# Edge-Oriented Tree-Guided Temporal Fusion for GNSS Spoofing Detection in UAV-Class Consumer Receivers

Abima Obim Abima, Ali Kashif Bashir, Md. Israfil Biswas, Muhammad Atif Ur Rehman, and Norah Saleh Alghamdi

***Abstract*—Low-cost Global Navigation Satellite System (GNSS) receivers are increasingly used in consumer unmanned aerial vehicles (UAVs), compact autonomous platforms, and location-aware consumer devices, where spoofed observables can affect localisation, return-to-home logic, route execution, and geofencing. Existing anti-spoofing methods often depend on auxiliary sensors, multi-antenna front-ends, specialised receiver access, or platform-specific fusion, which restricts their practicality on cost- and resource-constrained consumer platforms. To address this gap, this paper proposes TGTF-NET, an edge-oriented Tree-Guided Temporal Fusion Network for receiver-centric GNSS spoofing detection from synchronised eight-channel receiver observations under unmanned aerial system (UAS)-emulated conditions. The design integrates shared bidirectional recurrent temporal encoding, channel-validity-aware attention and max pooling, differentiable threshold-bank evidence that approximates tree-style splits, residual temporal-summary fusion, and auxiliary channel-level supervision within a receiver-side detector with validation-selected thresholding. The method is evaluated on the curated Mendeley v3 UAS spoofing dataset using mission-aware segmentation, classical and neural baselines, repeated-seed stability, ablation, robustness, calibration, and cross-segment validation. On the held-out test split, TGTF-NET achieves an F1-score of 0.9505, recall of 0.9808, receiver operating characteristic area under the curve (ROC AUC) of 0.9555 and precision–recall area under the curve (PR AUC) of 0.9883 at a validation-selected operating threshold. Across repeated seeds, it obtains a mean F1-score of 0.9377 $\pm$ 0.0074. Within the evaluated protocol, TGTF-NET gives the highest recall and F1-score among the thresholded models, while gradient boosting retains higher ROC AUC and PR AUC. The results support TGTF-NET as a receiver-side anti-spoofing component for UAV-class consumer GNSS devices and indicate its relevance to location-oriented security in future edge-enabled and Beyond-6G systems that depend on trustworthy positioning, with remaining work on calibration, false-alarm reduction, embedded-hardware validation, and cross-segment generalisation.**

***Index Terms*—Consumer UAVs, GNSS spoofing detection, Edge AI, Location-oriented security, Beyond-6G systems, Temporal fusion, Multi-channel GNSS receivers, Receiver security, threshold calibration, Trustworthy positioning, Consumer electronics.**

A. O. Abima, A. K. Bashir, and M. A. Ur Rehman are with the Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, United Kingdom (e-mail: a.abima@mmu.ac.uk; dr.alikashif.b@ieee.org; m.atif.ur.rehman@mmu.ac.uk).

M. I. Biswas is with the School of Computer Science and Technology, University of Bedfordshire, Bedfordshire, United Kingdom (e-mail: mdisrafil.biswas@beds.ac.uk).

N. S. Alghamdi is with the Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia (e-mail: nosalghamdi@pnu.edu.sa).

## I. Introduction

LOW-COST Global Navigation Satellite System (GNSS) receivers are now embedded in consumer unmanned aerial vehicles (UAVs), prosumer aerial platforms, compact autonomous systems, and location-aware consumer devices. These receivers support localisation, return-to-home behaviour, geofencing, route execution, image geotagging, and safety-related navigation functions. Their widespread use creates a consumer-electronics integrity problem: a device may make operational decisions from civilian GNSS signals that are openly specified and vulnerable to spoofing. In a spoofing attack, counterfeit GNSS-like signals or manipulated receiver observables can induce an incorrect navigation solution while preserving superficially plausible receiver behaviour [1]–[4]. For UAV-class consumer platforms, this can lead to route deviation, false geofence decisions, unsafe return-to-home behaviour, or mission loss.

Many GNSS anti-spoofing techniques rely on additional sensing, specialised receiver access, or platform-specific integration. Examples include multi-antenna spatial processing, inertial consistency checks, perception-assisted validation, correlation-shape monitoring, and tightly coupled sensor-fusion pipelines [1], [2], [5]–[7], [9]. These approaches can provide strong physical evidence, but they may increase hardware cost, size, weight, power consumption, calibration effort, and software-integration complexity. Such constraints are particularly relevant to consumer electronics, where security functions are often implemented using the device's sensing and computation capabilities. This motivates receiver-side learning methods that operate directly on GNSS observables, avoid auxiliary sensors, and remain compatible with low-latency edge inference.

Public receiver-side GNSS spoofing datasets remain limited. The Mendeley UAS spoofing dataset v3 provides synchronised observations from an eight-channel Global Positioning System (GPS) receiver under static and dynamic unmanned aerial system (UAS)-emulated conditions, with legitimate and spoofed labels [12]. Tree-based models have shown competitive performance on this dataset when preprocessing and feature handling are controlled [13], providing a demanding reference for neural methods. Any neural alternative therefore requires a clear architectural rationale, calibrated operating behaviour, robustness, interpretability, and a plausible receiver-side computational profile.

This paper proposes TGTF-NET, a tree-guided temporal-fusion architecture for receiver-centric GNSS spoofing detection in UAV-class consumer receivers. The design combines temporal sequence modelling of synchronised receiver channels, differentiable threshold-bank evidence inspired by tree-style splits, and validity-aware channel aggregation. Temporal modelling captures changes in carrier-to-noise density ratio, Doppler-related quantities, pseudorange, carrier phase, and correlator-derived measurements. The threshold branch approximates split-like evidence associated with competitive tree-based classifiers, while the receiver-centric formulation treats the eight channels as synchronised streams from one GNSS receiver rather than UAV nodes or a communication graph. The dataset therefore supports multi-channel receiver intelligence under UAS-emulated conditions, not physical multi-UAV graph reasoning.

The proposed detector is evaluated from a deployment-oriented perspective rather than only as a ranking model. It performs receiver-local edge inference for consumer-device security without raw-observation offloading or additional sensing infrastructure [24]. In this broader context, trustworthy receiver-side positioning is relevant to location-oriented security in future edge-enabled and Beyond-6G communication systems; however, the present study does not model or evaluate a 6G radio network. The experimental protocol includes mission-aware curation, segment-level train/validation/test splitting, validation-selected thresholding, comparison with classical and neural baselines, repeated-seed stability, ablation, channel-dropout and noise robustness, subgroup reporting, calibration analysis, cross-segment validation, and runtime profiling. This evaluation structure reflects the practical requirements of consumer receiver security: beyond ranking spoofed windows, the detector needs to expose false-alarm behaviour, calibration quality, robustness to degraded channel information, and computational feasibility.

The main contributions are as follows:

1) **Consumer-receiver spoofing formulation:** We formulate GNSS spoofing detection as an edge-oriented receiver-security problem for UAV-class consumer devices, using synchronised multi-channel GNSS observations without auxiliary sensors, radio-frequency array hardware, or external reference infrastructure.
2) **Tree-guided temporal-fusion architecture:** We design TGTF-NET to combine shared bidirectional recurrent temporal encoding, channel-validity-aware attention and max pooling, differentiable threshold-bank evidence approximating tree-style splits, residual temporal-summary fusion, and auxiliary channel-level supervision.
3) **Deployment-aware evaluation protocol:** We evaluate the detector using validation-selected thresholding, classical and neural baseline comparison, repeated-seed stability, ablation, robustness to channel dropout and input noise, calibration assessment, subgroup analysis, cross-segment validation, and runtime profiling.
4) **Bounded empirical positioning:** We show that TGTF-NET provides a higher-recall thresholded operating point than the evaluated baselines on the curated Mendeley v3 split, while gradient boosting yields higher ROC AUC and PR AUC.

The remainder of this paper reviews related work, defines the threat model and dataset curation process, presents the TGTF-NET architecture and experimental protocol, reports the results, discusses deployment implications, and concludes the paper.

## II. Related Work

GNSS spoofing detection for UAV-class consumer receivers draws on several connected strands: receiver-level anti-spoofing, UAV spoofing demonstrations, spatial signal processing, perception-assisted detection, receiver-observable learning, and multi-channel GNSS benchmark analysis. These strands are relevant because consumer UAV-class receivers require detection methods that remain practical under limited sensing, limited receiver access, and low-latency onboard inference. General GNSS spoofing reviews describe signal-level, receiver-level, and navigation-domain evidence, including signal-quality monitoring, correlation checks, and receiver-side consistency tests [1]. A complementary anti-spoofing review examines GPS vulnerability and receiver-side defences from the perspective of receiver-based consistency testing and mitigation options [2]. These works provide useful threat and mitigation context, but they are broad reviews rather than edge-deployable temporal-learning detectors. In addition, several reviewed defences assume specialised receiver access, signal-level observables, or platform capabilities that may not be available in consumer UAV-class devices. TGTF-NET therefore targets a narrower receiver-side setting in which detection is performed from available multi-channel GNSS-derived features.

Practical spoofing studies establish why this receiver-side setting is operationally relevant. A civilian spoofing demonstration shows the feasibility of GPS spoofing using a portable GPS civilian spoofer [3]. Its value for the present work is the threat evidence, although it does not provide a receiver-side detection architecture. A UAV capture study further shows that spoofed GPS inputs can redirect unmanned-aircraft navigation behaviour [4]. However, it does not provide a calibrated multi-channel temporal detector for consumer receiver streams. A separate attack-requirements analysis examines the conditions for successful GPS spoofing, including attacker constraints and synchronisation requirements [5]. The study clarifies spoofing feasibility but it remains a threat-analysis study rather than an empirical learning-based detector. TGTF-NET complements these threat-oriented studies by detecting spoofing-labelled receiver windows from synchronised GNSS observations.

A further line of work uses spatial evidence and receiver-observable learning. Signal spatial-correlation detection can separate authentic and spoofed GNSS signals using spatial diversity [6]. The method provides direct physical evidence but it requires spatial-correlation information and receiver or antenna assumptions that may be restrictive for low-cost consumer UAV receivers. Spatial-processing detection similarly uses signal-geometry evidence to identify spoofing [7]. Although effective in settings where spatial evidence is available, it normally requires antenna geometry, receiver capability, or

signal access beyond many consumer devices. This motivates receiver-observable learning methods that avoid a spatial-processing front-end. Classical machine-learning models applied to GPS receiver features show that spoofing evidence can be extracted from GPS-derived measurements [8]. However, such pipelines mainly treat receiver features as machine-learning inputs rather than modelling synchronised multi-channel temporal windows with calibrated operating thresholds. TGTF-NET extends this direction by learning recurrent temporal representations and applying validation-selected thresholding to receiver-channel sequences.

Learning-based UAV spoofing detection has also been explored through perception-assisted, adversarial-learning, and image-representation approaches. Perception-assisted detection compares GNSS-derived behaviour with perception data to support machine-learning-based UAV GPS spoofing detection [9]. The additional sensing improves the evidence base, but it depends on additional sensing modalities and platform integration, which reduces portability to consumer receivers that may not expose camera, inertial, barometric, or perception features in a detection pipeline. Adversarial-learning-based detection studies GPS spoofing in UAVs from an intrusion-detection perspective [10]. The study provides a learning-based security view but it does not centre on synchronised multi-channel receiver streams, temporal-fusion design, calibrated threshold selection, or cross-segment generalisation. Image-based intrusion detection transforms spoofing-related UAV data into image-like representations for convolutional learning [11]. The image representation offers an alternative learning route, but it may obscure the native temporal and channel structure of GNSS receiver observations. TGTF-NET instead preserves the ordered receiver-window representation and learns temporal evidence directly from multi-channel streams.

The Mendeley UAS spoofing dataset is the closest public benchmark for the present study because it provides synchronised observations from an eight-channel GPS receiver under static and dynamic UAS-emulated conditions [12]. The dataset enables reproducible receiver-centric spoofing studies, but it is a benchmark resource rather than a detection architecture. Its UAS-emulated setting also introduces risks related to segment boundaries, class imbalance, timestamp artefacts, and mission-level distribution shift. TGTF-NET uses this dataset with mission-aware curation, segment-level splitting, calibration analysis, robustness testing, and cross-segment validation. The closest Mendeley v3 benchmark study explores multi-channel GPS receiver features for UAV spoofing detection using machine learning [13]. It is the closest empirical reference because it shows that carefully processed multi-channel receiver features, particularly with tree-based models, can achieve strong spoofing-detection performance. However, the approach remains primarily a conventional machine-learning treatment of engineered or tabular receiver features and does not provide a neural temporal-fusion architecture with differentiable tree-style split evidence, auxiliary channel-level supervision, validation-selected thresholding, and deployment-oriented calibration analysis. TGTF-NET complements this non-neural reference by integrating recurrent temporal encoding, differentiable threshold-bank evidence, and validity-aware channel fusion.

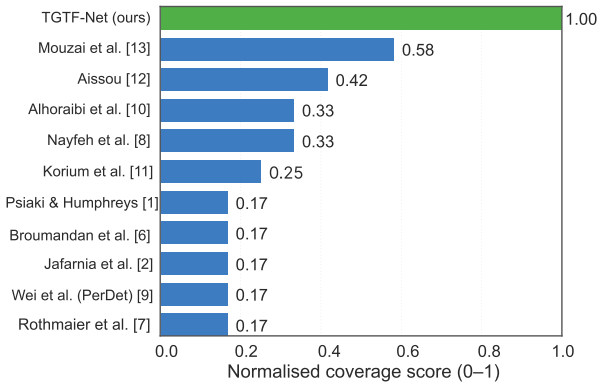



Fig. 1. Normalised related-work coverage.

Existing studies establish the spoofing threat and provide spatial, perception-assisted, image-based, and receiver-feature detectors. However, no reviewed study combines receiver-window temporal encoding, validity-aware aggregation, and differentiable threshold evidence while also reporting calibration, robustness, cross-segment generalisation, and runtime on the Mendeley v3 benchmark. This gap motivates the TGTF-NET architecture.

Figure 1 converts the qualitative related-work analysis into a normalised rubric-based coverage score. The score is computed across six detector-relevant dimensions: receiver-only or low-overhead operation, temporal fusion, threshold/tree-guided evidence, relevance to the Mendeley v3 UAS spoofing benchmark, deployment-oriented evaluation, and robustness or distribution-shift analysis. Each dimension is assigned an ordinal value in the interval $[0, 1]$, where explicit coverage is scored as 1.00, partial or indirect coverage as 0.50, and absent or non-applicable coverage as 0.00. For the overhead-related dimension, the score is inverted so that lower hardware or sensing dependence receives a higher value. For a reviewed study $i$ with $K = 6$ dimensions, the final score is

$$S_i = \frac{1}{K} \sum_{k=1}^{K} \tilde{x}_{i,k}, \qquad 0 \le S_i \le 1, \tag{1}$$

where $\tilde{x}_{i,k} = x_{i,k}$ for benefit-oriented dimensions and $\tilde{x}_{i,k} = 1 - x_{i,k}$ for the overhead-related dimension. The score is used only to visualise coverage of design dimensions and is not interpreted as empirical detection performance. Zero-score threat-only and feasibility studies are discussed in the text but excluded from the plotted ranking because the rubric measures detector-capability alignment rather than threat significance.

## III. THREAT MODEL AND CONSUMER-RECEIVER SETTING

The assumed defender is a UAV-class consumer or prosumer device equipped with a low-cost multi-channel GNSS receiver. The detector is deployed as a receiver-side software component and observes only GNSS-derived channel features and timestamps available from the receiver data stream. It does

not assume an inertial measurement unit, camera, barometer, magnetometer, multi-antenna array, external infrastructure, trusted reference station, or cryptographically authenticated GNSS/correction service. This constrained setting reflects a consumer-electronics use case in which spoofing detection is implemented with limited sensing, limited integration complexity, and low-latency onboard inference.

At inference time, the detector operates on a fixed-length multi-channel receiver window. In this paper, each window contains eight receiver channels, ten per-channel input features after preprocessing, and 60 temporal samples. The task is binary window-level spoofing detection, with auxiliary channel-level outputs used only for training support and interpretability. The channel-level outputs are not treated as independent physical UAV decisions, and the receiver channels are not interpreted as communication-graph nodes.

The adversary is external to the device and attempts to induce GNSS observables that cause the receiver to infer an incorrect navigation state. The attack is modelled at the receiver-observable level rather than at the RF propagation or spoofer-localisation level. Accordingly, the detector does not estimate the physical spoofer position, reconstruct RF propagation, or authenticate satellite signals. It distinguishes legitimate from spoofed receiver windows using the temporal and cross-channel structure present in the available GNSS observations.

The empirical instantiation follows the Mendeley v3 UAS spoofing dataset. The spoofing cases are therefore represented by the dataset labels: legitimate, simplistic spoofing, intermediate spoofing, and sophisticated spoofing [12], [13]. These labels define the supervised detection setting used in the experiments and are not presented as an exhaustive taxonomy of all physically possible GNSS spoofing strategies.

The evaluation uses UAS-emulated data rather than real consumer-drone field trials. The dataset was collected from an eight-channel receiver under vehicle-based dynamic emulation and static elevated locations intended to emulate UAS-like operation [12]. Consequently, the claims are restricted to receiver-centric spoofing detection under UAS-emulated conditions. The paper does not claim validation on deployed consumer UAV hardware, embedded power consumption, real flight-control integration, or real-time response under operational flight dynamics.

## IV. Dataset and Curation

### A. Dataset

The experiments use the Mendeley UAS spoofing dataset v3 [12]. After curation, the dataset contains 3839 temporal windows. Each window contains eight receiver channels, 10 input features, and 60 time steps. The 10 input features comprise nine GNSS observables and one channel-validity indicator. The retained dataset field names are CN0, Carrier_Doppler_hz, Pseudorange_m, Carrier_phase_cycles, EC, LC, PIP, PQP, and TCD. The preserved class labels are legitimate, simplistic spoofing, intermediate spoofing, and sophisticated spoofing.

TABLE I
Curated Dataset and Split Summary

| Item | Value |
|---|---|
| Receiver channels | 8 |
| Window length | 60 samples |
| Input features | 9 GNSS observables + 1 validity indicator |
| Total windows | 3839 |
| Training windows | 2858 |
| Validation windows | 588 |
| Test windows | 393 |
| Test legitimate/spoofed | 80 / 313 |
| Test spoof prevalence | 79.6% |
| Split strategy | Segment-level |
| Attack labels | Legitimate, simplistic, intermediate, sophisticated |

### B. Curation and Leakage Control

Raw records are segmented using mission and time-reset boundaries, and sliding windows are not allowed to cross segment boundaries. Time differencing is applied where appropriate to reduce shortcut learning from non-stationary receiver variables. Robust scaling is fitted only on the training fold, followed by 0.005–0.995 training-quantile clipping, a conservative range chosen to suppress extreme logging or acquisition artefacts while retaining distribution-tail behaviour that may contain spoofing evidence. The clipping thresholds were not tuned on the test fold. The main protocol retained this clipping range, with all clipping thresholds derived from the training fold. A channel-validity indicator is appended after preprocessing, and the split procedure enforces class coverage in every fold.

The held-out test fold is spoof-heavy: 313 of 393 windows are spoofed, giving 79.6% positive prevalence. Accordingly, F1-score and Brier score are interpreted with specificity, ROC AUC, PR AUC, calibration error, and confusion counts rather than used as sole indicators.

## V. TGTF-Net Architecture

### A. Input Representation

Each curated window is represented as

$$\mathbf{X} \in \mathbb{R}^{N \times F \times T}, \tag{2}$$

where $N = 8$ denotes the receiver channels, $F = 10$ denotes the input features after appending the validity indicator, and $T = 60$ denotes the temporal window length. The primary target is a binary window-level label

$$y^{(w)} \in \{0, 1\}, \tag{3}$$

where 0 denotes legitimate operation, and 1 denotes spoofed operation. Auxiliary channel-level labels

$$\mathbf{y}^{(c)} \in \{0, 1\}^N \tag{4}$$

are retained for additional supervision and interpretability. These auxiliary labels are not treated as independent physical UAV decisions.

### B. Temporal Encoder

For each receiver channel $i$, the input sequence $\mathbf{X}_i \in \mathbb{R}^{F\times T}$ is processed by a shared bidirectional gated recurrent unit (GRU) encoder:

$$\mathbf{h}_i = f_{\text{temp}}(\mathbf{X}_i), \quad i = 1, \ldots, N, \tag{5}$$

where $\mathbf{h}_i \in \mathbb{R}^d$ is the latent channel embedding. The encoder uses shared weights across channels to promote consistent interpretation of receiver-channel dynamics while limiting parameter growth.

### C. Validity-Aware Channel Pooling

A channel-validity mask is used in the temporal pooling path to reduce the influence of unreliable receiver-channel streams. The temporal representation combines validity-aware attention pooling and max pooling:

$$\mathbf{u}_{\text{temp}} = \left[\sum_{i\in\mathcal{V}} \alpha_i \mathbf{h}_i \;;\; \max_{i\in\mathcal{V}} \mathbf{h}_i\right], \tag{6}$$

where $\mathcal{V}$ is the set of valid receiver channels and the attention weights are normalised over $\mathcal{V}$. Attention pooling emphasises channels assigned larger learned weights, while masked max pooling preserves high-salience responses from the valid-channel set.

### D. Tree-Guided Threshold-Bank Branch

The summary branch computes global temporal statistics over the 60-sample window for each channel-feature stream: mean, standard deviation, final value, minimum, maximum, mean first difference, and first-difference standard deviation. These statistics are pooled across valid receiver channels using mean, maximum, and standard-deviation summaries before being passed to a differentiable threshold bank. For summary component $s$ and threshold $\theta_k$, the branch produces

$$r_k^{+} = \sigma(a(s-\theta_k)), \tag{7}$$

$$r_k^{-} = \sigma(-a(s-\theta_k)), \tag{8}$$

where $a$ is a learned scale. In the evaluated configuration, four thresholds are initialised over $[-1, 1]$, with $a = \exp(\ell)$, $\ell = \log 4$, and clamping to $[1, 12]$. This initialisation gives moderately steep but non-saturated sigmoid responses, so the branch behaves as a soft split near each threshold while preserving gradients during early training. The four-level initialisation and scale constraints were fixed whenever the branch was present across the repeated-seed, ablation, grouped cross-segment, and leave-one-segment-out experiments; the no-threshold-bank ablation removed the branch. The resulting responses approximate tree-based threshold behaviour while remaining differentiable [13], [17], [18].

### E. Residual Fusion, Auxiliary Head, and Loss

Temporal and summary representations are combined through a learned gate:

$$\mathbf{g} = \sigma(f_{\text{g}}([\mathbf{u}_{\text{temp}}; \mathbf{u}_{\text{sum}}])), \tag{9}$$

followed by

$$\mathbf{u}_{\text{fused}} = \text{LN}([\mathbf{u}_{\text{temp}}; \mathbf{g}\odot\mathbf{u}_{\text{sum}}]), \tag{10}$$

where $[\cdot;\cdot]$ denotes concatenation, $\odot$ denotes element-wise multiplication, and $\text{LN}(\cdot)$ denotes layer normalisation. The window logit is

$$z^{(w)} = z^{(w)}_{\text{temp}} + \rho_w \Delta z^{(w)}_{\text{sum}} + 0.5\rho_w \Delta z^{(w)}_{\text{fuse}}, \tag{11}$$

where

$$\rho_w = 0.40\,\sigma(\eta_w). \tag{12}$$

The trainable scale logit $\eta_w$ is initialised so that $\rho_w = 0.16$. The window-level spoofing probability is

$$\hat{p}^{(w)} = \sigma\left(z^{(w)}\right). \tag{13}$$

The auxiliary channel logits use an analogous summary correction:

$$z_i^{(c)} = z^{(c)}_{\text{temp},i} + \rho_c \Delta z^{(c)}_{\text{sum},i}, \tag{14}$$

$$\rho_c = 0.40\,\sigma(\eta_c), \tag{15}$$

$$\hat{p}_i^{(c)} = \sigma\left(z_i^{(c)}\right), \quad i = 1, \ldots, N, \tag{16}$$

where $\eta_c$ is initialised so that $\rho_c = 0.16$. The total loss is

$$\mathcal{L} = \mathcal{L}_w + \lambda\mathcal{L}_c, \quad \lambda = 0.05, \tag{17}$$

where $\mathcal{L}_w$ is the window-level binary loss and $\mathcal{L}_c$ is the auxiliary channel-level loss. The weight $\lambda = 0.05$ was selected a priori as a conservative regularisation weight rather than through exhaustive test-set optimisation and was fixed whenever the auxiliary branch was present across the repeated-seed, ablation, grouped cross-segment, and leave-one-segment-out experiments. The auxiliary head provides weak training supervision and channel-level diagnostic scores; inference produces one window-level spoofing decision from $\hat{p}^{(w)}$, and the auxiliary outputs are not interpreted as independent UAV-level decisions.

Figure 2 summarises the proposed TGTF-NET architecture. The model maps an 8-channel, 10-feature, 60-sample receiver window through a shared bidirectional encoder, validity-aware pooling, a threshold-bank summary path, and gated fusion. Solid arrows show forward computation; dashed arrows denote training-only supervision and parameter updates. The deployed output is the window-level spoofing probability, while the auxiliary channel head supports training and interpretation.

## VI. EXPERIMENTAL PROTOCOL

The evaluated TGTF-NET configuration uses batch size 24, learning rate $2.5\times10^{-4}$, weight decay $2\times10^{-4}$, dropout 0.18, temporal hidden size 112, one bidirectional recurrent layer, and four threshold-bank levels, for a total of 806,187 trainable parameters. For each seed or cross-segment fold, the random seed is set and all trainable parameters are freshly initialised before training. The operating threshold is then selected independently on that run's validation fold by maximising F1 subject to a specificity floor of 0.60; the same rule is used for the primary, repeated-seed, grouped cross-segment, and leave-one-segment-out evaluations. Test labels are not used for

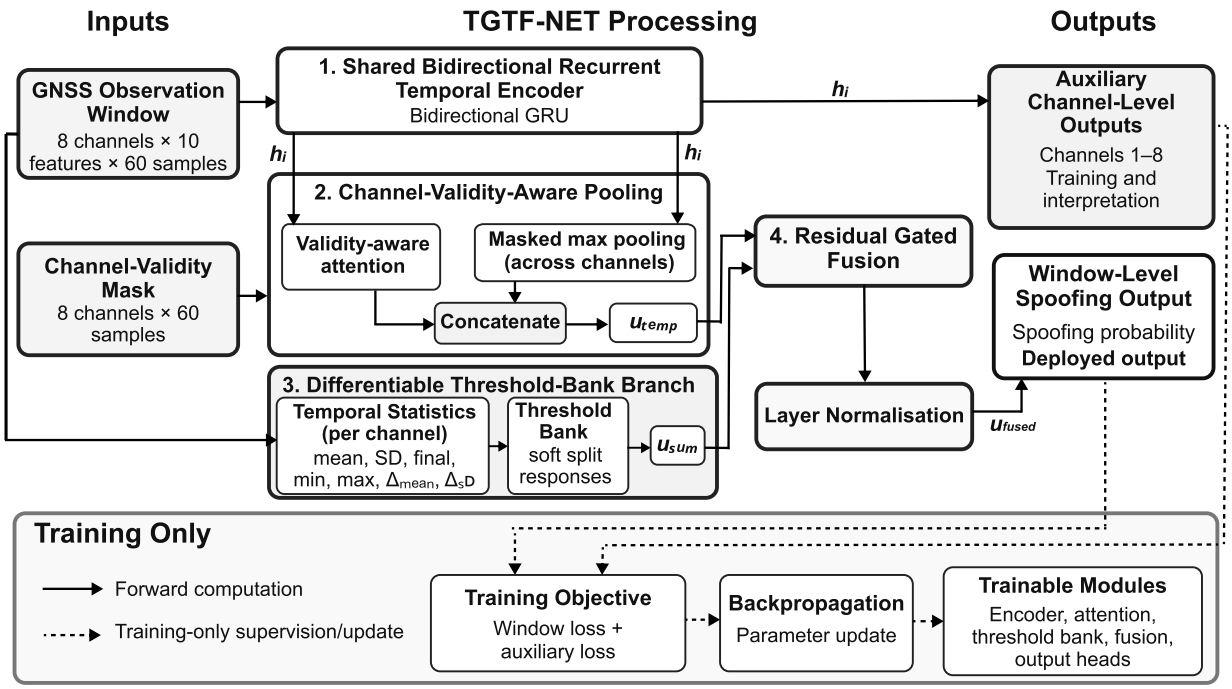



Fig. 2. TGTF-NET architecture for receiver-centric GNSS spoofing detection. Solid arrows denote forward computation; dashed arrows denote training-only supervision and parameter updates. The deployed output is the window-level spoofing probability; auxiliary channel outputs support training and interpretation.

threshold selection. The auxiliary-loss weight and threshold-bank initialisation remain fixed as specified in Section V.

The baseline suite comprises a support vector machine (SVM) [19], gradient boosting [17], a rule-based topology-check comparator, long short-term memory (LSTM) [15], gated recurrent unit (GRU) [14], temporal convolutional network (TCN)-only [16], and multilayer perceptron (MLP)-only models, all evaluated on the identical curated split. For the topology-check comparator, each window is assigned the mean absolute value of the preprocessed `Carrier_Doppler_hz` feature across receiver channels and time. Its reference threshold is the 95th percentile of this statistic among legitimate training windows. The raw score is the statistic divided by twice this reference threshold and clipped to $[0, 1]$, after which calibration and operating-threshold selection use the validation fold. The primary metrics are accuracy, precision, recall, specificity, balanced accuracy, F1-score, Matthews correlation coefficient (MCC), ROC AUC, PR AUC, Brier score, expected calibration error (ECE), post-window processing time, and throughput. ROC and precision–recall analyses provide complementary views of classifier discrimination, particularly under class imbalance [22], [23], while Brier score and ECE assess probability calibration [20], [21].

Additional studies include repeated-seed stability using seeds 42–46, architectural ablation, channel-dropout robustness at 12.5–37.5%, Gaussian-noise robustness with $\sigma = 0.05$, subgroup reporting by motion and attack type, grouped cross-segment validation using five folds, leave-one-segment-out evaluation, and runtime profiling in the evaluated Python environment. Each repeated-seed and cross-segment run uses fresh model initialisation and the same validation-only threshold rule with the 0.60 specificity floor. All experiments use the preprocessing protocol described in Section IV.

TABLE II
PRIMARY HELD-OUT TEST PERFORMANCE OF TGTF-NET

| Metric | Value |
|---|---|
| Accuracy | 0.9186 |
| Precision | 0.9219 |
| Recall | 0.9808 |
| Specificity | 0.6750 |
| Balanced accuracy | 0.8279 |
| F1-score | 0.9505 |
| MCC | 0.7342 |
| ROC AUC | 0.9555 |
| PR AUC | 0.9883 |
| Decision threshold $\tau^*$ | 0.3506 |
| TP / TN / FP / FN | 307 / 54 / 26 / 6 |

## VII. RESULTS

### A. Primary Held-Out Performance

Table II summarises performance on the held-out test split. TGTF-NET achieves accuracy 0.9186, recall 0.9808, F1-score 0.9505, ROC AUC 0.9555, and PR AUC 0.9883 at the validation-selected threshold $\tau^* = 0.3506$. The detector misses 6 of 313 spoofed windows and flags 26 of 80 legitimate windows, corresponding to a specificity of 0.6750 and a false-alarm rate of 0.3250. This operating point prioritises spoofing sensitivity over balanced false-alarm control, which may be appropriate when missed spoofing events carry higher operational risk.

**Algorithm 1** TGTF-NET Training and Calibrated Inference

**Require:** Curated windows $\mathbf{X}$, window labels $y^{(w)}$, channel labels $\mathbf{y}^{(c)}$

**Ensure:** Trained parameters $\theta^*$, selected threshold $\tau^*$

Segment records and construct leakage-aware train/validation/test folds.
Set the run seed and freshly initialise all trainable parameters.
Apply differencing, robust scaling fitted on the training fold, clipping, and channel-validity tagging.
**repeat**
    Sample a mini-batch of windows.
    Encode each receiver channel using the shared bidirectional recurrent encoder.
    Apply channel-validity-aware attention and max pooling to obtain $\mathbf{u}_{\text{temp}}$.
    Compute global 60-sample temporal summary statistics and differentiable threshold-bank responses to obtain $\mathbf{u}_{\text{sum}}$.
    Apply gated residual fusion and layer normalisation to obtain $\mathbf{u}_{\text{fused}}$.
    Compute window-level and auxiliary channel-level losses.
    Update model parameters by backpropagation.
**until** early stopping criterion is met
Sweep candidate thresholds on validation probabilities.
Select $\tau^*$ by maximising validation F1 subject to specificity $\geq 0.60$.
Evaluate once on the held-out test split.

### B. Comparison with Classical and Neural Baselines

Table III compares TGTF-NET with the baselines on the identical split. TGTF-NET gives the highest F1-score and recall among the evaluated thresholded models. Gradient boosting retains higher ROC AUC and PR AUC, with ROC AUC 0.9895 and PR AUC 0.9975, but has lower recall at its selected operating point. The main empirical advantage of TGTF-NET is therefore high-sensitivity thresholded behaviour, not dominance across all metrics. The neural architecture supports window-based temporal inference because the bidirectional recurrent encoder, validity-aware pooling, and threshold-bank branch operate directly on ordered receiver windows. Since the bidirectional encoder uses the complete 60-sample window, the detector is not causal at the individual-sample level. In contrast, gradient boosting relies on precomputed summary features and does not learn a shared temporal representation of the eight synchronised receiver channels.

Figure 3 compares the evaluated classical and neural baselines from two complementary perspectives. The thresholded metrics in Fig. 3(a) show that TGTF-NET gives the highest recall and F1-score at the selected operating point, whereas the ranking metrics and ROC curves in Fig. 3(b) show that gradient boosting yields higher ROC AUC and PR AUC. This distinction bounds the empirical claim: TGTF-NET is a high-recall thresholded configuration for the tested setting, whereas gradient boosting performs better on the reported ranking metrics.

TABLE III
COMPARISON WITH CLASSICAL AND NEURAL BASELINES ON THE HELD-OUT TEST SPLIT

| Model | Accuracy | Recall | Specificity | F1-score | ROC AUC | PR AUC |
|---|---|---|---|---|---|---|
| TGTF-NET | 0.9186 | 0.9808 | 0.6750 | 0.9505 | 0.9555 | 0.9883 |
| SVM | 0.6743 | 0.6613 | 0.7250 | 0.7638 | 0.7362 | 0.8290 |
| Gradient Boost. | 0.7684 | 0.7093 | 1.0000 | 0.8299 | 0.9895 | 0.9975 |
| Topology Check | 0.5954 | 0.5623 | 0.7250 | 0.6888 | 0.7499 | 0.8633 |
| LSTM | 0.8448 | 0.9425 | 0.4625 | 0.9063 | 0.8907 | 0.9725 |
| GRU | 0.8626 | 0.9425 | 0.5500 | 0.9161 | 0.9035 | 0.9753 |
| TCN-Only | 0.7786 | 0.8403 | 0.5375 | 0.8581 | 0.8680 | 0.9673 |
| MLP-Only | 0.7812 | 0.8307 | 0.5875 | 0.8581 | 0.8327 | 0.9541 |

TABLE IV
REPEATED-SEED STABILITY AND ABLATION SUMMARY (FULL MODEL: $n = 5$ SEEDS; ABLATION VARIANTS: $n = 3$ SEEDS; VALUES ARE MEAN $\pm$ STANDARD DEVIATION)

| Variant | Accuracy | F1-score | ROC AUC |
|---|---|---|---|
| Full TGTF-NET | $0.8972 \pm 0.0124$ | $0.9377 \pm 0.0074$ | $0.9469 \pm 0.0110$ |
| Temporal-only | $0.8626 \pm 0.0062$ | $0.9166 \pm 0.0036$ | $0.9043 \pm 0.0044$ |
| No threshold bank | $0.8634 \pm 0.0168$ | $0.9185 \pm 0.0100$ | $0.9243 \pm 0.0142$ |

Figure 4(a) shows that gradient boosting retains the highest average precision, while TGTF-NET remains close to the leading precision–recall profiles. Figure 4(b) confirms the high-sensitivity operating profile of TGTF-NET: spoofed windows are detected with high recall, but legitimate-window specificity remains limited. This operating point may be suitable when missed spoofing events are costlier than false alarms, although false-alarm reduction is still needed before deployment.

### C. Ablation and Repeated-Seed Stability

Across five repeated seeds, TGTF-NET achieves mean accuracy $0.8972 \pm 0.0124$, F1-score $0.9377 \pm 0.0074$, and ROC AUC $0.9469 \pm 0.0110$. Ablation results in Table IV indicate that the threshold-bank branch contributes measurably to performance: removing it reduces mean F1-score to 0.9185. The temporal-only variant remains competitive, with an F1-score of 0.9166, indicating that temporal dynamics provide substantial discriminative information on this dataset and supporting the temporal-first design choice.

Figure 5(a) shows that the full TGTF-NET configuration gives the highest repeated-seed window-level F1 in the ablation study, while removing the threshold-bank branch reduces performance. The temporal-only variant remains competitive, indicating that temporal dynamics provide substantial discriminative information. Figure 5(b) shows that probability calibration remains imperfect; the reliability curve departs from the diagonal, and the reported ECE and Brier score support validation-selected thresholding before deployment.

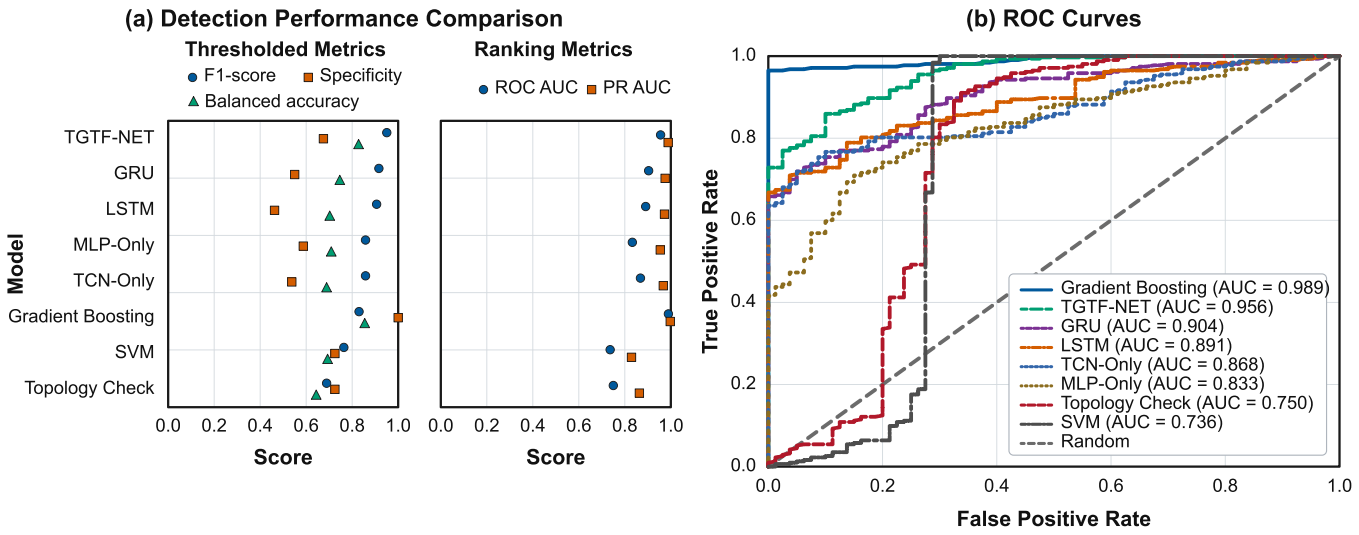



Fig. 3. Baseline comparison using thresholded metrics and ROC analysis. (a) Operating-point and ranking metrics across evaluated models. (b) ROC curves for the same baselines.

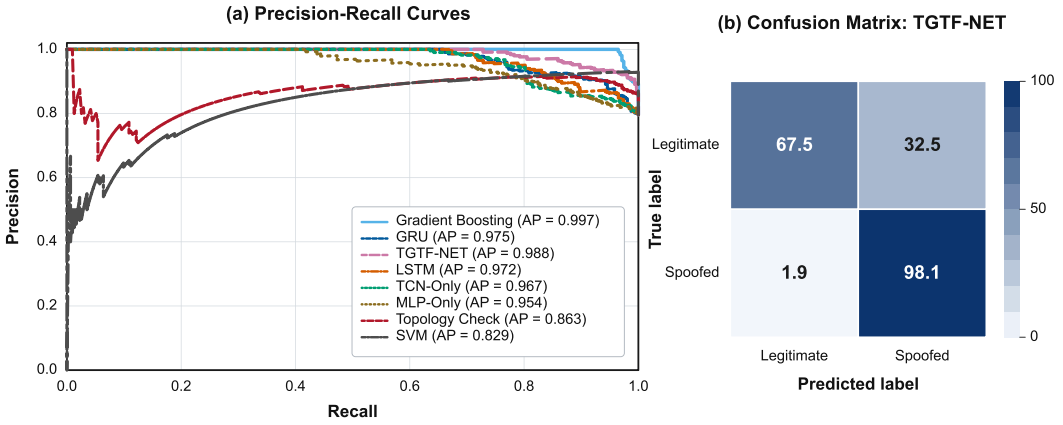



Fig. 4. Precision–recall and operating-point behaviour. (a) Precision–recall curves and average precision (AP) across evaluated baselines. (b) Row-normalised confusion matrix for TGTF-NET at the validation-selected threshold.

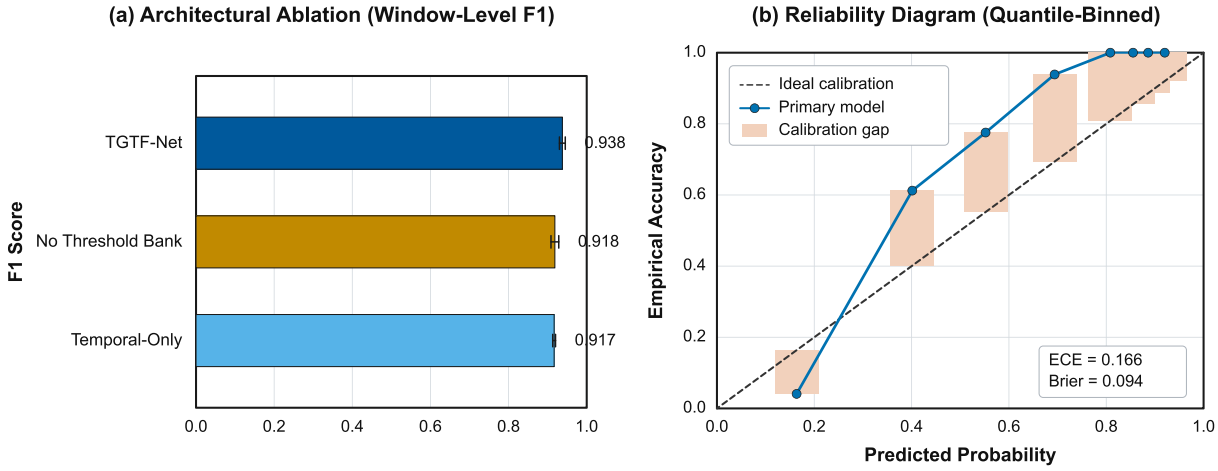



Fig. 5. Ablation and calibration behaviour. (a) Window-level F1 under architectural ablations. (b) Reliability diagram for the selected operating model.

TABLE V
ROBUSTNESS UNDER CHANNEL DROPOUT AND NOISE

| Condition | Accuracy | F1-score | ROC AUC |
|---|---|---|---|
| Clean | 0.9186 | 0.9505 | 0.9555 |
| Channel dropout 12.5% | 0.8779 | 0.9228 | 0.9179 |
| Channel dropout 25.0% | 0.8321 | 0.8904 | 0.8983 |
| Channel dropout 37.5% | 0.7837 | 0.8517 | 0.8683 |
| Gaussian noise ($\sigma = 0.05$) | 0.9135 | 0.9475 | 0.9552 |

TABLE VI
CALIBRATION AND RUNTIME PROFILE

| Metric | Value |
|---|---|
| Brier score | 0.0937 |
| Expected calibration error | 0.1663 |
| Maximum calibration error | 0.2449 |
| Parameters | 806,187 |
| Post-window processing time | 0.297 ms |
| Throughput | 3367 windows/s |

### D. Robustness and Subgroup Results

Table V reports robustness under channel dropout and Gaussian noise. With 12.5% channel dropout, F1-score remains 0.9228, whereas heavier dropout produces progressive degradation. Gaussian noise with $\sigma = 0.05$ leaves performance nearly unchanged. Static windows yield a higher F1-score than dynamic windows, 0.9547 versus 0.9460. This difference is consistent with greater Doppler variation and first-difference non-stationarity under motion, which can reduce the separability of spoofing-related temporal changes. Within the held-out attack-type slices, all six spoofed false negatives occur in the intermediate subgroup, whereas all 205 simplistic spoofing windows are detected.

### E. Calibration and Computational Profile

Table VI reports calibration and runtime. The model obtains a Brier score of 0.0937 and an expected calibration error of 0.1663, so probability calibration remains a deployment concern. Because the test fold is spoof-heavy, with 79.6% positive prevalence, the Brier score is interpreted with ECE, specificity, and confusion counts rather than as a standalone measure. A prevalence-only constant predictor at $p = 0.796$ would have an expected Brier score of approximately $p(1-p) = 0.162$, while an always-spoofed predictor would be penalised by the legitimate-window fraction of approximately 0.204. The observed value of 0.0937 is lower than these prevalence-driven references, but remains influenced by the smaller legitimate subset. In the evaluated Python environment, TGTF-NET processed preconstructed windows at 0.297 ms per window, corresponding to 3367 windows/s, with 806,187 trainable parameters. Timing used batch size 24, one warm-up pass, and three synchronised passes over the 393-window test loader; the reported value is the mean elapsed time divided by the number of windows. The measurement includes data-loader iteration and host-to-device transfer, but excludes receiver sampling, preprocessing, and construction of the 60-sample window. Because the exact accelerator and software stack were not archived, the result is reported only as an environment-specific post-window processing measurement, not as end-to-end or hardware-independent latency.

### F. Cross-Segment Generalisation

Cross-segment evaluation is substantially harder than the primary split. Grouped five-fold evaluation yields mean accuracy 0.7701, F1-score 0.7993, ROC AUC 0.8690, and specificity 0.7759. Leave-one-segment-out evaluation gives aggregate fold-level means of 0.7915 accuracy, 0.6996 F1-score, 0.8125 ROC AUC, and 0.8225 specificity, with substantial dispersion. The reduction from 0.9186 held-out accuracy to 0.7701 grouped accuracy indicates that segment composition, receiver context, and motion regime materially affect generalisation.

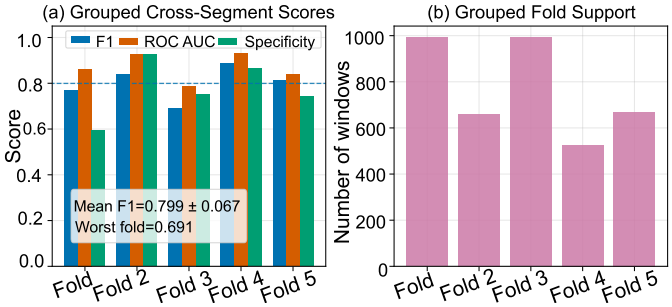



Fig. 6. Grouped cross-segment generalisation. Performance decreases under segment-level distribution shift.

Figure 6 shows the grouped cross-segment behaviour across five folds. The fold dispersion indicates that the primary held-out test result is not deployment-invariant.

## VIII. DEPLOYMENT DISCUSSION

The results support TGTF-NET as a receiver-side anti-spoofing component for UAV-class consumer GNSS devices under the evaluated UAS-emulated conditions, but they are not embedded-system validation. The measured 0.297 ms value is an environment-specific post-window processing time and excludes acquisition and formation of the 60-sample observation window. The detector is therefore window-based rather than a per-sample causal stream processor. Deployment on consumer UAV hardware still requires profiling on representative embedded processors, neural accelerators, or microcontroller-class platforms, including device-specific latency, memory footprint, power consumption, and thermal behaviour.

The selected operating point prioritises spoofing recall over false-alarm suppression. This may be appropriate for safety-oriented detection, but the resulting false-alarm rate on legitimate windows remains non-trivial. In a consumer product, the detector would require alert persistence, operator-facing warning logic, or secondary validation before triggering disruptive navigation responses.

Calibration and distribution shift are also material constraints. The measured ECE and Brier score indicate that raw probabilities are not calibrated posterior estimates. The grouped and leave-one-segment-out results further show that mission and environment shift reduce generalisation relative

to the primary held-out split. The receiver-local design is also compatible with future federated adaptation or generative augmentation without centralising raw receiver observations and may serve as an integrity component for location-oriented security in edge-enabled and Beyond-6G systems; these extensions are not evaluated here. The current findings motivate refinement of calibration, broader dataset validation, and hardware-in-the-loop testing before extending deployment claims beyond the UAS-emulated receiver setting.

## IX. Conclusion

This paper addressed receiver-centric GNSS spoofing detection for UAV-class consumer platforms, where low-cost receivers often operate without auxiliary sensors, antenna arrays, or external reference infrastructure. TGTF-Net was proposed as an edge-oriented Tree-Guided Temporal Fusion Network that integrates bidirectional recurrent temporal encoding, channel-validity-aware attention and max pooling, differentiable threshold-bank evidence, residual temporal-summary fusion, auxiliary channel-level supervision, and validation-selected thresholding. On the curated Mendeley v3 UAS spoofing dataset, TGTF-Net achieves F1-score 0.9505, recall 0.9808, ROC AUC 0.9555, and PR AUC 0.9883, with an environment-specific post-window processing time of 0.297 ms. Among the evaluated models, it gives the highest recall and F1-score at the selected operating threshold, while gradient boosting retains higher ROC AUC and PR AUC. Repeated-seed, ablation, robustness, calibration, and cross-segment results support the temporal-first, tree-guided design while identifying calibration quality, false alarms, and mission-level distribution shift as the main remaining limitations. Within these bounds, receiver-side GNSS integrity is relevant to location-oriented security in future edge-enabled and Beyond-6G communication systems that depend on trustworthy positioning. Future work will examine segment-balanced training, domain-robust objectives, motion-aware augmentation, operating-regime calibration, validation on additional real-world GNSS spoofing corpora, hardware-in-the-loop receiver traces, embedded-hardware latency, power consumption, and false-alarm reduction.

## Acknowledgment

The authors extend their appreciation to the Deanship of Scientific Research and Libraries at Princess Nourah bint Abdulrahman University for funding this work through the Visiting Researcher Program VR-2025-010.

## References

[1] M. L. Psiaki and T. E. Humphreys, "GNSS spoofing and detection," *Proc. of the IEEE*, vol. 104, no. 6, pp. 1258–1270, 2016.

[2] A. Jafarnia-Jahromi, A. Broumandan, J. Nielsen, and G. Lachapelle, "GPS vulnerability to spoofing threats and a review of antispoofing techniques," *Int. Journal of Navigation and Observation*, vol. 2012, Art. no. 127072, 2012.

[3] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner Jr., "Assessing the spoofing threat: Development of a portable GPS civilian spoofer," in *Proc. 21st Int. Tech. Meeting Satellite Division Institute Navigation (ION GNSS 2008)*, 2008, pp. 2314–2325.

[4] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.

[5] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Čapkun, "On the requirements for successful GPS spoofing attacks," in *Proc. 18th ACM Conf. on Computer and Communications Security*, 2011, pp. 75–86.

[6] A. Broumandan, A. Jafarnia-Jahromi, V. Dehghanian, J. Nielsen, and G. Lachapelle, "GNSS spoofing detection in handheld receivers based on signal spatial correlation," in *Proc. 2012 IEEE/ION Position, Location and Navigation Symposium*, 2012, pp. 479–487.

[7] F. Rothmaier, Y.-H. Chen, S. Lo, and T. Walter, "GNSS spoofing detection through spatial processing," *Navigation*, vol. 68, no. 2, pp. 243–258, 2021.

[8] M. Nayfeh, Y. Li, K. Al Shamaileh, V. Devabhaktuni, and N. Kaabouch, "Machine learning modeling of GPS features with applications to UAV location spoofing detection and classification," *Computers & Security*, vol. 126, Art. no. 103085, 2023.

[9] X. Wei, Y. Wang, and C. Sun, "PerDet: Machine-learning-based UAV GPS spoofing detection using perception data," *Remote Sensing*, vol. 14, no. 19, Art. no. 4925, 2022.

[10] L. Alhoraibi, D. Alghazzawi, and R. Alhebshi, "Detection of GPS spoofing attacks in UAVs based on adversarial machine learning model," *Sensors*, vol. 24, no. 18, Art. no. 6156, 2024.

[11] M. S. Korium, M. Saber, A. M. Ahmed, A. Narayanan, and P. H. J. Nardelli, "Image-based intrusion detection system for GPS spoofing cyberattacks in unmanned aerial vehicles," *Ad Hoc Networks*, vol. 163, Art. no. 103597, 2024.

[12] G. Aissou, "A dataset for GPS spoofing detection on unmanned aerial system," *Mendeley Data*, V3, 2022, doi: 10.17632/z7dj3yyzt8.3.

[13] M. Mouzai, M. A. Riahla, A. Keziou, and H. Fouchal, "Exploring multi-channel GPS receivers for detecting spoofing attacks on UAVs using machine learning," *Sensors*, vol. 25, no. 13, Art. no. 4045, 2025.

[14] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, arXiv:1803.01271.

[17] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[18] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[19] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[20] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. Int. Conf. on Machine Learning*, 2017, pp. 1321–1330.

[21] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proc. 22nd Int. Conf. on Machine Learning*, 2005, pp. 625–632.

[22] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[23] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLOS ONE*, vol. 10, no. 3, Art. no. e0118432, 2015.

[24] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.