS-2025 are shown in Table 3 considering the test set performance. The results of the GNN-Only model are as follows: macro accuracy (94.28%), precision (94.73%), recall (94.59%), F1 score (94.60%), and AUC-ROC (0.9945). The BiLSTM baseline performed substantially worse, with 78.12% accuracy, 77.87% F1-score, and 0.9248 AUC-ROC. The more than 16% points difference in the accuracy values underline the importance of graph topology in classification of multi UAV traffic. On the test set, the hybrid model achieved 98.05% accuracy, 98.22% precision, 98.15% recall, 98.18% F1-score, and 0.9993 AUC-ROC. The results are 3.77% points higher in accuracy and 3.58% points higher in F1 score compared to GNN-Only.

Table 3. Overall test set performance of the three model configurations on the UAVIDS-2025 dataset.

Model	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1-Score (%)*	*AUC-ROC*
BiLSTM Baseline	78.12	79.09	77.77	77.87	0.9248
GNN-Only (UAVGNNOnly)	94.28	94.73	94.59	94.60	0.9945
GNN-BiLSTM + XGBoost (Proposed)	**98.05**	**98.22**	**98.15**	**98.18**	**0.9993**

4.2.2. Per-Class Analysis

Table 3 presented the aggregated per class metrics, however, these do not show behavior per class for each model. Fig. 7 plots the per class precision, recall and f1 score at the GNN-Only baseline. There are three classes which have F1 scores over 98%; Flooding Attack, Normal Traffic and Sybil Attack. The two routing-manipulation classes perform less strongly with a F1 value of 88.05% and 87.83% respectively for Blackhole Attack and Wormhole Attack. The same families of signatures in the network level also exist for both classes, resulting in lower inter class separability, namely drop rate spikes, abnormal reduction of the end to end hop count and asymmetric delay.

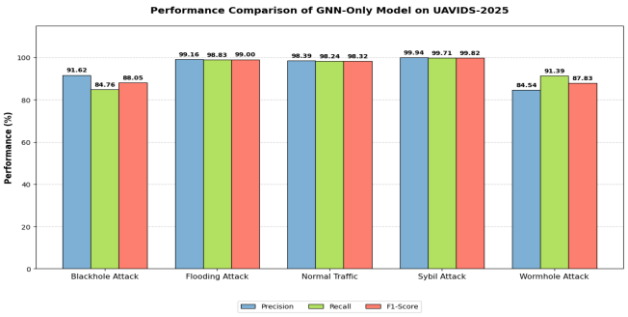

Fig. 7. Per class Precision, Recall, and F1 score obtained by the GNN-Only baseline on the UAVIDS-2025 test set.

The confusion matrix of the GNN-Only model on the UAVIDS-2025 test set is given in Fig. 8. The most important off diagonal mass in this case is the one for the pair Blackhole - Wormhole, with 768 Blackhole samples labeled as Wormhole and 366 Wormhole samples marked as Blackhole. Essentially, we see that the Sybil class is perfectly separated, which is also in line with the structurally different forging of identity attack signature of artificial UAVs into various virtual identities.

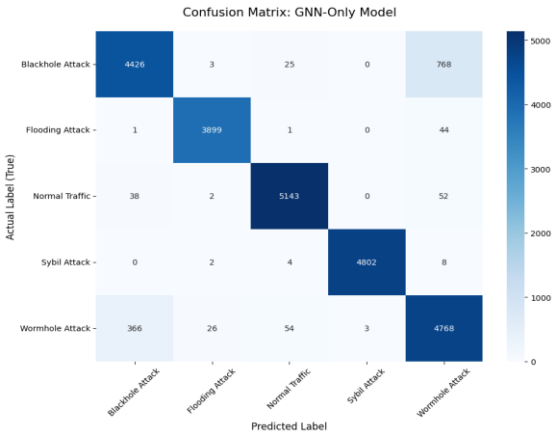

Fig. 8. Confusion matrix of the GNN-Only baseline on the UAVIDS-2025 test set.

The per class performance for the BiLSTM (baseline) model is reported in Fig. 9. This one looks quite different to the GNN-Only model. Normal Traffic remains at a still moderate F1 score of 90.29%, while all four classes of attack are in the 70% – 78% range, with Flooding Attack at the bottom with F1 score of 70.79%. The reason for the reduction is seen in the following confusion matrix of the BiLSTM baseline when presented in Fig. 10. Flooding traffic is often misclassified as Sybil traffic (1059 samples) or sybil traffic is mistaken as Flooding (1141 samples). In the raw feature space, an increased rate of packets per edge is the signature of both volumetric bursts and identity multiplication, and hence, without the relational context the BiLSTM is unable to distinguish between the two.

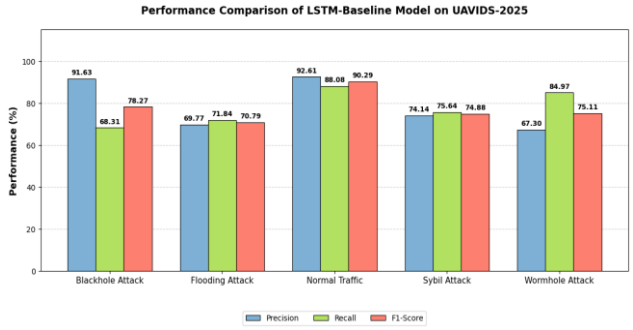

Fig. 9. Per class Precision, Recall, and F1 score obtained by the BiLSTM baseline on the UAVIDS-2025 test set.

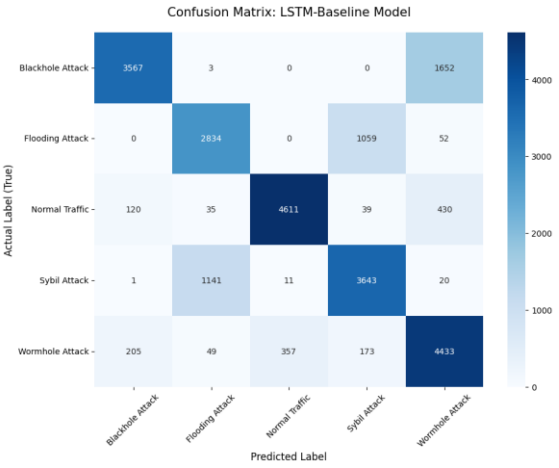

Fig. 10. Confusion matrix of the BiLSTM baseline on the UAVIDS-2025 test set.

The per-class results of the proposed hybrid model are reported in Fig. 11. The hybrid achieves almost perfect detection for Flooding (F1 = 99.87%), Sybil (F1 = 99.79%), and Normal Traffic (F1 = 99.42%), while also enhancing the detection of the two routing manipulation classes: by 7.13% for "Blackhole" and by 7.09% for "Wormhole". It can be seen in the confusion matrix in Fig. 12 that there are just 24 Wormhole samples that are misclassified as Blackhole and 16 Blackhole samples that are misclassified as Wormhole with essentially near zero entries elsewhere on the off diagonal. As a result, the previously described temporal sequencing of edge level graph features and subsequent XGBoost decision surface results in exactly the complementary inductive bias to overcome the remaining semantic ambiguity between the seemingly overlapping routing attack classes.

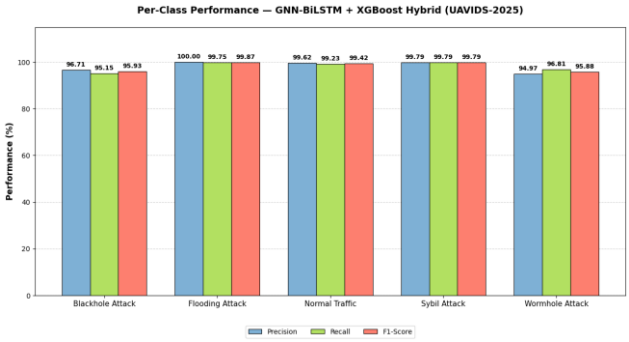

Fig. 11. Per class Precision, Recall, and F1 score obtained by the proposed GNN-BiLSTM + XGBoost hybrid model on the UAVIDS-2025 test set.

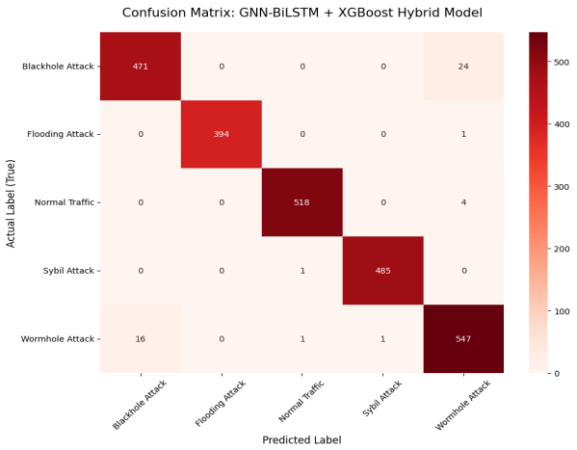

Fig. 12. Confusion matrix of the proposed GNN-BiLSTM + XGBoost hybrid model on the UAVIDS-2025 test set.

4.3. Results on the T-ITS Cyber Dataset

4.3.1. Training Dynamics

Figs. 13, 14 and 15 summarize training behavior of all three architectures on T-ITS. The training loss curves are shown in Fig. 13. The proposed hybrid model reduces from 0.84 at an initial epoch, to approximately 0.30 after only a few epochs and eventually settles at 0.28. The GNN-AE Only and BiLSTM + AE baselines have loss values around 1.03 and 1.08, respectively, that means that these two methods in isolation cannot be used to fit the training distribution.

Fig. 13. Training loss curves of the three model configurations on the T-ITS dataset.

The results of the validation loss curves can be seen in Fig. 14. Compared to the proposed hybrid that stabilized at ~0.4, the GNN-AE Only baseline has a fluctuation range of [1.2, 1.4] and the BiLSTM + AE baseline drifts down towards 1.30 from ~1.65 with high variance. The fact that the proposed hybrid clearly outperforms both baseline methods in every case (by more than 3 fold) indicates that none of these methods (in isolation), can overcome the combined problems of high feature dimensionality, the increase in graph density caused by KNN construction and class imbalance.

Fig. 14. Validation loss curves of the three model configurations on the T-ITS dataset.

The curves for the validation F1 score are shown in Fig. 15. The proposed hybrid has a steady upward trend to stabilize above 0.58 whereas the GNN-AE Only baseline shows a fluctuation around 0.4, and the BiLSTM + AE baseline shows a fluctuation around 0.3 with high variability. This latter trajectory indicates that the BiLSTM + AE baseline didn't converge to any steady-state optimum, and probably would be affected by minor changes in the input distribution at test time.

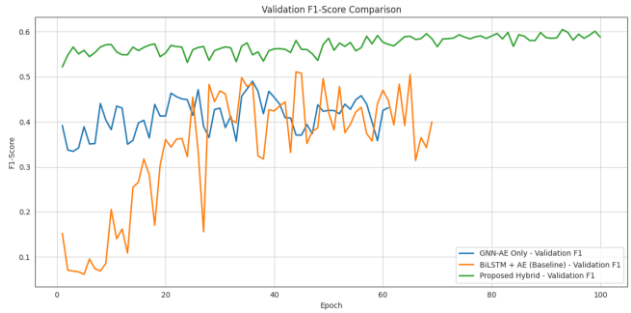

Fig. 15. Validation F1 score curves of the three model configurations on the T-ITS dataset.

4.3.2. Overall Performance

The test set performance for the three configurations on T-ITS for the five-class multi-class task and T-ITS for binary attack and benign class tasks are shown in Table 4. The corresponding multi class accuracy of the GNN-AE Only baseline is 54.31% and the weighted F1 score is 59.84%. Its binary performance results in 66.12% accuracy and 79.61% F1 score (recall-based). The baseline BiLSTM + AE model achieves a superficially high binary F1 score of 99.73%, but only 32.01% multi-class F1-scour demonstrating class collapse: this is a phenomenon where the model provides good coverage for the two most frequent classes (Replay, DoS) and at the same time shows very poor coverage for the minority categories (Evil Twin and FDI). This failure mode is especially problematic in deployed cases of IDS, in which rarer classes of attacks—representing the most sophisticated and most targeted attacks—may be the most important. High binary F1 alone does not imply that the model is useful for multi-class operational detection

The proposed hybrid model achieves multi class accuracy of 99.16%, weighted f1 score of 99.57% and binary f1 of 99.69%. The Multi class F1 Gain over GNN-AE Only is 39.73% points. This improvement on the same task with respect to the BiLSTM + AE baseline is 67.56% points. These results quantify the synergistic effect of graph topology, which captures inter-sample structural relationships in the autoencoder latent space, and temporal modeling, which captures intra-sequence dynamics within the 10-step sliding window.

Table 4. Performance comparison of the three model configurations on the T-ITS cyber dataset, for both the multi class and binary tasks.

Model	*Task*	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1-Score (%)*
GNN-AE Only	Multi class	54.31	77.26	54.31	59.84
GNN-AE Only	Binary	66.12	100.00	66.12	79.61
BiLSTM + AE (Baseline)	Multi class	37.10	28.14	37.10	32.01
BiLSTM + AE (Baseline)	Binary	99.47	100.00	99.47	99.73
AE + GNN + BiLSTM (Proposed)	**Multi class**	**99.16**	**99.99**	**99.16**	**99.57**
AE + GNN + BiLSTM (Proposed)	**Binary**	**99.37**	**100.00**	**99.37**	**99.69**

4.3.3. Stability Across Attack Classes

The distribution of per class F1 scores achieved by three architectures is reported in Fig. 16. The BiLSTM + AE baseline shows an extremely large range of per class F1 value, where several classes are clustered close to 0.0. Both baseline models (GNN-AE Only, BiLSTM + AE) have a very large spread on the per class F1 values, with several classes grouped near 0.0 The pattern is to find one or two predominant labels and that poses a class collapse failure mode because the model was predicting nearly all or just slightly two of those labels. The proposed hybrid produces a tighter grouping of per-class F1 values very tightly around the necessary asymptotes with only a few that deviate from the latter in the minority. One of the key characteristics of any deployed IDS is stability across classes, in that an operator deals with an unknown, time varying attack.

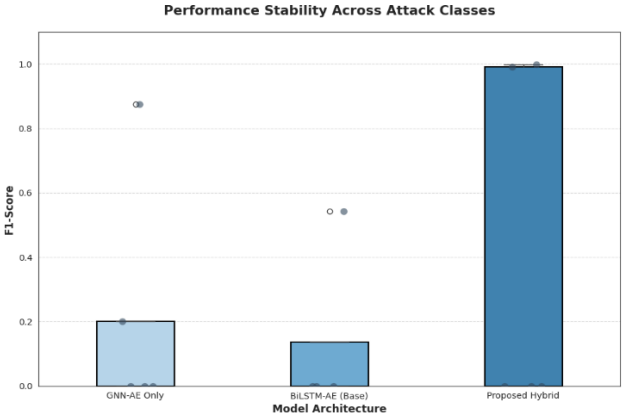

Fig. 16. Distribution of per class F1 scores produced by the three model configurations on the T-ITS test set.

4.4. Comparison with Related Works

Tables 5 and 6 compare the proposed method to the directly relevant published baselines on the two examined datasets.

4.4.1. Comparison on UAVIDS-2025

In the original release of UAVIDS-2025, a common 80 / 20 stratified split was used to validate 10 supervised classifiers [8]. The top classical model was XGBoost with the raw flow features and achieved an accuracy of 96% and macro F1 score of 96%. The highest deep learning performance was given by LSTM model with the same features (94% on both metrics). The final accuracy and F1 score of the proposed hybrid are 2.05% points and 2.18% points higher than that of XGBoost respectively. The accuracy of the Hybrid is also 4.05% points higher than that of the LSTM. The improvement is prominent for the class Wormhole in which published LSTM has an F1 score of about 91% and the proposed hybrid shows an F1 score of 95.88%. The 4.88%-point increase on Wormhole is a result of the inductive bias of graph message passing, because graph message passing helps capture routing-shortcut patterns of strange AverageHopCount reductions, or asymmetric latency patterns on Wormhole.

Table 5. Comparison of the proposed GNN-BiLSTM + XGBoost hybrid with published baselines on the UAVIDS-2025 dataset.

Method	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1-Score (%)*
Logistic Regression	82	83	82	82
Support Vector Machine	88	89	88	88
Decision Tree	93	94	94	94
Random Forest	95	96	95	95
k Nearest Neighbours	91	91	91	91
XGBoost	96	96	96	96
LightGBM	95	95	95	95
DNN	92	93	93	93
CNN	93	94	93	93
LSTM	94	94	94	94
GNN-BiLSTM + XGBoost (Proposed)	**98.05**	**98.22**	**98.15**	**98.18**

4.4.2. Comparison on the T-ITS Cyber Dataset

The most relevant work carried out before on the T-ITS data set includes a classical autoencoder to compress the original feature space, which is then fed to a set of traditional classifiers, such as Decision Tree, Random Forest, k Nearest Neighbors, Multilayer Perceptron and Support Vector Machine, including feature selection variants based on SHAP. The best published binary classifier (Decision Tree with an 8 feature autoencoder) has an F1 of 94.21% and an accuracy rate of 94.54%. The highest published multi class classifier is MLP with an 8 feature autoencoder with an F1 score of 84.09% and accuracy of 75.34%. The proposed hybrid is significantly better than these baselines. The binary F1 score increases to 99.69% (+ 5.48% points) and the multi class F1 score goes up to 99.57% (+ 15.48% points). The improvement seems to be especially noticeable in the minority FDI class (8.2% of samples) and in the Evil Twin class (13.4%) – the most difficult cases in the published baseline. The KNN structure of the latent space naturally captures pairwise similarities and adds the local context to the model for those samples with the rare class.

Table 6. Comparison of the proposed AE + GNN + BiLSTM hybrid with published baselines on the T-ITS cyber dataset.

Method	*Task*	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1-Score (%)*
DT + AE (N = 8)	Binary	94.54	94.87	94.54	94.21
RF + AE (N = 8)	Binary	91.45	92.31	91.45	91.23
MLP + AE (N = 8)	Binary	91.19	93.22	91.19	92.54
SVM + SHAP (N = 8)	Binary	93.47	98.17	94.12	95.02
FNN + SHAP (N = 8)	Binary	94.96	94.11	95.23	96.14
MLP + AE (N = 8)	Multi class	75.34	85.99	82.26	84.09
FNN + SHAP (N = 8)	Multi class	72.37	81.78	80.02	80.89
AE + GNN + BiLSTM (Proposed)	**Binary**	**99.37**	**100.00**	**99.37**	**99.69**
AE + GNN + BiLSTM (Proposed)	**Multi class**	**99.16**	**99.99**	**99.16**	**99.57**

The published baselines on T-ITS are in the classical machine learning paradigm where the features are handcrafted and are independent of the downstream classifier. The introduced method aims at tackling it as a node anchored relational learning problem by introducing an explicit graph topology layer. The compression ratio of the pipeline design (N = 8) is equivalent to the bottleneck size of the proposed design, and the structural information lost by the compression is partially recovered by neighborhood aggregation conducted for GATv2 message passing in the subsequent KNN graph construction. The classical baselines do not make use of this inductive bias.

4.5. Discussion

4.5.1. Effect of Graph Topology

For both datasets, graph structure information is the most significant factor affecting the detection performance. In the GNN-Only setting of UAVIDS-2025, the accuracy has already been improved to 94.28%, without any temporal modelling being used. This lower performance does not necessarily indicate that graph learning is ineffective; rather, it underscores a critical architectural insight: that this architecture alone is insufficient for the T-ITS setting. This highlights the extreme complexity of building a meaningful graph on embeddings from an 8-dimensional autoencoder for a five-way classification problem in the presence of large class imbalance. Crucially, when the BiLSTM temporal component is integrated on top of the GNN encoder, the hybrid model achieves a breakthrough accuracy of 99.16%. This successfully demonstrates that the static graph snapshots leave some ambiguity that sequential context helps resolve. This pivotal

finding significantly advances previous graph-based intrusion detection research, solidifying the fact that relational aggregation acts as an exceptionally strong inductive bias in the classification of network traffic.

4.5.2. Complementary Role of XGBoost in the UAVIDS-2025 Pipeline

The GNN-BiLSTM model on UAVIDS-2025 already reaches a test F1 score of 96.23%. A concatenation of its 256-dimensional temporal embedding with the original 19 raw flow features produces a 275-dimensional hybrid representation, on top of which an Optuna-tuned XGBoost head further elevates the F1 score to a superior 98.18%. This additional gain of 1.95% points presents a highly favorable trade-off, as it comes at moderate computational overhead and empirically validates the complementary inductive biases of the two learners. Crucially, the architecture synergizes two distinct modeling paradigms: the deep neural component safeguards the extraction of intricate, nonlinear relational patterns, while the decision tree ensemble expertly partitions the residual feature space and corrects the few remaining errors associated with the routing manipulation classes. Furthermore, the exceptional AUC-ROC of 0.9993 establishes that this two-stage pipeline guarantees near-perfect class separability across all decision thresholds. This outstanding metric represents an invaluable property in operational UAV settings, where the false positive rate must be tightly controlled to prevent mission disruption.

4.5.3. Class Imbalance and Minority Class Detection

The T-ITS dataset exhibits a severe class imbalance, with only 8.2% of records belonging to the FDI class, 13.4% to Evil Twin, and 56% to the Replay and DoS classes. Remarkably, the proposed hybrid model demonstrates exceptional granular performance, yielding an individual per-class F1-score exceeding 98% across all five distinct categories, including the highly underrepresented FDI minority class. This outstanding resilience is directly attributed to the Focal Loss objective, which strategically down-weights well-classified, easily predictable majority class samples and concentrates gradient updates on the more difficult examples within the minority class.

The BiLSTM+AE baseline model, on the other hand, suffered a catastrophic class collapse, predicting with near-zero F1-scores for the two least abundant classes (FDI and Evil Twin), despite misleadingly achieving a high binary F1-score of 99.73% by solely capturing the two major attack classes. This critical failure mode exposes a dangerous vulnerability in an in-service IDS deployment, as the rarest attack vectors are frequently the most sophisticated, targeted, and destructive threats to UAV missions. These results unambiguously validate the synergistic effect of graph topology encoding, temporal modeling, and Focal Loss in erecting a robust safeguard against the neglect of minority classes in highly imbalanced UAV cybersecurity scenarios.

4.5.4. Generalisation Across Dataset Types

The GATv2 backbone demonstrates exceptional adaptability to both dataset types, requiring minimal architectural modifications. Remarkably, the core structural integrity of the GATv2 encoder remains rigorously consistent across all experiments in this study, maintaining a uniform configuration of 3 layers, 8/4/1 attention heads, and a hidden width of 64. The only varying graph construction strategy is that a direct flow graph is utilized for UAVIDS-2025, whereas a KNN-AE graph is deployed for T-ITS. Similarly, the classification heads are seamlessly interchanged, employing an Optuna-tuned XGBoost model in the former setup and an end-to-end BiLSTM with attention in the latter. Given the highly modular blueprint of the proposed framework—where the GATv2 encoder design is dataset-agnostic and the graph construction strategies are easily customized—this architecture establishes a highly scalable paradigm capable of seamlessly generalizing to broader, unseen UAV communication ecosystems. While extending this framework to fundamentally different network topologies, novel attack vectors, or diverse feature distributions remains a promising avenue for future empirical testing, the current cross-dataset success underscores an invaluable deployment advantage: a single, robustly trained backbone can be effortlessly projected across highly heterogeneous UAV environments, eliminating the need for computationally prohibitive retraining from scratch

5. Conclusion

The proposed models were evaluated on held-out test partitions, achieving 98.18% F1-score on UAVIDS-2025 and 99.57% weighted F1-score on T-ITS. This approach successfully resolves a fundamental bottleneck of many existing UAV IDS approaches: disregarding the relational and temporal aspects of UAV communication and only treating the traffic records as tabular observations. The framework represented UAV interactions as graph structures and trained graph node and edge embeddings using a GATv2-based graph encoder to learn the contextual dependency of UAVs, ground stations, and network flows. The following BiLSTM module then captured the dynamics of attack behavior within short time horizons, and classification heads specific to the traffic format allowed the model to accommodate various traffic formats. The experimental tests on UAVIDS-2025 and T-ITS cyber data set showed that the proposed hybrid models consistently achieved the best results compared to the other graph-only and sequence-only baselines, validating that spatial topology and temporal dynamics are critical, complementary dimensions for UAV intrusion detection.

Results confirm the outstanding cross-domain efficacy of the framework, proving it effective in simulated multi-UAV ad hoc traffic and in real cyber-physical communication data collected from Wi-Fi-based communication. The GNN-BiLSTM + XGBoost model demonstrated superior discriminative power on UAVIDS-2025 to effectively isolate sophisticated routing-manipulation attacks (Blackhole and Wormhole), which are challenging to differentiate using traditional feature-based models. The AE + GNN + BiLSTM pipeline provided a robust, scalable solution to the critical class imbalance problem in the T-ITS dataset, significantly enhancing the detection rates of minority attack classes by combining latent space graph construction with temporal learning. These results establish that graph-based spatiotemporal modeling is a highly promising and indispensable direction for scalable and adaptive UAV security systems. Future work should consider online and resource-constrained deployment scenarios, and the robustness of the framework to unseen and adversarial attacks should be tested and explored, as well as lightweight variants for real-time execution on UAV edge devices.

Acknowledgment

The authors would like to acknowledge the College of Computers and Information Technology, Taif University, Saudi Arabia, for funding this research and for providing academic and institutional support through the Department of Computer Engineering. The authors also appreciate the supportive research environment provided during the preparation of this work.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] R. Shrestha, A. Omidkar, S. A. Roudi, R. Abbas, and S. Kim, "Machine-learning-enabled intrusion detection system for cellular connected UAV networks," Electronics, vol. 10, no. 13, Art. no. 1549, 2021, doi: 10.3390/electronics10131549.

[2] S. C. Hassler, U. A. Mughal, and M. Ismail, "Cyber-physical intrusion detection system for unmanned aerial vehicles," IEEE Trans. Intell. Transp. Syst., vol. 25, no. 6, pp. 6106–6117, Jun. 2024, doi: 10.1109/TITS.2023.3339728.

[3] S. N. Ashraf et al., "IoT empowered smart cybersecurity framework for intrusion detection in internet of drones," Sci. Rep., vol. 13, no. 1, Art. no. 18422, 2023.

[4] U. A. Mughal, R. Atat, and M. Ismail, "Graph neural network-based intrusion detection system for a swarm of UAVs," in Proc. MILCOM 2024 IEEE Mil. Commun. Conf. (MILCOM), Oct. 2024, pp. 578–583.

[5] D. H. Tran and M. Park, "FN-GNN: A novel graph embedding approach for enhancing graph neural networks in network intrusion detection systems," Appl. Sci., vol. 14, no. 16, Art. no. 6932, 2024.

[6] R. Wang et al., "Network traffic analysis based on graph neural networks: A scoping review," Big Data Cogn. Comput., vol. 9, no. 11, Art. no. 270, 2025.

[7] M. Homaei, I. Khazrak, R. Molano, A. Caro, and M. Avila, "Graph attention networks with physical constraints for anomaly detection," arXiv preprint arXiv:2601.12426, 2026.

[8] Q. Zeng, A. Bashir, and F. Nait-Abdesselam, "UAVIDS-2025: A benchmark dataset for intrusion detection in UAV networks using machine learning techniques," in Proc. 2025 IEEE Conf. Commun. Netw. Secur. (CNS), Sep. 2025, pp. 1–9.

[9] T. C. Vuong et al., "Effective intrusion detection for UAV communications using autoencoder-based feature extraction and machine learning approach," arXiv preprint arXiv:2410.02827, 2024.

[10] R. A. Ramadan, A. H. Emara, M. Al-Sarem, and M. Elhamahmy, "Internet of drones intrusion detection using deep learning," Electronics, vol. 10, no. 21, Art. no. 2633, 2021.

[11] L. Kou, S. Ding, T. Wu, W. Dong, and Y. Yin, "An intrusion detection model for drone communication network in SDN environment," Drones, vol. 6, no. 11, Art. no. 342, 2022.

[12] M. Aldossary, I. Alzamil, and J. Almutairi, "Enhanced intrusion detection in drone networks: A cross-layer convolutional attention approach for drone-to-drone and drone-to-base station communications," Drones, vol. 9, no. 1, Art. no. 46, 2025.

[13] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?," arXiv preprint arXiv:2105.14491, 2021.

[14] Y. Zhang, C. Yang, K. Huang, and Y. Li, "Intrusion detection of industrial internet-of-things based on reconstructed graph neural networks," IEEE Trans. Netw. Sci. Eng., vol. 10, no. 5, pp. 2894–2905, 2023.

[15] F. Kong, J. Li, B. Jiang, H. Wang, and H. Song, "Integrated generative model for industrial anomaly detection via bidirectional LSTM and attention mechanism," IEEE Trans. Ind. Informat., vol. 19, no. 1, pp. 541–550, Jan. 2023.

[16] F. S. Alrayes, M. Zakariah, S. U. Amin, Z. I. Khan, and M. Helal, "Intrusion detection in IoT systems using denoising autoencoder," IEEE Access, vol. 12, pp. 122401–122425, 2024.

[17] Y. Sun, "An XGBoost-Softmax framework for multi-class DDoS network attack detection," in Proc. 2025 Int. Conf. Intell. Commun. Netw. Comput. Techn. (ICICNCT), Sep. 2025, pp. 1–5.

[18] S. Xie, C. Zhan, J. Li, and Y. Li, "Intrusion detection method based on graph edge attention and focal loss," in Proc. 2025 4th Int. Conf. Cryptography, Netw. Secur. Commun. Technol., Jan. 2025, pp. 21–28.

[19] H. Elwahsh et al., "Hyperparameter optimization of XGBoost and hybrid CNN-SVM for cyber threat detection using modified Harris hawks algorithm," PeerJ Comput. Sci., vol. 11, Art. no. e3169, 2025.

[20] A. H. Hamad, N. K. Hussein, and A. M. Abdulghani, "A deep learning paradigm for intrusion detection in unmanned aerial vehicle networks using extended LSTM," Int. J. Intell. Eng. Syst., vol. 18, no. 4, 2025.

[21] K. Takahashi, K. Yamamoto, A. Kuchiba, and T. Koyama, "Confidence interval for micro-averaged F1 and macro-averaged F1 scores," Appl. Intell., vol. 52, no. 5, pp. 4961–4972, 2022.

[22] H. M. Attaullah, I. U. Khan, M. M. Alam, and K. Kaushik, "RoboLSTM-IDS: Multi-dataset evaluation of deep learning framework for UAV network," PeerJ Comput. Sci., vol. 12, Art. no. e3500, 2026.