IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 59, No. 10, October, 2026, pp. 3027-3041
Journal homepage: http://iieta.org/journals/jesa

Sliding-Window Machine Learning for Cross-Gateway Availability and Outage Prediction in Batteryless Light-Aware Environmental IoT Nodes

Saiprasad Potharaju[1*], M.V.V Prasad Kantipudi[2], Ravi Kumar Tirandasu[3], Nekkalapu Srikanth[4], Lingamallu Raghu Kumar[5], Kiranmai Bejjam[6]

[1] Department of CSE, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, India
[2] Department of EnTC, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, India
[3] Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram 522302, India
[4] Department of CSE, Lakireddy Bali Reddy College of Engineering, Mylavaram 521230, India
[5] KG Reddy College of Engineering and Technology, Mylavaram 521230, India
[6] Department of Computer Science and Engineering, Sreyas Institute of Engineering and Technology, Hyderabad 500068, India

Corresponding Author Email: saiprasad.potharaju@sitpune.edu.in

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/jesa.591022

Received: 7 August 2026
Revised: 15 September 2026
Accepted: 20 October 2026
Available online: 31 October 2026

Keywords:
batteryless IoT, sliding window, availability prediction, leave-one-gateway-out validation, Extra Trees, XGBoost, Random Forest

ABSTRACT

Batteryless environmental IoT nodes provided a sustainable alternative to conventional battery-powered sensing, but their operation was strongly affected by intermittent light energy, weak links, and deployment-specific packet loss. This manuscript presented a sliding-window machine-learning framework for predicting future packet availability and outage events in a batteryless light-aware IoT dataset collected from three gateway deployments. Raw packet-level observations were cleaned, regularized into five-minute bins, and transformed into multi-resolution temporal windows of 5, 10, and 15 minutes. Two complementary binary tasks were defined: future availability prediction and future outage early warning. Random Forest, XGBoost, and Extra Trees were evaluated using a chronological 70-15-15 split and a leave-one-gateway-out protocol in which gateways A and B were used for training and gateway C was reserved as an unseen test deployment. The results showed that the proposed formulation achieved strong predictive performance under both validation settings. In the more stringent cross-gateway experiment, Extra Trees with a 10-minute window achieved the best overall performance for both tasks, reaching balanced accuracies of 0.9902 for availability prediction and 0.9903 for outage prediction. Feature-importance analysis indicated that packet-activity, inactive-bin, and light-related temporal descriptors were the dominant predictors. The study demonstrates the potential of short historical windows for early-warning applications in batteryless Internet of Things (IoT) dependability management.

1. INTRODUCTION

As the use of environmental Internet of Things (IoT) continued to expand, the requirement for sensing nodes with extended operational life without needing to be maintained grew. The advent of batteryless and energy harvesting devices was particularly significant since the deployment of large numbers of sensors had relied on replacement of batteries that were costly, hard to access in a distributed environment and undesirable to dispose of. In recent prototypes, communicating with bidirectional Bluetooth Low Energy (BLE), indoor light was found to be useful when appropriate hardware and duty cycling were used [1]. Batteryless BLE and light-based sensing nodes were also developed, demonstrating that hybrid Radio Frequency (RF)- and light-driven sensing architectures could enable reliable environmental monitoring explicitly accounting for energy availability [2]. These advances have ushered in batteryless IoT as a viable platform for smart buildings, environmental monitoring and low-maintenance indoor sensing.

But there were still key issues with reliability. The nodes were not scratched because of permanent hardware failures but also because of a lack of harvested energy, weak link quality and inadequate sensing context. The need for intermittent batteryless networks was stressed as being different from low-power wireless networks due to the power disruptions, irregular charging, and limited battery storage durations [3]. For batteryless BLE nodes, studies indicated that downlink timing, harvesting state and low-power operation constraints affected the performance of the communication [4]. Ambient energy IoT was also assessed in a broader review, which found that addressing practical self-powered sensing demanded the consideration of power generation, storage, communication and power management, not just the optimization of devices [5].

The research gap in this paper was the lack of a general data-driven approach to modelling the future state of the batteryless IoT node availability and outage based on previous multi-sensor behavior, as most of the existing studies focused on energy-aware batteryless IoT reliability as a hardware

optimization problem, or a scheduling problem, or a link-quality optimization problem. Most of the recent models for self-powered green IoT nodes considered a threshold on energy consumption and provided the corresponding probability of service outage, but they typically made explicit assumptions about energy states and energy consumption regimes [6]. Analytical energy consumption models for IoT devices have enhanced the energy profiling of such devices, but were not intended to deduce future packet absence from raw gateway-level sensing logs [7]. Energy forecast reviews for wireless sensor harvesting systems emphasized the benefit of being able to predict the energy availability, but also revealed that the energy prediction was often considered at the energy trace level instead of at the operational reliability level on the part of a gateway [8].

A second gap was related to validation. A prior study on link quality prediction employed temporal features and physical-layer features to predict the quality of the communication link between the two nodes in a low-power wireless link [9]. The improvement of link prediction accuracy and the ability to support routing decisions in wireless sensor networks (WSNs) using gradient-boosting approaches [10]. However, many of these studies did not evaluate if a model trained on a particular gateway deployment would transfer to a new gateway not previously seen. This was a crucial factor because there was variation in propagation, illumination and node-activity patterns for practical deployments of IoT based systems.

To address this gap, this paper positioned batteryless IoT dependability as an intelligent prediction problem for automated reliability management rather than as only a model-performance comparison. The study formulated two operationally distinct future-oriented tasks: packet availability prediction for planning when a node is likely to remain observable, and outage early warning for identifying windows in which complete packet absence is likely. Although the two labels are mathematically complementary, they support different automation actions at the gateway level: availability prediction can support scheduling and data-continuity planning, whereas outage prediction can trigger alerts, gateway-side diagnostics, or maintenance prioritization.

The novelty of the work therefore lies in the complete deployment-aware framework: cleaned packet-level gateway logs were converted into leakage-controlled sliding-window descriptors, three time horizons were compared, and model transfer was evaluated using leave-one-gateway-out validation. This focus is different from conventional link-quality prediction, which usually estimates communication quality between known nodes, and from sensor-fault diagnosis, which usually classifies abnormal sensor readings. Here, the central question was whether recent packet activity, light-aware context, RSSI, and environmental descriptors were sufficient to predict future operational observability in an unseen gateway deployment.

2. LITERATURE REVIEW

The underlying theories behind this work were related to time-series classification, temporal aggregation, wireless link reliability and fault-aware IoT intelligence. The problem of classification in time series transitioned from a matching approach based on the whole sequence to a window-based and feature-rich representation, with the goal of preserving local temporal variations in the sequence while keeping the computation time efficient. WEASEL 2.0 showed that significant temporal substructures could be extracted without very deep models, and showed that randomized dilated transformations could be used to improve fast and memory-constrained time-series classification, which was the general idea that was proven in WEASEL 2.0 [11]. Interpretable temporal features also yielded competitive performance in random time series classifiers with the benefit of lower model complexity [12]. Designed sliding window summaries were used in the current work, with these principles in mind.

Interpretability also played a role in time-series decision systems. The post hoc explanation methods have recently been re-evaluated with regard to the quality of their explanations in time-series classification and it was found that the quality of explanations was dependent on the temporal structure of the data itself and the type of the classifiers [13]. The feature importance was therefore used with caution as a ranking criterion, and not as a causal explanation as used in the present study. The purpose was to determine if the learned decisions were dominated by packet activity, light-related, and environmental descriptors. This was useful to link up model behaviour with batteryless nodes.

Another relevant theory was temporal aggregation. Previous forecasting studies revealed that the accuracy of the model was influenced by temporal aggregation level [14] and that a fixed aggregation level was not optimally correct every time. More empirical studies on temporal aggregation and machine learning revealed that various aggregation levels may be useful for various series if the statistical structure of the series is different [15]. These results led to the deliberate comparison of 5-, 10-, and 15-min windows. For batteryless IoT logs, a very brief window might be responsive but noisy and a longer window might smooth transient evidence. This then required a quantitative validation of the selected window.

IoT fault diagnosis studies focused on identifying faulty sensors, incomplete readings, drift and perception layer faults. For IoT sensor fault prediction and classification, hybrid deep-learning models were applied, but they tend to focus on sensor-value faults instead of future communication availability [16]. Methane sensing algorithms for edge deployment were found to enable compact pipelines for diagnosing the reliability of the perception layer in an IoT system [17]. The effectiveness of boosting in a fault diagnosis scenario was also demonstrated with improved XGBoost methods for vibration-sensor faults [18]. But these papers are primarily concerned with whether the sensed value is correct, or with which types of faults can be classified, not with whether or not a batteryless node would be able to transmit any packet during the next operational horizon.

For automation-oriented IoT systems, reliability prediction also had to be connected to decision support. Recent work on energy-adaptive batteryless sensing showed that sensing decisions must respond to changing energy conditions [19]. Dynamic energy scheduling for energy-harvesting environmental sensors further indicated that the sensing policy should be adjusted according to available energy and monitoring requirements [20]. Lightweight edge-intelligence studies also emphasized that reliability and anomaly decisions should be computationally feasible close to the sensing infrastructure [21]. These studies motivated the present work to evaluate not only classification scores, but also the feasibility of using predictions as gateway-level triggers for rescheduling, warning, and reliability management.

This study was also inspired by multivariate temporal anomaly-detection techniques. Anomaly detection in multivariate time-series was carried out using multi-scale temporal variational autoencoders, which demonstrated the ability of multi-scale representations to learn different temporal patterns [22]. Multi-feature time-series anomaly-detection models also highlighted the need for a number of statistical descriptors to support the successful detection of anomalies [23]. These techniques were effective but they were sometimes more training intensive, less clearly represented or had anomaly labels that were not directly related to the deployment-level dependability.

The challenge remained, however, to bridge the gap between the recent time-window behavior and the availability/outage future label that was present in batteryless environmental IoT data. They also failed to quantify how reliable a reliability model is when it is transferred across gateways. This proposed solution mitigated these drawbacks by converting packet and received signal strength indicator (RSSI) histories to window-wise predictors and provided balanced accuracy, Macro-F1, receiver operating characteristic-area under the curve (ROC-AUC) and confusion matrices for both chronological and leave-one-gateway-out testing. The study shifted from sensor fault detection to quantifiable early-warning reliability prediction in this manner.

3. METHODOLOGY

This study followed a system-level, chronological, and reproducible pipeline that converted raw batteryless environmental IoT logs into window-wise reliability-prediction datasets. The workflow was designed as a gateway-side decision-support loop rather than an on-node algorithm: batteryless nodes transmitted packets, gateways aggregated the received observations, sliding-window features summarized recent behavior, and the trained model produced availability or outage predictions that could support automated reliability-management decisions. The experimental data consisted of three comma-separated files corresponding to three gateway deployments, denoted as Gateway A, Gateway B, and Gateway C. Each raw file contained packet-level measurements from multiple batteryless IoT nodes. The measured variables included timestamp, gateway identifier, node identifier, packet identifier, RSSI, temperature, humidity, pressure, indoor air-quality index, and light intensity.

Only the deployment details available in the logs were used for the analysis. The dataset did not provide direct hardware model numbers, exact physical distances between gateways and nodes, antenna orientation, capacitor voltage, harvested-power measurements, or detailed light-sensor calibration. Therefore, the cross-gateway results were interpreted as operational transfer across recorded gateway deployments rather than as a controlled radio-propagation or electrical-energy characterization experiment. This boundary was explicitly considered while defining labels and discussing generalization.

The system framework in Figure 1 shows the complete loop from light-aware node sensing to gateway aggregation, sliding-window analysis, state prediction, and automated decision support. The first stage loaded three gateway files and standardized column names. A robust parser that could process both datetime strings and epoch representations was used to parse the timestamp field. All numeric variables were converted to floating point. Sentinel values for known embedded-system placeholders and invalid negative codes were replaced with missing values.

The second stage replaced the irregular packet logs by a 5 minute bin-level data set. The number of packets was counted for each 5-minute bin, for each stream (defined as a unique gateway-node pair). A binary variable (active_bin) was created with a value set to 1 if at least one packet was received in the bin, and 0 otherwise. Means, minimums and maximums were calculated where appropriate for the continuous measurements. The mean and minimum values were used to represent RSSI, the mean and maximum values were used to represent light, and the inter-packet time and packet identifier difference were computed between successive packet records. To capture cyclic time patterns but without discontinuities, sine and cosine transformations of hour and day-of-week were also used to derive calendar features.

A layer of a gateway-context was added next. The average (mean) temperature, humidity, pressure, IAQ, light and RSSI values were computed for each gateway and each 5-minute window for all streams that were measured at each gateway during that window. These gateway-context descriptors enabled incorporation of deployment level environmental state without the need for any future information. The leave-one-gateway-out scenario was used with the same protocol, but the gateways and nodes identifiers were not used as predictive features as the test gateway was unseen only as metadata.

Let $x_{\{s,t\}}$ be the vector of the binned features in stream s at 5-minute time bin t. A candidate window size W was set to one of the following window sizes: 5, 10, 15 minutes, with n_W = W/5 bins per window. A temporal feature vector was calculated over the historical sequence $x_{\{s,t-n_W+1\}}$, ..., $x_{\{s,t\}}$ for each stream and each end time t of the window. The following statistics were extracted for each numeric sensor and context variable z: mean, standard deviation, minimum, maximum, last value, and delta over the window. Other reliability descriptors were also calculated: past_packet_count_sum, past_active_ratio, past_inactive_bins, and past_low_light_ratio. The low-light ratio was computed by comparing light_mean with the lower quartile of the light distribution of the same stream.

The prediction horizon was set the same as the length of the input window. So, a 10 min window took the "previous" 10 minutes as input and forecasted the "next" 10 minutes. The future packet count was defined as the sum of all packet_count bins in the future, excluding the current bin, n_W bins in the future. The two targets were then set as:

3.1 Dataset description and statistical profile

The experimental dataset consisted of packet-level observations collected from three gateway deployments, denoted as Gateways A, B, and C. After cleaning and timestamp normalization, the combined dataset contained 3,487,067 packet records from 28 unique gateway-node streams. The deployment covered the period from 10 July 2025 00:00:01 to 17 October 2025 23:59:58, corresponding to approximately 100 days of monitoring. The packet logs were regularized into 5-minute stream-wise bins, resulting in 772,426 bin-level records. These bin-level records formed the basis for sliding-window feature generation. For each gateway-node stream, packet presence was represented using

packet_count and active_bin, while RSSI, temperature, humidity, pressure, IAQ, and light intensity were aggregated as contextual descriptors. The class distribution, as shown in Table 1, varied slightly with the prediction horizon because the future target window length changed from 5 to 10 and 15 minutes. Table 2 summarizes the Variables measured and derived in the proposed framework.

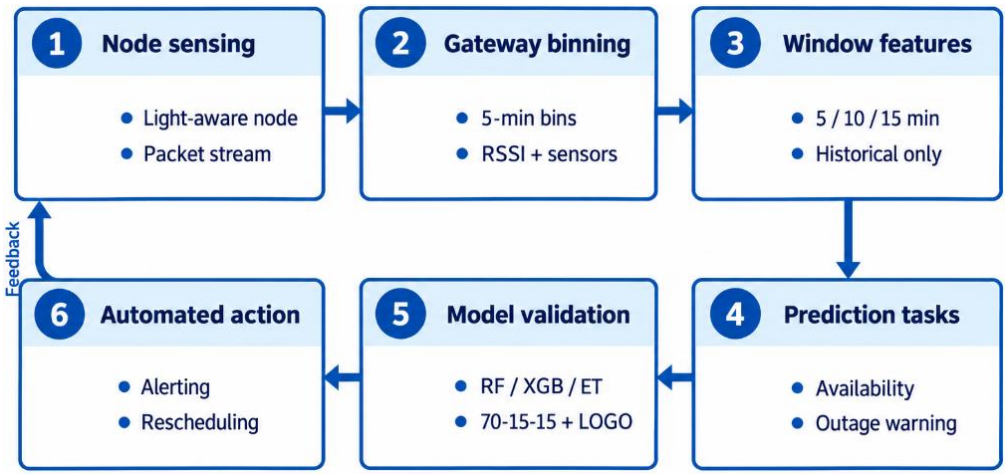


Figure 1. System-level sliding-window reliability-management workflow for batteryless light-aware environmental IoT nodes
Note: Figure 1 emphasizes that prediction is performed at the gateway/edge level after packet aggregation; the batteryless node is not burdened with model inference, and the prediction output can support scheduling, warning, and maintenance actions.

Table 1. Class distribution after window-wise labeling

Window	Total Samples	Available	Unavailable	Outage Class: No Outage	Outage Class: Outage
5 min	772,398	457,207/59.19%	315,191/40.81%	457,207/59.19%	315,191/40.81%
10 min	772,370	468,293/60.63%	304,077/39.37%	468,293/60.63%	304,077/39.37%
15 min	772,342	472,855/61.23%	299,487/38.77%	472,855/61.23%	299,487/38.77%

Table 2. Variables measured and derived in the proposed framework

Column Name	Data Type	Description	Observed/Cleaned Range
time	Float/epoch timestamp	Packet timestamp converted to datetime	10 July 2025 to 17 October 2025
testID	Integer	Experiment/test identifier	Constant value: 1
gateway	Categorical/string	Gateway deployment identifier	a, b, c
packetID	Integer	Packet sequence identifier	17,356 to 1,608,248
nodeID	Categorical/string	Batteryless IoT node identifier	28 unique nodes/streams
rssi	Integer	Received signal strength indicator	-109 to -28 dBm
temperature	Float	Temperature reading	5.13 to 55.72
humidity	Float	Relative humidity	0 to 65.535%
pressure	Float	Atmospheric pressure	743.20 to 1145.96
iaqindex	Float	Indoor air-quality index	177 to 12,946,860
light	Float	Sensor-reported light intensity/illumination proxy	0 to 154,272 after cleaning

In this study, light intensity was not used as a direct measurement of harvested energy or stored capacitor voltage. Instead, it was treated as an indirect environmental proxy for energy-harvesting opportunity. This interpretation is appropriate because the considered batteryless/light-aware IoT nodes depend on ambient illumination for energy availability; however, the dataset does not include direct measurements of harvested power, capacitor voltage, or energy-buffer state. Therefore, the relationship examined in this work is operational rather than electrical: recent light conditions, together with packet activity, RSSI, and environmental context, are used to predict whether the node will be observable at the gateway in a future time window. Consequently, the target labels should be interpreted as future packet availability and outage states, not as direct batteryless hardware failure or explicit energy-storage depletion.

This clarification is important for reproducibility: the learned labels represent future gateway observability and future packet absence. They should not be interpreted as direct labels of photovoltaic harvesting efficiency, capacitor discharge, or permanent hardware damage. The communication protocol and physical deployment geometry were treated as part of the observed gateway-node context because they were not separately parameterized in the dataset.

The following equations define the sliding-window formulation, target construction, and evaluation metric used in the study.

$$n_W = \frac{W}{5}, \quad W \in \{5,10,15\} \tag{1}$$

$$X_{s,t}(W) = \phi\left(x_{s,t-n_W+1}, \ldots, x_{s,t}\right) \tag{2}$$

$$F_{s,t}(W) = \sum_{k=1}^{n_W} \text{packet_count}_{s,t+k} \tag{3}$$

$$y_{\text{avail}} = \begin{cases} 1, & F_{s,t}(W) \geq 1 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$y_{\text{outage}} = \begin{cases} 1, & F_{s,t}(W) = 0 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$\text{Balanced Accuracy} = \frac{1}{2}\left[\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right] \quad (6)$$

As shown in Table 3, Task 1 was availability prediction; in which a positive label meant at least one packet was expected to be received within the next W-minute horizon. Outage early-warning (Task 2) was when the positive label meant that no packets were received in the next W-minute horizon. These two were complementary but useful for different purposes. Task 1 evaluated the observability of the node and Task 2 focused on early detection of complete future packet absence.

Table 3. Problem definition and binary target formulation

Task	Binary Target Definition	Purpose
Task 1: Availability	1 if at least one packet appeared in the next W-minute horizon; otherwise 0.	Operational availability prediction
Task 2: Outage	1 if no packet appeared in the next W-minute horizon; otherwise 0.	Outage early-warning prediction

Target columns, quantities derived from the future and horizon metadata were stripped from the data prior to model training, ensuring there is no information leakage. The final feature matrix had only historical window statistics, time descriptors, gateway-context variables, and packet-history features. Infinite values were replaced with missing values and then the missing feature values were imputed by using median of training data. For the chronological experiment, the set of data for each task was divided by the end time of the window in order to examine the chronological aspect, and was divided into 70% training, 15% validation, and 15% testing. This design maintained a sequential and not randomized order in time. The 200,000-row limit was applied only to the training set for computational efficiency. Unless otherwise stated, the validation and test sets were kept as is.

For classifiers, the following models were used: Random Forest, Extra Trees, and XGBoost. Random Forest and Extra Trees were based on 250 trees, with unrestricted depth, class_weight="balanced", random_state = 42, and all available CPU cores. XGBoost used 350 estimators, max_depth = 4, learning_rate = 0.05, subsample = 0.90, colsample_bytree = 0.90, binary logistic objective, log-loss evaluation, random_state = 42, and n_jobs = -1. These values were selected after preliminary validation-oriented trials that favored stable performance and computational feasibility over exhaustive search. Because the dataset was large and the study compared multiple windows, tasks, and validation settings, the main reported results were generated from fixed-seed single runs; the absence of repeated stochastic runs was retained as a study limitation. The second validation protocol examined deployment generalization. For the primary leave-one-gateway-out experiment, all data from Gateways A and B were used for training while Gateway C was used only for testing. This protocol was more stringent than a normal chronological split because the test gateway was not used during training. It tested whether the learned temporal reliability patterns transferred to a new deployment domain. The predicted outputs were interpreted as gateway-level automation signals: a high availability probability can support sensing continuity planning, while a high outage probability can trigger node rescheduling, gateway-side alerting, maintenance inspection, or energy-aware sampling adjustments.

The following performance metrics were computed: accuracy, balanced accuracy, positive-class precision, positive-class recall, positive-class F1, Macro-F1, Weighted-F1, ROC-AUC and AUPRC. An imbalanced availability and outage labels were important for the balanced accuracy. Macro-F1 was used to assess class-balanced F1 performance. ROC-AUC and AUPRC were used to evaluate probability-ranking quality. The confusion matrices were produced for each model-window-task combination. When the fitted classifier returned feature_importances_ a feature-importance plot was created.

A simple example may help to illustrate the method. If W = 10 minutes, then n_W = 2 since the base bin size is 5 minutes. If the previous two bins had packet_count values [3, 0] then there was a recent state that was active and a recent state that was inactive. In this case, if the next two bins contained [0, 0] then future_packet_count_sum would be 0. In this situation, target_available_next_window = 0 is assigned to Task 1 and target_outage_next_window = 1 is assigned to Task 2. Had there been at least one packet in the future, the 1 in the Task 1 label would have been followed by a 0 in the Task 2 label. This example demonstrated that the targets were directly related to future packet observability, as opposed to raw environmental values.

Algorithm 1: Sliding-window reliability prediction for batteryless IoT nodes

Input: Raw gateway files D_A, D_B, D_C; window set W = {5, 10, 15} min
Output: Predicted availability and outage labels, metrics, confusion matrices
1: Load raw packet logs and standardize column names
2: Parse timestamps and remove invalid sensor values and sentinel codes
3: For each gateway-node stream, resample packets into 5-min bins
4: Compute packet_count, active_bin, RSSI, environmental, light, and time features
5: Compute gateway-level context features at each 5-min timestamp
6: For each W in {5, 10, 15}:
7: n_W <- W/5
8: For each stream and each window end time t:
9: Extract historical rolling statistics over bins t-n_W + 1 to t
10: Calculate future_packet_count_sum over bins t + 1 to t + n_W
11: Set Task1 target = 1 if future_packet_count_sum >= 1 else 0
12: Set Task2 target = 1 if future_packet_count_sum = 0 else 0
13: Remove future-derived columns from the feature matrix
14: Train RF, Extra Trees, and XGBoost classifiers
15: Evaluate using chronological 70-15-15 split and LOGO validation
16: Save metrics, predictions, confusion matrices, and feature importance

Algorithm 1 for the proposed method is summarized as follows: The design aimed to allow another researcher to replicate the study by using the same raw gateway files, 5 minute base aggregation, sliding-window sizes, target definitions, leakage-control rules, model hyperparameters and validation protocols. Both chronological and leave one gateway out validation enabled the study to separate temporal performance from actual cross-deployment generalization. This validation was especially significant since recent "lightweight" machine-learning studies on the IoT focused on the need for anomaly detection in light of limited energy resources, and recent batteryless scheduling research revealed the challenges of maintaining consistent sensing under non-trivial energy dynamics [23, 24]. The integration of embedded systems, edge computing, and machine learning also facilitated the application of small-sized tree sets to intelligent reliability analysis of IoT systems [25].

4. EXPERIMENTAL RESULTS AND DISCUSSION

The experimental analysis was conducted in two phases: chronological (70-15-15) and leave-one-gateway-out (LOGO). The chronological split evaluated the extent to which the proposed sliding-window features predicted future availability and outage with temporally separate training and testing. The leave-one-gateway-out protocol examined the generalization of the learned reliability patterns when the trainer for the Gateway C was completely removed from the learning set. The notebook initially loaded and cleaned the raw gateway files, created the 5-minute bin-level dataset, created the feature importance plots, created the line charts, created the confusion matrices, trained the three classifiers, and saved the metrics, predictions, confusion matrices, line charts, and feature importance plots. Table 4 presents the full results of the test for the chronological split for both tasks, all three window lengths and all three classifiers. These findings indicated that the 15-minute window, in general, was the best chronological window. Random Forest with 15-min window had the highest overall test accuracy of 0.9888, balanced accuracy of 0.9896, Macro-F1 of 0.9880, ROC-AUC value of 0.9970, and AUPRC value of 0.9985 for Task 1. Although Random Forest performed best overall in the balanced accuracy, XGBoost performed very well in the future availability cases with a higher ROC-AUC of 0.9985 and a higher balanced accuracy of 0.9897. The confusion matrix comparison between the models in Task 1 (Figure 2) indicated that the 15-minute models outperformed the shorter models in terms of false unavailable and false available decisions.

Table 4. Complete chronological 70-15-15 test performance for Task 1 and Task 2

Task	Win	Model	Acc	BA	Prec+	Rec+	F1+	Macro-F1	W-F1	ROC-AUC	AUPRC
T1-Avail.	5	Random Forest	97.30%	96.94%	97.10%	98.54%	97.81%	97.15%	97.29%	99.64%	99.78%
T1-Avail.	5	XGBoost	97.86%	97.79%	98.39%	98.11%	98.25%	97.75%	97.86%	99.72%	99.83%
T1-Avail.	5	Extra Trees	97.88%	97.81%	98.39%	98.14%	98.27%	97.77%	97.88%	99.68%	99.81%
T1-Avail.	10	Random Forest	98.76%	98.77%	99.29%	98.72%	99.00%	98.68%	98.76%	99.76%	99.87%
T1-Avail.	10	XGBoost	98.73%	98.75%	99.29%	98.66%	98.98%	98.65%	98.73%	99.85%	99.91%
T1-Avail.	10	Extra Trees	98.75%	98.76%	99.27%	98.71%	98.99%	98.67%	98.75%	99.78%	99.88%
T1-Avail.	15	Random Forest	98.88%	98.96%	99.59%	98.62%	99.10%	98.80%	98.88%	99.70%	99.85%
T1-Avail.	15	XGBoost	98.86%	98.97%	99.65%	98.54%	99.09%	98.79%	98.87%	99.85%	99.92%
T1-Avail.	15	Extra Trees	98.85%	98.93%	99.54%	98.62%	99.08%	98.77%	98.85%	99.79%	99.89%
T2-Outage	5	Random Forest	95.27%	94.20%	98.20%	89.44%	93.62%	94.93%	95.23%	99.59%	99.18%
T2-Outage	5	XGBoost	97.88%	97.81%	97.08%	97.48%	97.28%	97.77%	97.89%	99.71%	99.53%
T2-Outage	5	Extra Trees	97.88%	97.81%	97.06%	97.48%	97.27%	97.77%	97.88%	99.70%	99.49%
T2-Outage	10	Random Forest	98.75%	98.77%	97.89%	98.82%	98.35%	98.67%	98.75%	99.75%	99.50%
T2-Outage	10	XGBoost	98.53%	98.42%	98.11%	97.98%	98.05%	98.43%	98.53%	99.85%	99.73%
T2-Outage	10	Extra Trees	98.75%	98.76%	97.90%	98.80%	98.35%	98.67%	98.75%	99.82%	99.68%
T2-Outage	15	Random Forest	98.87%	98.96%	97.69%	99.30%	98.49%	98.79%	98.87%	99.74%	99.39%
T2-Outage	15	XGBoost	98.85%	98.97%	97.51%	99.45%	98.47%	98.77%	98.85%	99.85%	99.73%
T2-Outage	15	Extra Trees	98.85%	98.93%	97.69%	99.26%	98.47%	98.77%	98.85%	99.80%	99.59%

Random Forest (with a 15-minute window) again yielded the highest overall performance with an accuracy of 0.9887, a balanced accuracy of 0.9896, a Macro-F1 score of 0.9879, ROC-AUC of 0.9974 and AUPRC of 0.9939 for Task 2 (chronological split). The balanced accuracy of XGBoost for 15 minutes was the highest at 0.9897 and the ROC-AUC was 0.9985. The outage confusion matrix comparison in Task 2 (Figure 3) indicated that the 15-minute window in most cases resulted in lower outage confusion rates. The results indicated that, for the chronological setting, there was enough information in the temporal context to allow the model to distinguish between sustained inactivity and short gaps in the packet stream. Table 5 includes the results of the more stringent leave-one-gateway-out tests. This protocol was used to train the model on Gateways A and B, and to test it on Gateway C. The best overall accuracy for Task 1 was obtained by Extra Trees with a 10-minute window (accuracy = 0.9900, balanced accuracy = 0.9902, Macro-F1 = 0.9898, ROC-AUC = 0.9978 and AUPRC = 0.9985). The Extra Trees 10-minute model successfully predicted 99,973 unavailable instances and 127,506 available instances, with only 800 unsuccessful predictions of available and 1,501 unsuccessful predictions of unavailable. This can be seen in the LOGO Task 1 confusion matrix table in Figure 4. This showed good generalization for the unseen gateway and showed that the gateway identifier was not needed as a feature in the model.

Extra Trees with a 10-minute window again achieved the best overall performance with accuracy of 0.9900, balanced accuracy of 0.9903, Macro-F1 of 0.9899, ROC-AUC of 0.9979, and AUPRC of 0.9968 for Task 2 of leave-one-gateway-out validation. The model was used to predict the number of outage windows and no-outage windows during the period, as depicted in Figure 5, and the model correctly predicted 99,997 outage windows and 127,494 no-outage windows, and yielded 776 false no-outage windows and 1,513 false outage windows. The model with the highest outage

recall was XGBoost with the window length of 15 minutes, indicating that this is the most aggressive model for detecting outages. Extra Trees with a 10-minute window, however, showed the best compromise between all the metrics, accuracy, balanced accuracy, Macro-F1, ROC-AUC and AUPRC.

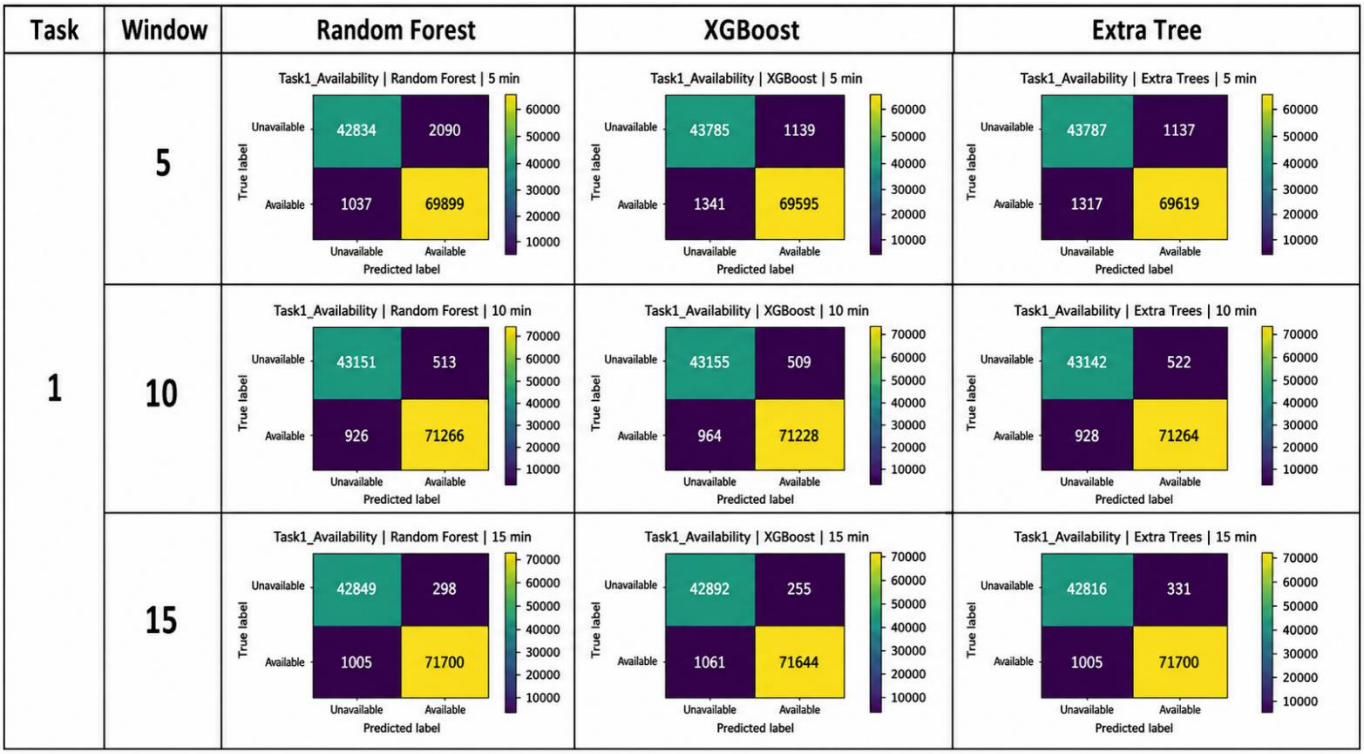


Figure 2. Chronological split confusion-matrix comparison for Task 1 availability prediction

Note: The chronological Task 1 as shown in Figure 2 matrices compare availability prediction errors across windows and classifiers and shows that longer windows reduced false availability/unavailability decisions.

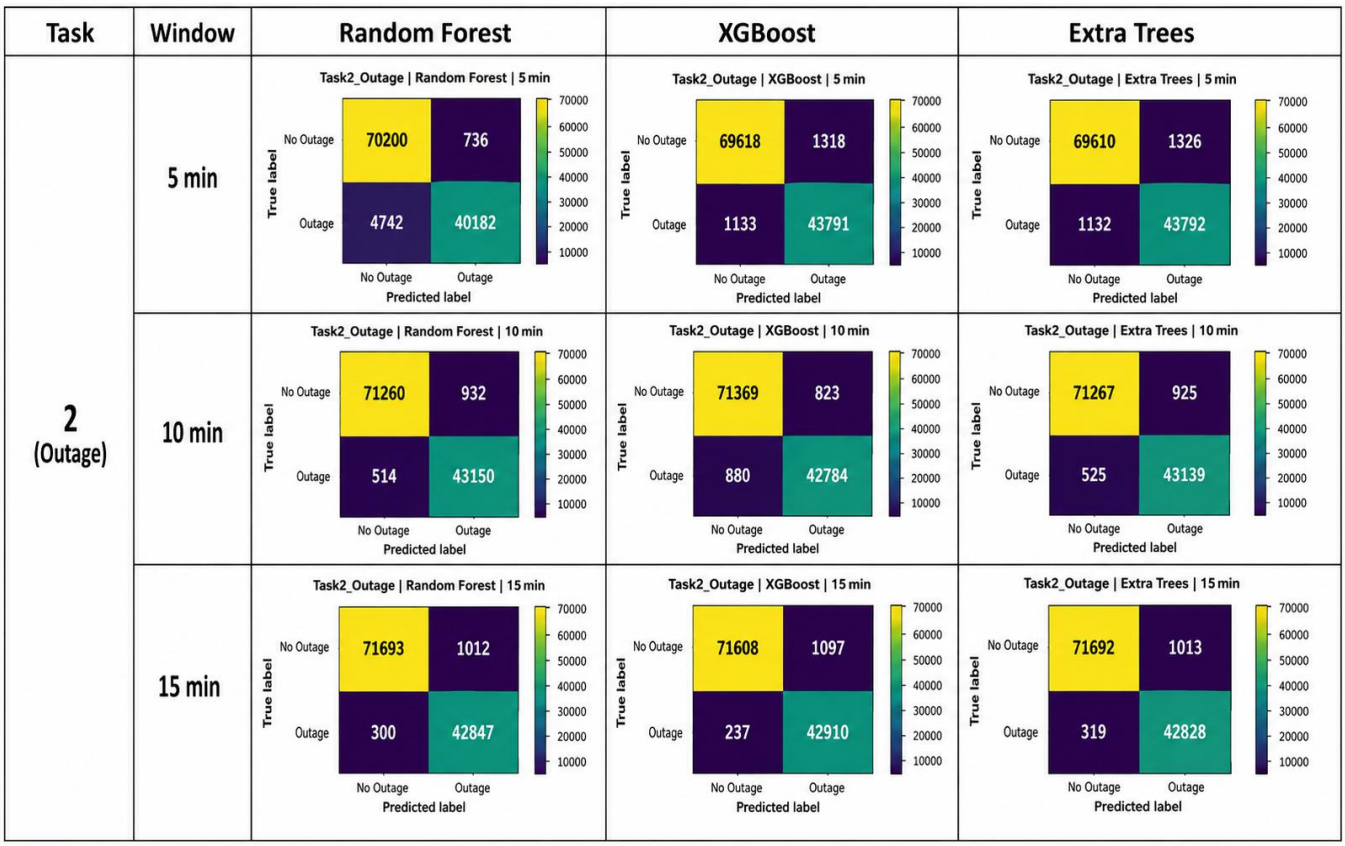


Figure 3. Chronological split confusion-matrix comparison for Task 2 outage prediction

Note: The chronological Task 2 matrices, as shown in Figure 2, summarize no-outage/outage classification and demonstrate improved separation of outage states at longer windows.

Table 5. Complete leave-one-gateway-out test performance for Task 1 and Task 2 using unseen Gateway C

Task	Win	Model	Acc	BA	Prec+	Rec+	F1+	Macro-F1	W-F1	ROC-AUC	AUPRC
T1-Avail.	5	Random Forest	96.72%	96.44%	95.22%	99.04%	97.09%	96.66%	96.71%	99.71%	99.78%
T1-Avail.	5	XGBoost	98.47%	98.47%	98.74%	98.49%	98.62%	98.45%	98.47%	99.74%	99.81%
T1-Avail.	5	Extra Trees	98.51%	98.50%	98.72%	98.59%	98.66%	98.49%	98.51%	99.70%	99.78%
T1-Avail.	5	ARIMA	97.82%	97.70%	98.22%	98.21%	98.22%	97.70%	97.82%	97.70%	97.56%
T1-Avail.	5	Heuristic	97.82%	97.70%	98.22%	98.21%	98.22%	97.70%	97.82%	97.70%	97.56%
T1-Avail.	10	Random Forest	98.77%	98.76%	98.94%	98.86%	98.90%	98.75%	98.77%	99.61%	99.77%
T1-Avail.	10	XGBoost	98.95%	98.99%	99.46%	98.67%	99.06%	98.94%	98.95%	99.75%	99.83%
T1-Avail.	10	Extra Trees	99.00%	99.02%	99.38%	98.84%	99.11%	98.98%	99.00%	99.78%	99.85%
T1-Avail.	10	ARIMA	98.53%	98.60%	99.36%	98.01%	98.68%	98.51%	98.53%	98.96%	99.08%
T1-Avail.	10	Heuristic	98.53%	98.60%	99.36%	98.01%	98.68%	98.51%	98.53%	98.60%	98.50%
T1-Avail.	15	Random Forest	98.78%	98.81%	99.31%	98.52%	98.91%	98.76%	98.78%	99.53%	99.71%
T1-Avail.	15	XGBoost	98.92%	98.99%	99.69%	98.38%	99.03%	98.90%	98.92%	99.73%	99.82%
T1-Avail.	15	Extra Trees	98.87%	98.92%	99.48%	98.51%	98.99%	98.85%	98.87%	99.72%	99.81%
T1-Avail.	15	ARIMA	97.86%	98.13%	99.53%	97.05%	98.27%	97.73%	97.87%	98.13%	98.44%
T1-Avail.	15	Heuristic	97.86%	98.13%	99.53%	97.05%	98.27%	97.73%	97.87%	98.48%	98.87%
T2-Outage	5	Random Forest	96.93%	96.69%	98.69%	94.38%	96.49%	96.88%	96.93%	99.68%	99.50%
T2-Outage	5	XGBoost	98.41%	98.43%	97.89%	98.56%	98.23%	98.39%	98.41%	99.75%	99.66%
T2-Outage	5	Extra Trees	98.52%	98.51%	98.24%	98.44%	98.34%	98.50%	98.52%	99.70%	99.58%
T2-Outage	5	ARIMA	98.47%	98.45%	98.28%	98.28%	98.28%	98.45%	98.47%	98.45%	97.36%
T2-Outage	5	Heuristic	98.47%	98.45%	98.28%	98.28%	98.28%	98.45%	98.47%	99.21%	98.15%
T2-Outage	10	Random Forest	98.79%	98.78%	98.53%	98.71%	98.62%	98.77%	98.79%	99.71%	99.47%
T2-Outage	10	XGBoost	98.27%	98.20%	98.33%	97.70%	98.02%	98.24%	98.27%	99.75%	99.63%
T2-Outage	10	ARIMA	98.53%	98.60%	97.50%	99.20%	98.34%	98.51%	98.53%	98.60%	97.06%
T2-Outage	10	Heuristic	98.53%	98.60%	97.50%	99.20%	98.34%	98.51%	98.53%	98.96%	97.43%
T2-Outage	10	Extra Trees	99.00%	99.03%	98.51%	99.23%	98.87%	98.99%	99.00%	99.79%	99.68%
T2-Outage	15	Random Forest	98.72%	98.75%	98.11%	98.96%	98.53%	98.70%	98.72%	99.56%	99.25%
T2-Outage	15	XGBoost	98.91%	98.99%	97.97%	99.56%	98.76%	98.90%	98.91%	99.68%	99.51%
T2-Outage	15	Extra Trees	98.85%	98.90%	98.09%	99.30%	98.69%	98.83%	98.85%	99.73%	99.58%
T2-Outage	15	ARIMA	97.86%	98.13%	95.23%	99.22%	97.18%	97.73%	97.87%	98.13%	94.78%
T2-Outage	15	Heuristic	97.86%	98.13%	95.23%	99.22%	97.18%	97.73%	97.87%	98.48%	95.16%

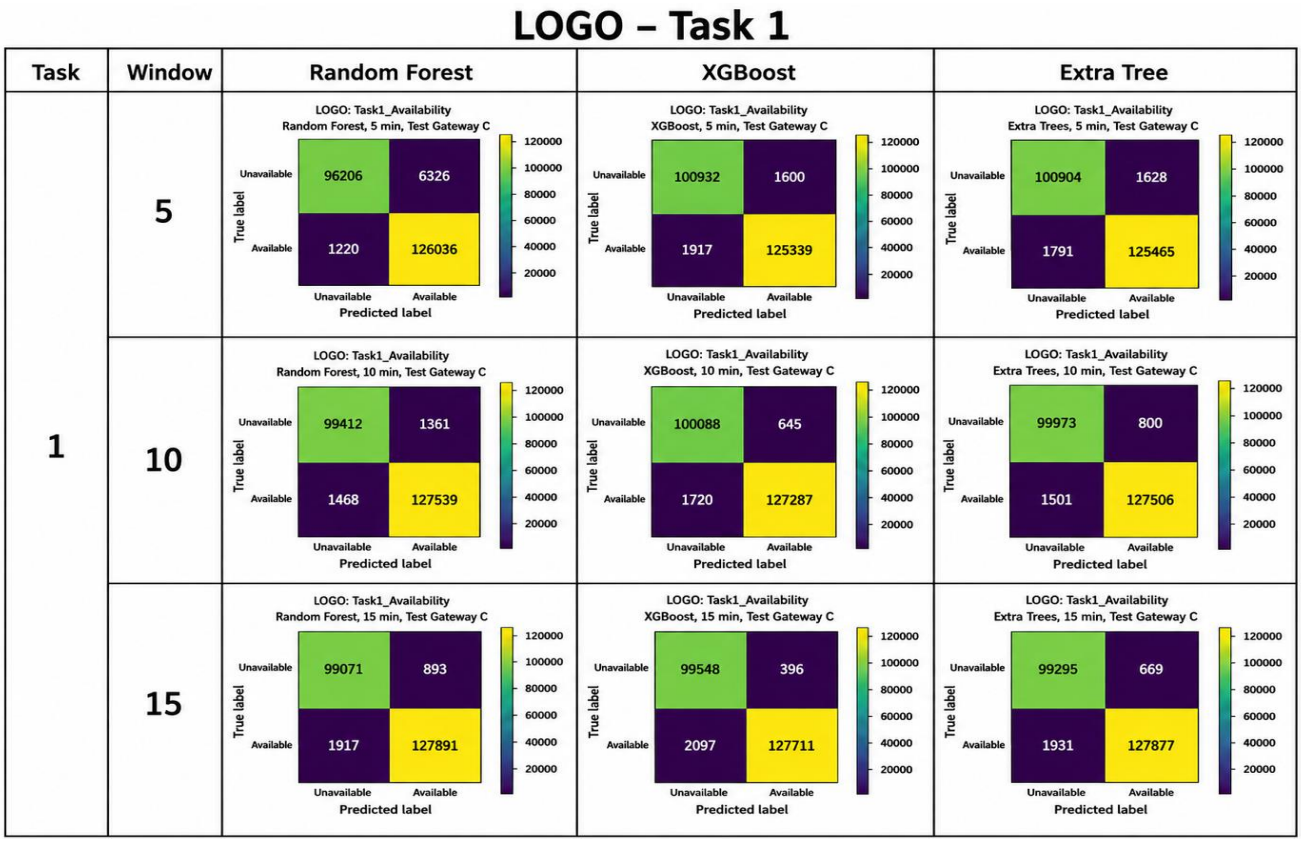


Figure 4. Leave-one-gateway-out (LOGO) confusion-matrix comparison for Task 1 using unseen Gateway C

Note: The LOGO Task 1 matrices shown in Figure 4, transfer performance when Gateway C was completely unseen during training; the 10-minute Extra Trees model produced the strongest balanced profile.

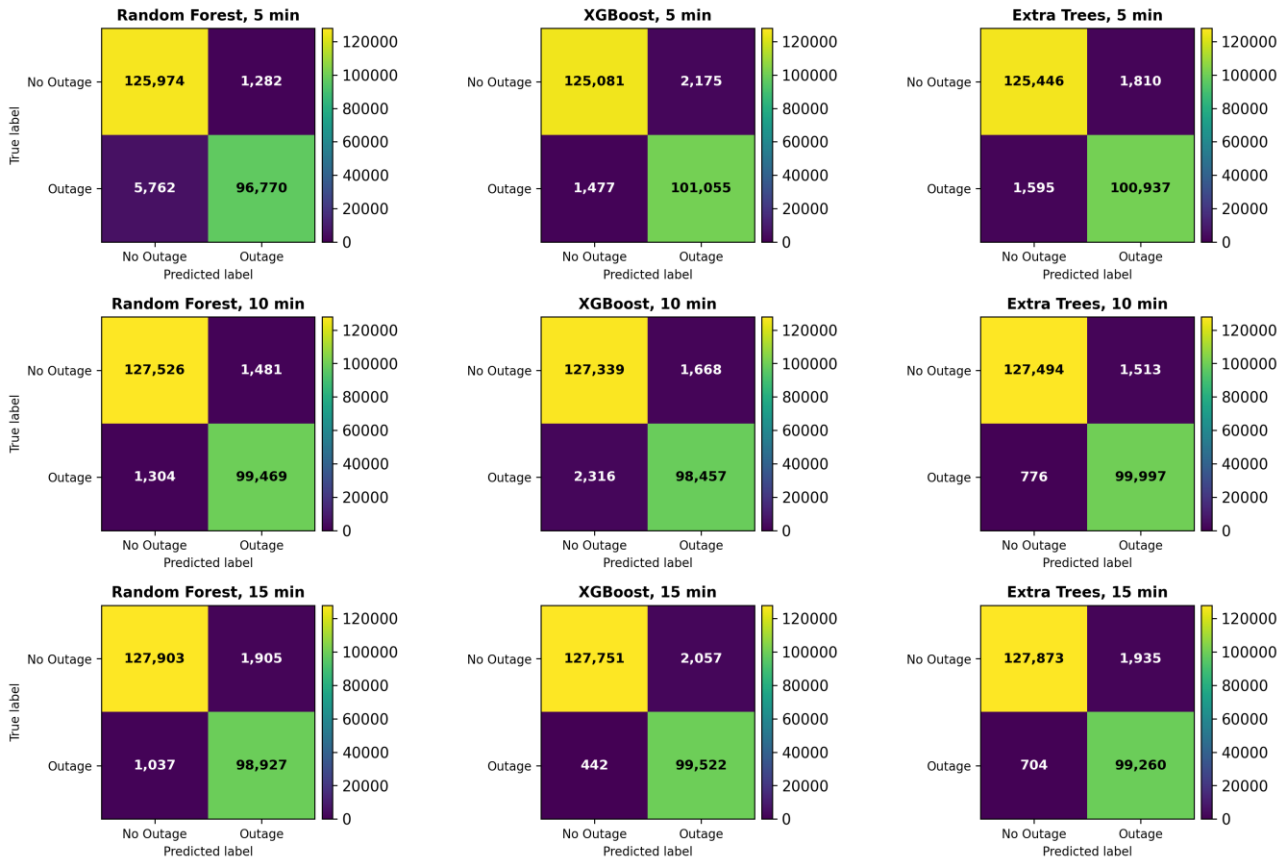


Figure 5. Leave-one-gateway-out (LOGO) confusion-matrix comparison for Task 2 outage prediction using unseen Gateway C

To better illustrate the window-wise behavior under LOGO validation, the line-chart comparisons in Figures 6 and 7 were used. For Task 1, Extra Trees and XGBoost clearly improved over Random Forest at the 5-minute window, and the 10-minute Extra Trees model achieved the strongest overall balance across balanced accuracy, Macro-F1, ROC-AUC, and AUPRC. For Task 2, the 10-minute Extra Trees model gave the most balanced cross-gateway profile, while the 15-minute XGBoost model showed very high outage recall. This indicates that the most suitable model may depend on the automation objective: balanced reliability management favors Extra Trees at 10 minutes, whereas a more aggressive warning system may favor the model with higher outage recall.

The results were operationalized with the help of feature-importance analysis. The top 7 models for chronological Task 1 were dominated by 'active_bin_win_max, past_packet_count_sum, active_bin_win_mean, past_inactive_bins, packet_count_win_mean, past_active_ratio, packet_count_win_max'. Figure 8 suggested that the recent packet activity/inactivity was the primary factor in determining the future availability of a batteryless node, a factor one would expect when the successful transmission of the packet directly depended on the energy and communication readiness. The other environmental variables (temperature, pressure, humidity, RSSI) did not seem to be the main factors, since they were ranked later than the other ones.

A baseline comparison experiment using heuristic persistence and ARIMA forecasting was also conducted. The results show that temporal packet activity was a strong predictor, but the proposed tree-based sliding-window models achieved higher balanced accuracy, Macro-F1, ROC-AUC, and AUPRC than the simple baselines in the more stringent LOGO setting. This supports the claim that the proposed framework did not rely only on packet persistence, but benefited from the combined use of packet history, light-aware indicators, link-quality features, temporal context, and gateway-level descriptors.

To further examine cross-gateway generalization, an additional leave-one-gateway-out experiment was conducted by reserving Gateway A as the unseen test deployment and using Gateways B and C for training. The results are presented in Table 6. The performance trend was consistent with the earlier A + B → C validation reported in Table 4, indicating that the proposed sliding-window framework did not depend on a single held-out gateway. Across both Task 1 and Task 2, the models retained high balanced accuracy, Macro-F1, ROC-AUC, and AUPRC values, showing that the learned temporal reliability patterns were transferable across different gateway deployments. This additional B + C → A experiment strengthens the evidence that the proposed model generalizes beyond one specific gateway-test set.

The feature importance plot for Task 2 (Figure 9) was quite similar. The most important predictors were active_bin_win_max, active_bin_win_last, past_active_ratio, active_bin_win_mean, past_inactive_bins and active_bin_win_min. Features derived from light also showed up in the top 20, such as light_max_win_mean, light_mean_win_min, light_mean_win_max, light_mean_win_mean, light_max_win_max,

light_max_win_min. This was relevant, because the prediction of the outages should better be linked to the energy-harvesting condition. Light-related features were thus found to support the hypothesis that recent illumination history provided useful information to predict future lack of packets.

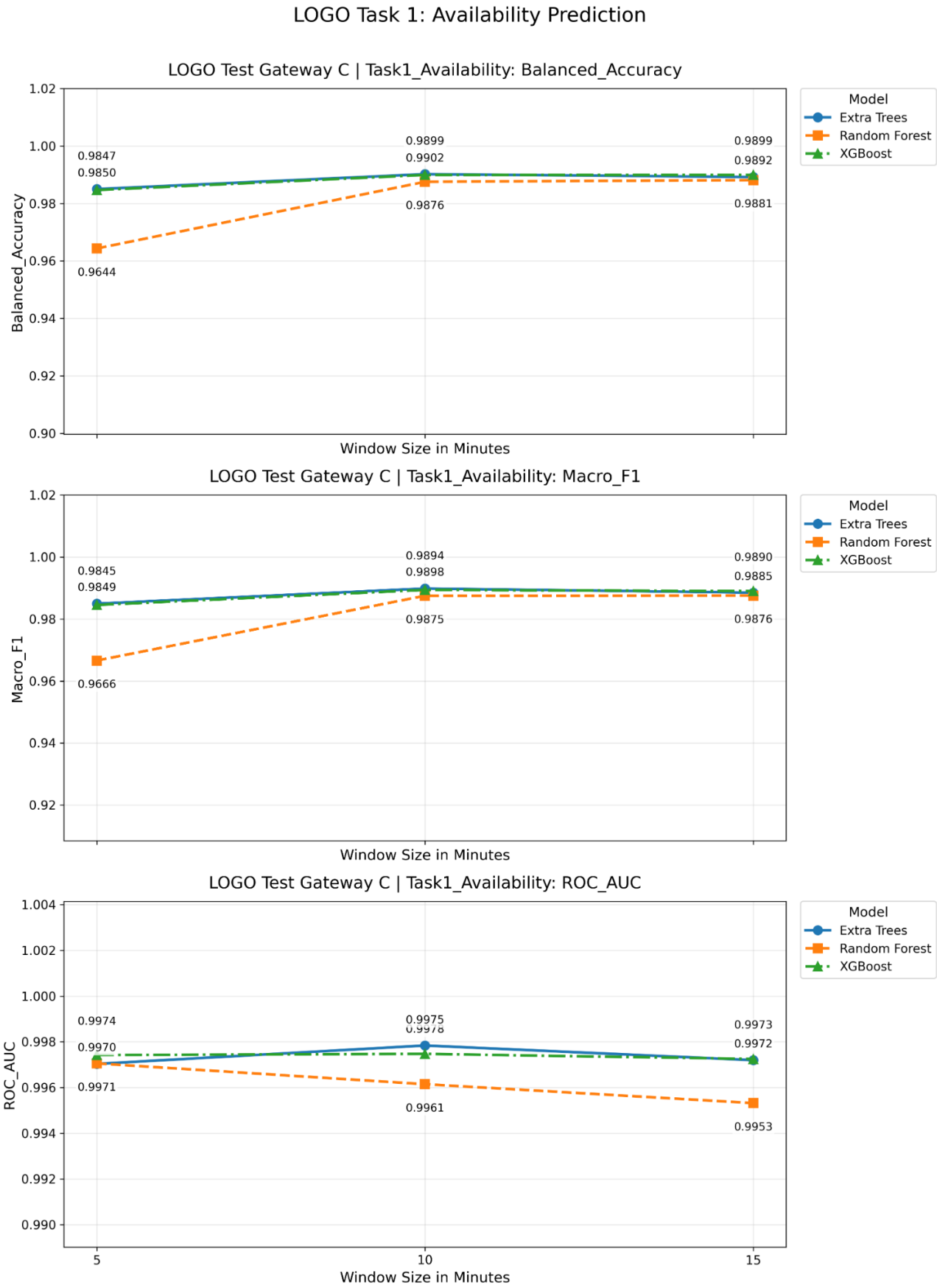


Figure 6. Leave-one-gateway-out (LOGO) Task 1 line-chart comparison across window sizes and models

Note: The line plots described in Figure 6 compare balanced accuracy, Macro-F1, and ROC-AUC for Task 1 and illustrate how the 10-minute window improves cross-gateway generalization.

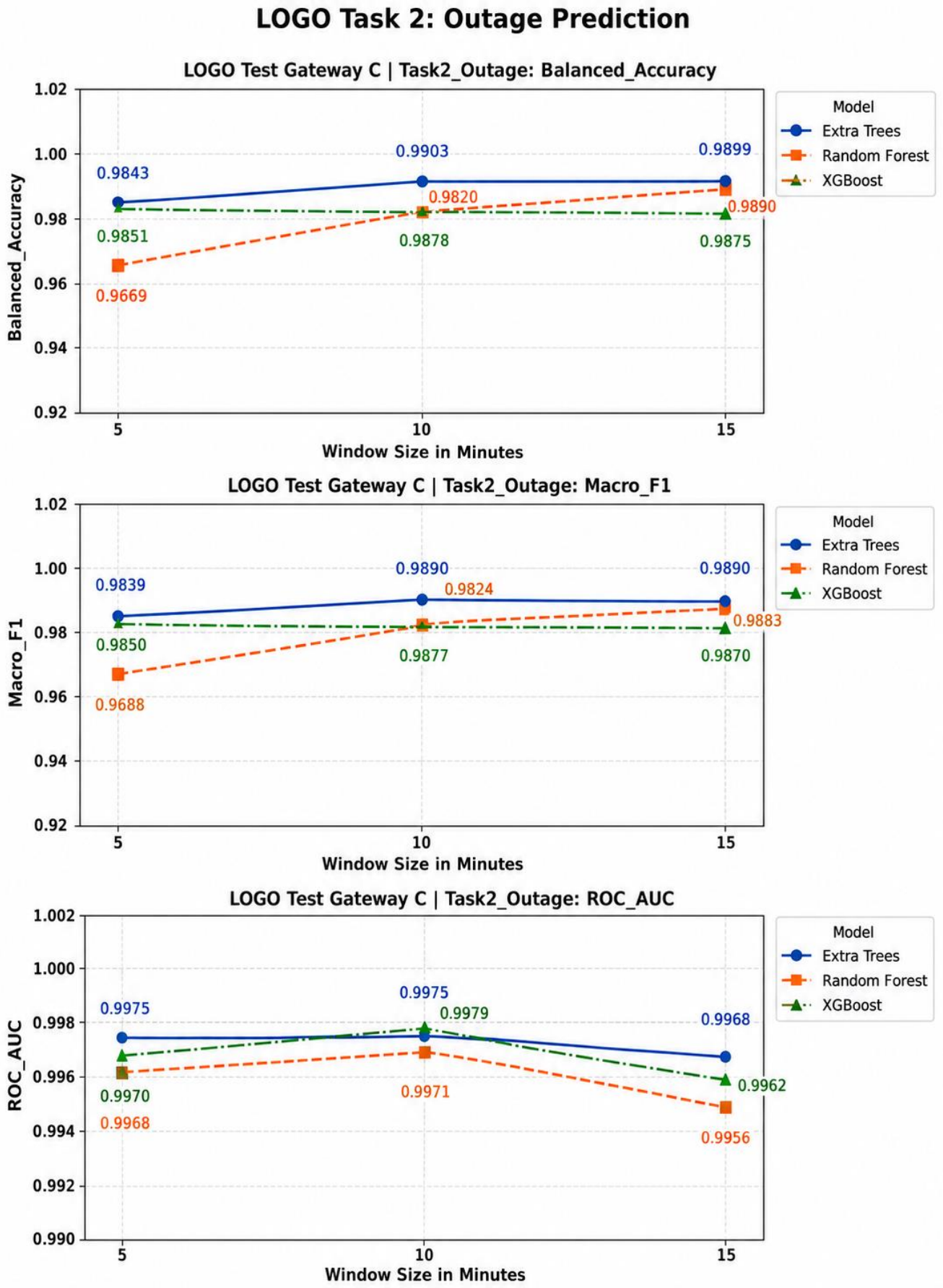


Figure 7. Leave-one-gateway-out (LOGO) Task 2 line-chart comparison across window sizes and models

Note: The line plots in Figure 7 compare Task 2 performance across windows and show that the 10-minute Extra Trees model was the most balanced choice, while XGBoost was competitive for high-recall outage detection.

Table 6. Complete leave-one-gateway-out test performance for Task 1 and Task 2 using unseen Gateway A

Task	Win	Model	Acc	BA	Prec+	Rec+	F1+	Macro-F1	W-F1	ROC-AUC	AUPRC
T1-Avail.	5	Random Forest	95.24%	94.98%	95.74%	96.30%	96.02%	95.04%	95.23%	98.92%	99.26%
T1-Avail.	5	Extra Trees	95.49%	95.34%	96.32%	96.13%	96.22%	95.32%	95.49%	99.00%	99.32%
T1-Avail.	5	XGBoost	95.48%	95.35%	96.39%	96.03%	96.21%	95.31%	95.48%	99.02%	99.36%
T2-Outage	5	Random Forest	95.20%	94.90%	94.62%	93.39%	94.00%	95.00%	95.19%	98.84%	98.16%
T2-Outage	5	Extra Trees	95.49%	95.33%	94.28%	94.54%	94.41%	95.31%	95.49%	98.99%	98.45%
T2-Outage	5	XGBoost	95.49%	95.36%	94.19%	94.65%	94.42%	95.32%	95.49%	99.03%	98.54%
T1-Avail.	10	Random Forest	97.43%	97.37%	98.22%	97.62%	97.92%	97.28%	97.43%	99.29%	99.54%
T1-Avail.	10	Extra Trees	97.44%	97.38%	98.22%	97.63%	97.93%	97.29%	97.44%	99.41%	99.62%
T1-Avail.	10	XGBoost	97.31%	97.29%	98.26%	97.38%	97.82%	97.16%	97.32%	99.46%	99.68%
T2-Outage	10	Random Forest	97.42%	97.35%	96.20%	97.06%	96.63%	97.27%	97.43%	99.30%	98.84%
T2-Outage	10	Extra Trees	97.45%	97.38%	96.22%	97.11%	96.66%	97.30%	97.45%	99.43%	99.05%
T2-Outage	10	XGBoost	97.36%	97.32%	95.92%	97.18%	96.55%	97.20%	97.36%	99.49%	99.18%
T1-Avail.	15	Random Forest	98.25%	98.29%	99.06%	98.15%	98.60%	98.13%	98.25%	99.53%	99.67%
T1-Avail.	15	Extra Trees	98.25%	98.29%	99.06%	98.15%	98.60%	98.13%	98.25%	99.60%	99.74%
T1-Avail.	15	XGBoost	98.14%	98.22%	99.13%	97.90%	98.51%	98.02%	98.14%	99.64%	99.80%
T2-Outage	15	Random Forest	98.25%	98.29%	96.91%	98.43%	97.66%	98.13%	98.25%	99.52%	99.23%
T2-Outage	15	Extra Trees	98.25%	98.29%	96.89%	98.44%	97.66%	98.13%	98.25%	99.57%	99.32%
T2-Outage	15	XGBoost	98.13%	98.22%	96.47%	98.57%	97.51%	98.00%	98.13%	99.65%	99.40%

In summary, the results agreed with the initial hypothesis that by using packet activity, link quality, light and environmental context, short-term sliding window summaries of these data could be used to predict future reliability states of batteryless IoT nodes. The chronological results indicated that the 15-minute window achieved the best in-domain performance and the leave-one-gateway-out results indicated that the 10-minute window generalized best to unseen gateways. This was not a one-off occurrence, but rather a valuable lesson learned in deployment. A longer window of 60 minutes or more might be suitable for this deployment in order to capture the observed temporal distribution, while an intermediate window of 10 minutes might be suitable if this model was applied to a new gateway in order to provide a good balance between responsiveness and generalization.

The results were similar to the results in the Related Theory section. The results of temporal aggregation studies indicated that the behavior of the model would differ slightly with different levels of temporal aggregation, and these differences were quantified in the present study for 5-, 10-, and 15-minute windows of aggregation. The feature-importance results supported the time-series classification theory, where local temporal features were expected to be used to make accurate predictions, and recent activity features turned out to be the most important ones in the models. The present study took the concept of reliability monitoring from the literature on IoT fault diagnosis to future packet availability and outage, as opposed to sensor value faults. The principal problem was that XGBoost did not necessarily provide the best recall or top ranking but did not always provide the best classification balance. This was probably due to the class-weighted boosting this class did, which might give preference to hard positive examples. This could be helpful from an operational perspective when alarm rates were more expensive than outages. But for a balanced approach to reliability management for the gateway level, Extra Trees with a 10-minute window proved to be the most defensible LOGO option.

The feature ranking shown in Figure 8 indicates that recent packet activity and inactive-bin descriptors were the main indicators for future availability.

To strengthen the interpretability analysis, SHAP-based explainability was added for the proposed tree-based sliding-window model for Window size 15. The SHAP results confirmed that packet-activity and inactivity descriptors were the dominant predictors for both availability and outage prediction. As shown in Figure 10, for availability prediction, packet-activity features contributed 76.91% of the total mean absolute SHAP importance, followed by light/energy-proxy features at 9.64%, environmental-context features at 8.05%, and RSSI/link-quality features at 3.76%. As shown in Figure 11, for outage prediction, packet-activity features contributed 75.15%, followed by environmental-context features at 9.60%, light/energy-proxy features at 9.10%, and RSSI/link-quality features at 3.80%. The most influential individual SHAP features included active_bin_win_max, active_bin_win_sum, packet_count_win_max, past_inactive_bins, and past_packet_count_sum. These results support the earlier feature-importance findings and show that the model decisions are mainly driven by recent packet continuity, while light-aware and contextual descriptors provide additional explanatory support for batteryless IoT outage prediction.

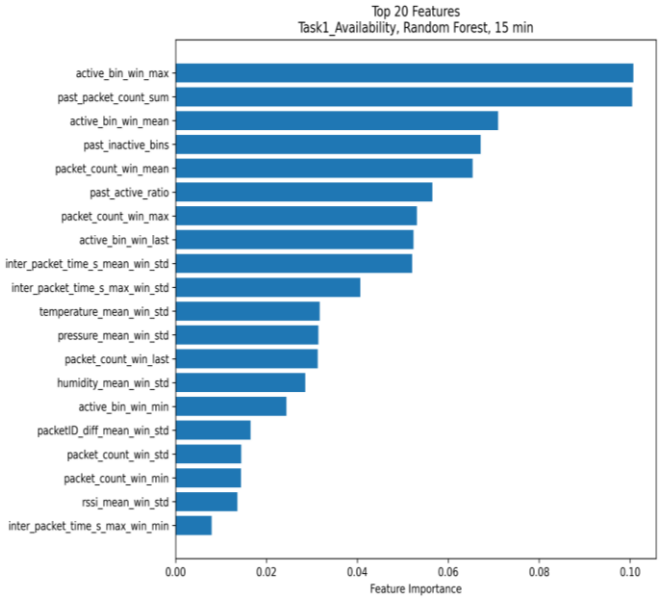


Figure 8. Top 20 feature importance for Task 1 best chronological model

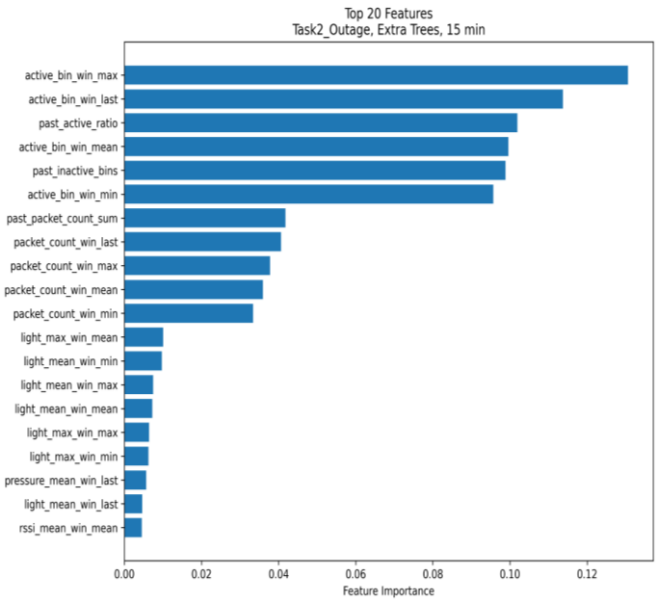


Figure 9. Top 20 feature importance for Task 2 outage prediction

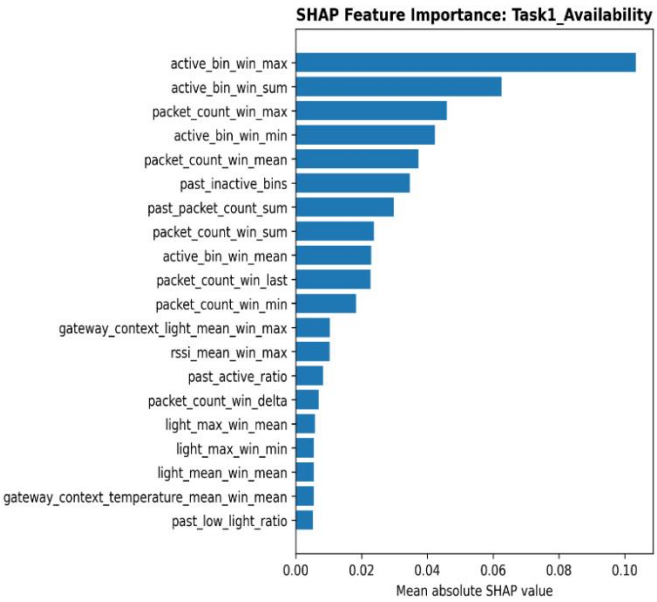


Figure 10. SHAP feature importance for Task 1

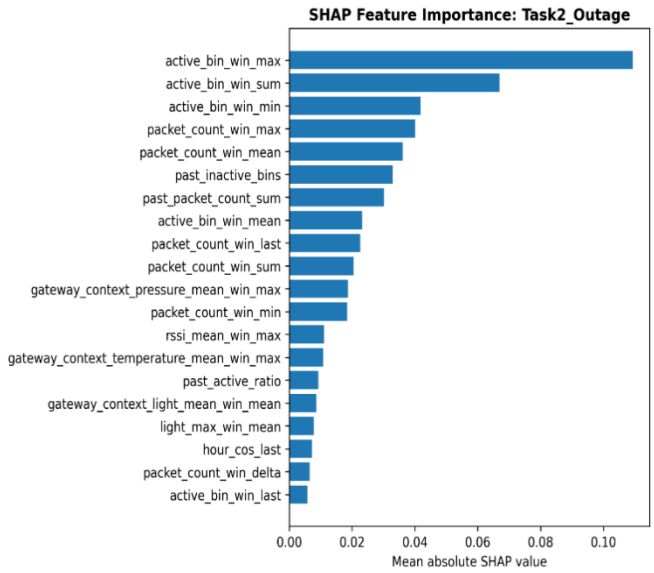


Figure 11. SHAP feature importance for Task 2 outage prediction

To quantify the contribution of different feature groups, an ablation study was performed under the 10-minute leave-one-gateway-out setting, where Gateways A and B were used for training, and Gateway C was reserved as the unseen test deployment. As shown in Table 7, the same Extra Trees classifier was used in all ablation settings, and only the feature groups were changed. The full feature set achieved balanced accuracies of 0.9897 for availability prediction and 0.9892 for outage prediction. The packet-activity-only model also achieved high performance, confirming that recent packet continuity is a strong temporal signal. However, it was lower than the full model, particularly for outage AUPRC. Removing light/energy-proxy features reduced balanced accuracy by 0.0035 for availability and 0.0039 for outage, indicating that illumination-related descriptors provide useful information for batteryless node reliability. Removing RSSI/link-quality and environmental/context features produced smaller reductions, suggesting that these variables provide complementary deployment context. Overall, the ablation study supports the SHAP findings and confirms that the proposed model benefits from multiple feature groups rather than relying only on packet persistence.

Table 7. Ablation study under 10-minute window level

Task	Ablation_Setting	Feature_ Count	BA	Macro_ F1	ROC_ AUC	AUPRC	Recall	Precision
T1-Avail.	Full features	53	98.97%	98.93%	99.73%	99.78%	98.77%	99.35%
T1-Avail.	Packet-activity only	11	98.54%	98.50%	99.08%	99.00%	98.33%	99.03%
T1-Avail.	No packet-activity features	42	98.96%	98.92%	99.70%	99.77%	98.77%	99.34%
T1-Avail.	No light/energy-proxy features	40	98.62%	98.63%	99.58%	99.64%	98.85%	98.74%
T1-Avail.	No RSSI/link-quality features	44	98.89%	98.86%	99.67%	99.73%	98.74%	99.25%
T1-Avail.	No environmental/context features	36	98.88%	98.85%	99.69%	99.75%	98.73%	99.23%
T1-Avail.	Light + RSSI + environment + time only	43	98.96%	98.92%	99.71%	99.77%	98.74%	99.35%
Task2_Outage	Full features	53	98.92%	98.89%	99.66%	99.39%	99.08%	98.43%
Task2_Outage	Packet-activity only	11	98.57%	98.52%	99.08%	98.16%	98.78%	97.92%
Task2_Outage	No packet-activity features	42	98.91%	98.88%	99.68%	99.42%	99.05%	98.43%
Task2_Outage	No light/energy-proxy features	40	98.54%	98.56%	99.56%	99.21%	98.20%	98.55%
Task2_Outage	No RSSI/link-quality features	44	98.88%	98.85%	99.67%	99.36%	98.97%	98.44%
Task2_Outage	No environmental/context features	36	98.66%	98.66%	99.65%	99.33%	98.54%	98.44%
Task2_Outage	Light + RSSI + environment + time only	43	98.81%	98.78%	99.64%	99.36%	98.86%	98.41%

4.1 Computational cost and deployment feasibility

The proposed framework is computationally feasible for practical deployment because most of the processing involves simple 5-minute aggregation, sliding-window statistical feature extraction, and tree-based classification over 53 engineered features in the full setting. Model training can be performed offline at the gateway server or cloud level, while real-time deployment only requires updating recent packet, light, RSSI, and environmental summaries and applying the trained classifier to the latest window. Since the batteryless nodes are resource-constrained, the prediction model is not intended to run on the sensing node itself; instead, it can be deployed at the gateway, edge device, or local monitoring server where packet logs are already collected. The tree ensembles used in this work are compatible with edge/server deployment; however, exact embedded-device memory size and per-sample inference latency were not benchmarked on a microcontroller and are identified as future deployment-validation requirements.

5. CONCLUSIONS

This study presented a sliding-window machine-learning framework for automated reliability management in batteryless light-aware environmental IoT nodes. The work addressed a practical automation problem: a batteryless node could become temporarily unavailable because of intermittent energy harvesting, link variation, or local context, and this future unavailability should be predicted before it affects monitoring continuity. The proposed pipeline converted raw gateway-level packet logs into leakage-controlled temporal windows of 5, 10, and 15 minutes and evaluated Random Forest, XGBoost, and Extra Trees under chronological and leave-one-gateway-out validation.

The main finding was that window length strongly influenced both performance and generalization. In the chronological 70-15-15 split, Random Forest with a 15-minute window achieved the best overall test performance for both availability and outage prediction. In the more rigorous LOGO setting, Extra Trees with a 10-minute window achieved the best overall cross-gateway performance, reaching balanced accuracy of 0.9902 for Task 1 and 0.9903 for Task 2. Feature-importance analysis further showed that packet-activity descriptors, inactive-bin patterns, and light-related features provided the strongest predictive evidence. These results demonstrated that recent short-term behavior could serve as an effective early-warning signal for batteryless IoT reliability.

The study had limitations. It used a limited number of gateway deployments, and the available logs did not include exact physical distances, direct harvested-energy measurements, or capacitor-voltage traces. The labels were derived from future packet presence, so they represented operational availability rather than direct hardware failure or measured energy-buffer depletion. Therefore, the results should be interpreted within this deployment boundary rather than generalized to all batteryless IoT systems.

Future work should repeat leave-one-gateway-out validation across larger and more diverse gateway deployments, evaluate low-light-only stress periods, integrate explicit energy-buffer measurements where available, and benchmark model size and inference latency on gateway-class edge hardware. A lightweight control-oriented version of the model could also be tested for adaptive duty-cycle management and automated gateway selection in real deployments.

DATA AVAILABILITY STATEMENT

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

CODE AVAILABILITY STATEMENT

Code is available at: https://github.com/psaiprasadcse/batteryless-Internet-of-Things.

REFERENCES

[1] Sultania, A.K., Famaey, J. (2022). Batteryless bluetooth low energy prototype with energy-aware bidirectional communication powered by ambient light. IEEE Sensors Journal, 22(7): 6685-6697. https://doi.org/10.1109/JSEN.2022.3153097

[2] Landivar, J.F., Botirov, K., Sallouha, H., Katz, M., Pollin, S. (2024). Batteryless BLE and light-based IoT sensor nodes for reliable environmental sensing. In 2024 IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Valencia, Spain, pp. 1-6. https://doi.org/10.1109/PIMRC59610.2024.10817258

[3] Fu, S., Narayanan, V., Wymore, M.L., Deep, V., Duwe, H., Qiao, D. (2023). No battery, no problem: Challenges and opportunities in batteryless intermittent networks. Journal of Communications and Networks, 25(6): 806-813. https://doi.org/10.23919/JCN.2023.000033

[4] Sultania, A.K., Delgado, C., Blondia, C., Famaey, J. (2022). Downlink performance modeling and evaluation of batteryless low power BLE node. Sensors, 22(8): 2841. https://doi.org/10.3390/s22082841

[5] Chatterjee, A., Lobato, C.N., Zhang, H.W., et al. (2023). Powering Internet-of-Things from ambient energy: A review. Journal of Physics: Energy, 5(2): 022001. https://doi.org/10.1088/2515-7655/acb5e6

[6] Kuaban, G.S., Czachórski, T., Gelenbe, E., et al. (2024). Energy performance of self-powered green IoT nodes. Frontiers in Energy Research, 12: 1399371. https://doi.org/10.3389/fenrg.2024.1399371

[7] Garcia-Martin, J.P., Torralba, A. (2023). Energy consumption analytical modeling of NB-IoT devices for diverse IoT applications. Computer Networks, 232: 109855. https://doi.org/10.1016/j.comnet.2023.109855

[8] Yuan, Z.B., Ge, Y.Q., Wei, J.Y., Yuan, S.H., Liu, R., Mo, X. (2023). Energy prediction for energy-harvesting wireless sensor: A systematic mapping study. Electronics, 12(20): 4304. https://doi.org/10.3390/electronics12204304

[9] Xu, M., Liu, W., Xu, J.W., et al. (2022). Recurrent neural network based link quality prediction for fluctuating low power wireless links. Sensors, 22(3): 1212. https://doi.org/10.3390/s22031212

[10] Liu, L.L., Niu, M.X., Zhang, C., Shu, J. (2022). Light

gradient boosting machine-based link quality prediction for wireless sensor networks. Wireless Communications and Mobile Computing, 2022(1): 8278087. https://doi.org/10.1155/2022/8278087

[11] Schafer, P., Leser, U. (2023). WEASEL 2.0: A random dilated dictionary transform for fast, accurate and memory constrained time series classification. Machine Learning, 112: 4763-4788. https://doi.org/10.1007/s10994-023-06395-w

[12] Cabello, N., Naghizade, E., Qi, J.Z., Kulik, L. (2024). Fast, accurate and explainable time series classification through randomization. Data Mining and Knowledge Discovery, 38: 748-811. https://doi.org/10.1007/s10618-023-00978-w

[13] Turbe, H., Bjelogrlic, M., Lovis, C., Mengaldo, G. (2023). Evaluation of post-hoc interpretability methods in time-series classification. Nature Machine Intelligence, 5: 250-260. https://doi.org/10.1038/s42256-023-00620-w

[14] Kaltsounis, A., Spiliotis, E., Assimakopoulos, V. (2023). Conditional temporal aggregation for time series forecasting using feature-based meta-learning. Algorithms, 16(4): 206. https://doi.org/10.3390/a16040206

[15] Rostami-Tabar, B., Mircetic, D. (2023). Exploring the association between time series features and forecasting by temporal aggregation using machine learning. Neurocomputing, 548: 126376. https://doi.org/10.1016/j.neucom.2023.126376

[16] Seba, A.M., Gemeda, K.A., Ramulu, P.J. (2024). Prediction and classification of IoT sensor faults using hybrid deep learning model. Discover Applied Sciences, 6: 9. https://doi.org/10.1007/s42452-024-05633-7

[17] Kumar, S.V., Mary, G.A., Mahdal, M. (2023). Integrated edge deployable fault diagnostic algorithm for the Internet of Things (IoT): A methane sensing application. Sensors, 23(14): 6266. https://doi.org/10.3390/s23146266

[18] Fan, C., Li, C., Peng, Y.F., Shen, Y.P., Cao, G.H., Li, S. (2023). Fault diagnosis of vibration sensors based on triage loss function-improved XGBoost. Electronics, 12(21): 4442. https://doi.org/10.3390/electronics12214442

[19] Pham, T.A., Lee, J.H., Park, C.S. (2022). MST-VAE: Multi-scale temporal variational autoencoder for anomaly detection in multivariate time series. Applied Sciences, 12(19): 10078. https://doi.org/10.3390/app121910078

[20] Tang, H.Y., Wang, Q.D., Jiang, G.S. (2022). Time series anomaly detection model based on multi-features. Computational Intelligence and Neuroscience, 2022(1): 2371549. https://doi.org/10.1155/2022/2371549

[21] Karimi, M., Wang, Y., Kim, H. (2022). Energy-adaptive real-time sensing for batteryless devices. In 2022 IEEE 28th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Taipei, Taiwan, pp. 205-211. https://doi.org/10.1109/RTCSA55878.2022.00028

[22] Hao, J.W., Oni, E., Kim, I.K., Ramaswamy, L. (2023). DynaES: Dynamic energy scheduling for energy harvesting environmental sensors. In 2023 IEEE International Performance, Computing, and Communications Conference (IPCCC), Anaheim, CA, USA, pp. 365-373. https://doi.org/10.1109/IPCCC59175.2023.10253869

[23] Reymond, H., Béchennec, J.L., Briday, M., Faucou, S., Puaut, I., Rohou, E. (2024). EarlyBird: Energy belongs to those who wake up early. In 2024 IEEE 30th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Sokcho, Republic of Korea, pp. 1-10. https://doi.org/10.1109/RTCSA62462.2024.00011

[24] Alwaisi, Z., Kumar, T., Harjula, E., Soderi, S. (2024). Securing constrained IoT systems: A lightweight machine learning approach for anomaly detection and prevention. Internet of Things, 28: 101398. https://doi.org/10.1016/j.iot.2024.101398

[25] Oliveira, F., Costa, D.G., Assis, F., Silva, I. (2024). Internet of Intelligent Things: A convergence of embedded systems, edge computing and machine learning. Internet of Things, 26: 101153. https://doi.org/10.1016/j.iot.2024.101153