Babylonian Journal of Internet of Things
Vol.2026, **pp**. 214–233
DOI: https://doi.org/10.58496/BJIoT/2026/015; ISSN: 3006-1083
https://mesopotamian.press/journals/index.php/BJIoT

Research Article

HARMONIC-IDS: A Hierarchical Adaptive Resolution-Aware Multimodal Framework with Bidirectional Selective State-Space Modeling for Fine-Grained IIoT Intrusion Detection

Ahmed Adil Nafea [1,*]

[1] *Department of Artificial Intelligence, College of Computer Science and IT, University of Anbar, Ramadi, Iraq*

ARTICLE INFO

Article History
Received 11 Aug 2026
Revised 09 Sep 2026
Accepted 25 Oct 2026
Published 18 Nov 2026

Keywords
Industrial Internet of Things
Intrusion detection
Hierarchical classification
Multi-resolution learning
Multimodal fusion

ABSTRACT

Accurate intrusion detection in heterogeneous Industrial Internet of Things (IIoT) environments remains challenging, particularly when attacks must be identified at multiple levels of granularity. This study proposes HARMONIC-IDS, a hierarchical multi-resolution framework that combines adaptive sensor–network fusion, bidirectional selective state-space modeling and prototype-enhanced learning. Each sample integrates ten temporal resolutions (1–10 s) within a 10-second macro-context. Separate branches encode sensor/log and network data, which are merged through scale-aware gated fusion and processed by bidirectional state-space blocks across scales. A shared representation supports binary, eight-class family and 50-class fine-grained classification simultaneously, guided by hierarchical consistency regularization and prototype learning. Evaluated on the DataSense IIoT 2025 dataset (23,939 samples) with stratified five-fold cross-validation, HARMONIC-IDS achieved accuracies of 99.36 ± 0.07%, 97.90 ± 0.32% and 86.05 ± 0.42%, with Macro-F1 scores of 99.35 ± 0.08%, 93.70 ± 1.08% and 79.75 ± 1.99%, for the three levels respectively. At the fine-grained level, it outperformed the best published benchmark on this dataset by 1.23 percentage points in accuracy and the strongest deep learning baseline by more than 9 points in weighted F1. Most remaining errors involved reconnaissance subtypes. With about 0.59 million parameters, HARMONIC-IDS offers a compact and effective solution for multi-level IIoT intrusion detection, while fine-grained discrimination of similar attacks remains an open challenge.

1. INTRODUCTION

The increasing adoption of the Industrial Internet of Things (IIoT) has transformed modern industrial environments by enabling interconnected sensors, intelligent devices, and data-driven automation [1]. Although these technologies improve operational visibility and efficiency, they also introduce significant cybersecurity challenges due to the growing diversity of communication protocols, device behaviors, and network interactions [2]. Unlike conventional information systems, IIoT environments integrate physical processes with networked infrastructure, making security incidents potentially disruptive to both digital services and industrial operations [3]. Intrusion detection systems (IDSs) therefore play an essential role in identifying malicious activities and supporting timely security responses. However, developing reliable IDS models remains challenging because industrial attacks exhibit diverse behavioral patterns, temporal characteristics, and levels of complexity [4], [5].

Recent advances in machine learning and deep learning have improved intrusion detection by enabling the automatic extraction of discriminative patterns from network observations. Nevertheless, relying exclusively on network traffic may provide an incomplete representation of attacks affecting industrial cyber–physical systems. Canonico et al. [5] demonstrated the value of integrating network and physical observations for threat detection, while Nie et al. [6] explored multi-view representations and multi-task learning for IoT security. These studies emphasize the importance of exploiting complementary information sources. However, integrating heterogeneous observations remains challenging when their relevance varies across attack scenarios and temporal resolutions.

Another important challenge concerns the temporal diversity of intrusion behavior. Certain attacks produce abrupt changes in traffic activity, whereas others involve repeated probing, distributed communication, or patterns that become distinguishable over longer observation windows. Consequently, a single temporal resolution may not adequately represent the characteristics of different attack categories. Recent research has investigated multi-scale representation learning to address this limitation. For example, Saidane et al. [7] proposed multi-scale attention fusion for transformer-based intrusion

*Corresponding author. Email: ahmed.a.n@uoanbar.edu.iq

detection, demonstrating the relevance of learning from complementary temporal patterns. Nevertheless, effectively organizing and integrating multiple temporal resolutions while preserving their individual contributions remains an important design consideration for IIoT intrusion detection.

Beyond feature representation, the granularity of attack identification is another critical issue. Binary classification can determine whether an observation is benign or malicious, but it provides limited information about the nature of the detected threat. Attack-family classification offers broader categorization, while fine-grained classification identifies individual attack subtypes. Uddin et al. [8] investigated hierarchical intrusion classification and highlighted the importance of considering relationships among classification levels. However, distinguishing closely related attack subtypes remains difficult, particularly when they exhibit overlapping statistical characteristics. This motivates the development of shared representations that support multiple classification levels while encouraging consistency between their predictions.

Recent developments in selective state-space models provide further opportunities for structured representation learning. The Mamba architecture introduced by Gu and Dao [9] demonstrated how input-dependent state transitions can selectively propagate information across sequences. Such mechanisms provide a useful foundation for modeling dependencies among ordered representations. In parallel, prototype-based learning has attracted attention in industrial intrusion detection. Shoukat et al. [10] explored prototype-guided reasoning to improve the representation of attack-related concepts. These developments suggest complementary approaches for modeling cross-resolution relationships and supporting the discrimination of related attack categories. However, their integration within a unified multi-resolution, multimodal, and hierarchical IIoT intrusion classification framework requires further investigation.

Motivated by these challenges, this study proposes HARMONIC-IDS, a Hierarchical Adaptive Resolution-Aware Multimodal Framework with Bidirectional Selective State-Space Modeling for Fine-Grained IIoT Intrusion Detection. The framework constructs unified 10-second macro-contexts from observations obtained at ten temporal resolutions (1–10 s). Sensor/log and network features are processed through dedicated encoding branches and combined using a scale-aware gated fusion mechanism. The resulting resolution-specific representations are modeled using stacked bidirectional selective state-space blocks to capture dependencies across shorter and longer observation scales. A shared representation subsequently supports binary detection, eight-class attack-family classification, and 50-class fine-grained classification. Hierarchical consistency regularization and learnable class prototypes are further incorporated to support coherent multi-level learning and improve the representation of closely related attack categories.

The main contributions of this study are summarized as follows:

1. Multi-resolution IIoT representation: A unified macro-context construction strategy organizes observations from ten temporal resolutions into structured representations, incorporating statistical feature summaries and availability masks to accommodate missing resolution-specific observations.
2. Adaptive multimodal feature fusion: Dedicated sensor/log and network encoders are integrated through a learnable scale-aware gating mechanism, enabling modality contributions to be adjusted independently across temporal resolutions.
3. Bidirectional selective state-space modeling: Stacked bidirectional selective state-space blocks are employed to model relationships among ordered resolution representations, allowing information from shorter and longer observation scales to contribute to a shared embedding.
4. Unified hierarchical intrusion classification: Three classification heads jointly address binary, eight-class attack-family, and 50-class fine-grained intrusion identification, while a hierarchical consistency objective encourages agreement across classification levels.
5. Prototype-enhanced fine-grained learning: Fifty learnable class prototypes are incorporated through temperature-scaled embedding similarity to provide additional supervision for fine-grained attack discrimination.

The remainder of this paper is organized as follows. Section 2 presents related studies. Section 3 describes the dataset, preprocessing, multi-resolution representations, HARMONIC-IDS architecture, and evaluation protocol. Section 4 presents results and discussion, including limitations. Section 5 concludes the study and outlines future research directions.

2. RELATED WORK

DataSense was released by Firouzi et al. [11] as an IIoT benchmark that records network traffic and sensor telemetry side by side, covering 50 attacks grouped into seven categories. Alongside the dataset, the authors tested 22 ML and DL models at three levels of granularity. Binary detection turned out to be relatively easy, but performance fell as the number of classes grew. Random Forest held up best at the family and fine-grained levels, whereas most deep models struggled once the task reached 50 classes. Obeten and Aliu [12] later asked a different question: how much does each modality actually contribute. Their answer was that network features do most of the work. Adding telemetry improved results by no more than 0.6 points,

and it only mattered noticeably for man-in-the-middle attacks. It is worth noting, though, that both studies worked with features from a single time window and combined modalities by simple concatenation.

Outside DataSense, a few studies have tried to combine physical and network data more deliberately. Wang et al. [13] trained separate autoencoders on sensor readings and traffic features in AD-FIT, then merged their reconstruction errors in a global autoencoder; on ToN_IoT, the fused model beat either modality alone. Pinto et al. [14] took a similar route on SWaT, pairing VAE-LSTM encoders with dual cross-attention and an adversarial discriminator, and detected 24 of 26 attack scenarios. Zhou et al. [15] applied the same idea within network data, using cross-attention to link raw byte streams with parsed ICS protocol fields. What these works have in common is that they target binary anomaly detection and look at the data through one temporal window.

Multi-scale ideas have appeared in intrusion detection, but mostly in feature space. Zhong et al. [16], for example, built a topological representation of traffic from several Mapper lenses and found that it helped with minority classes. State-space models are a more recent arrival. Sun et al. [17] pretrained a Mamba backbone on raw IIoT traffic and added early-exit heads, keeping F1 above 0.98 at an average latency of 1.58 ms. Ben Atitallah et al. [18] went further and embedded selective state-space blocks in a graph network with Markov message passing, reaching 99.70% accuracy on WUSTL-EHMS-2020. In both cases, the state-space model runs over bytes or graph nodes. To our knowledge, it has not yet been used to relate observations taken at different time resolutions.

Hierarchical IDSs split the problem into coarse and fine decisions. A representative example is the multi-stage system of Verkerken et al. [19], which achieved up to 96% balanced accuracy on CIC-IDS-2017 and CSE-CIC-IDS2018 and recognized 41 of 47 zero-day attacks. As with most designs of this kind, each stage is trained on its own. A mistake at the first stage therefore carries through to the next, and the model has no mechanism to keep its predictions consistent across levels.

These studies leave several gaps. Sensor–network fusion has been limited mostly to binary detection, and on DataSense it has only been tested by feature concatenation. Most models also rely on a single fixed window, state-space models have not been applied across time scales, and hierarchical detectors are still built from separate stages.

HARMONIC-IDS addresses these gaps. It encodes network and sensor data separately and merges them through a gate that can down-weight uninformative telemetry. Each sample combines ten windows (1–10 s), and bidirectional state-space blocks model how the signal changes across scales. All three classification levels share one representation, linked by a consistency loss, with prototype learning supporting rare classes. The model has about 590K parameters and is evaluated with 5-fold cross-validation using Macro-F1 and MCC.

3. METHODOLOGY

The proposed methodology follows a systematic multi-stage pipeline for hierarchical intrusion detection in heterogeneous IIoT environments. As illustrated in Figure 1, the workflow begins with the DataSense IIoT dataset and proceeds through data preprocessing, multi-resolution macro-context construction, stratified five-fold cross-validation with fold-local normalization, and the proposed HARMONIC-IDS architecture. The learned representation is then optimized using the hierarchical learning objective and training strategy, followed by model inference and final performance evaluation at the binary, attack-family, and fine-grained classification levels.

3.1 Dataset Description and Representation

This study utilizes the DataSense IIoT Dataset 2025, which was designed for attack analysis in heterogeneous IIoT environments [11]. The dataset provides synchronized observations from two complementary perspectives: network traffic and sensor/log data. Its processed component contains feature-level representations generated over multiple temporal windows, whereas the raw component contains the corresponding network packet captures and sensor records. The experiments employ the complete processed benign and attack archive file. The archive contains 20 processed files, comprising ten benign files and ten attack files corresponding to temporal windows from 1 second to 10 seconds. The processed files originally contain 94 columns. After identifying the numerical attributes required by the experimental pipeline, 71 numerical base features are retained. Across all temporal resolutions, the experiment processes 685,671 original window-based observations. The observations available at temporal resolution r are represented as presented in Equation (1):

$$D^{(r)} = \left\{\left(\mathrm{x}_i^{(r)}, y_i\right)\right\}_{i=1}^{N_r}, \qquad r \in \{1,2,\ldots,10\}. \tag{1}$$

where $D^{(r)}$ denotes the observation set at resolution r, $\mathrm{x}_i^{(r)}$ is the feature vector of observation i, y_i is its target label, N_r is the number of observations at resolution r, and r is the temporal-resolution index.

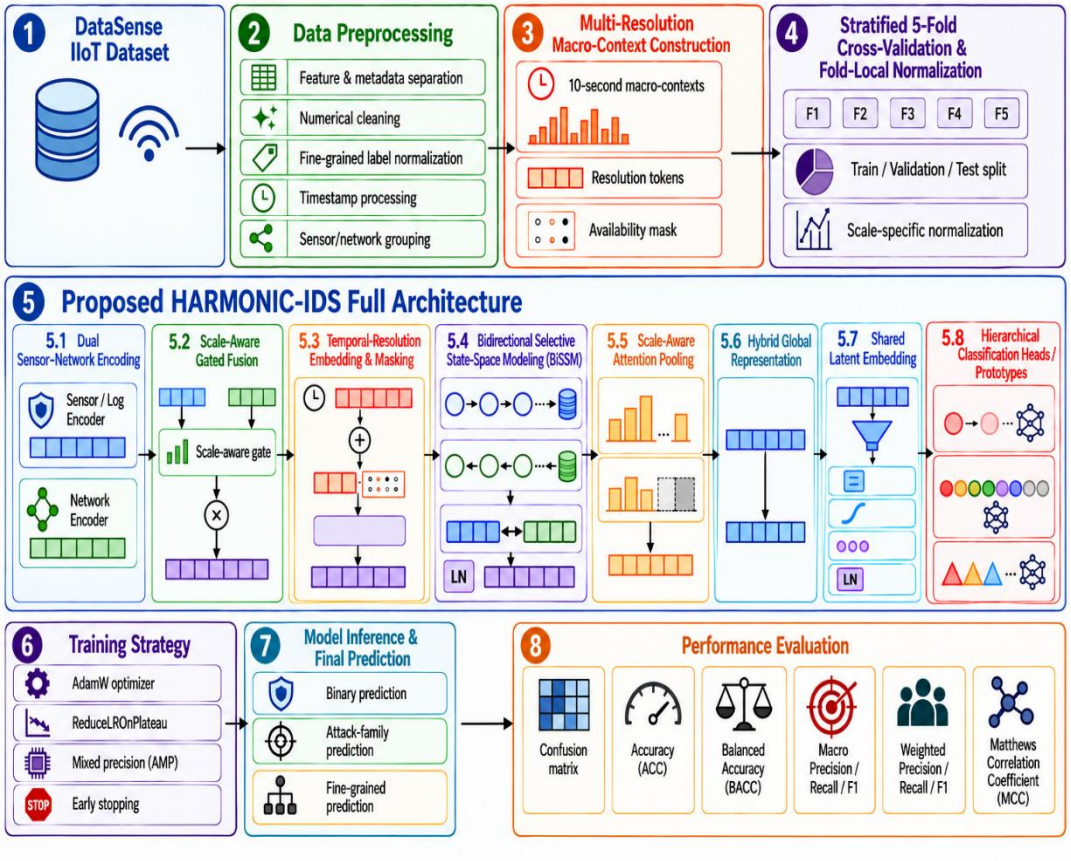


Fig. 1. The proposed framework block diagram

3.1.1 Multi-Resolution Data Organization

DataSense provides processed observations at ten temporal resolutions as presented in Equation (2):

$$\mathcal{R} = \{1,2,3,\ldots,10\}\ \text{s}. \tag{2}$$

where $\mathcal{R}$ denotes the set of temporal resolutions, and s denotes seconds.

At this stage, the ten resolution-specific observations are retained as separate temporal views. Their timestamp-based alignment into common 10-second macro-contexts is performed later in Section 3.3.

3.1.2 Hierarchical Label Representation

The target labels were organized into three hierarchical levels: binary detection, attack-family classification, and fine-grained attack classification. For sample i, the complete hierarchical label representation is defined in Equation (3):

$$\mathrm{y}_i = \left(y_i^{(B)}, y_i^{(F)}, y_i^{(C)}\right), \begin{cases} y_i^{(B)} \in \mathcal{Y}_B = \{\text{Benign}, \text{Attack}\}, & |\mathcal{Y}_B| = 2, \\ y_i^{(F)} \in \mathcal{Y}_F = \{\text{Benign}, \text{Bruteforce}, \text{DDoS}, \text{DoS}, \text{Malware}, \text{MITM}, \text{Recon}, \text{Web}\}, & |\mathcal{Y}_F| = 8, \\ y_i^{(C)} \in \mathcal{Y}_C, & |\mathcal{Y}_C| = 50. \end{cases} \tag{3}$$

where y_i is the complete hierarchical target of sample i, $y_i^{(B)}$) is the binary label, $y_i^{(F)}$ is the attack-family label, $y_i^{(C)}$ is the fine-grained label, respectively; and $\mathcal{Y}_B, \mathcal{Y}_F, \mathcal{Y}_C$ denote their corresponding class spaces. The fine-grained level comprises 49 attack classes and one benign class, with the port-specific label normalization.

3.2 Data Preprocessing

The DataSense records were preprocessed to obtain a consistent and reliable input representation before multi-resolution modeling. The preprocessing stage included feature and metadata separation, numerical data cleaning, fine-grained label normalization, timestamp processing, and sensor/network feature grouping.

3.2.1 Feature and Metadata Separation

The non-predictive fields were separated from the attributes used for model learning to prevent identifiers, labels, and timestamps from being treated as input features [20]. Each raw record was decomposed into metadata, predictive features, and target labels as presented in Equation (4):

$$d_i = [m_i,\ x_i,\ y_i], \qquad x_i \in \mathbb{R}^{71} \tag{4}$$

where d_i is the complete raw record, m_i is the metadata and timestamp fields, x_i is the retained numerical predictive feature vector, and y_i is the target labels. After excluding metadata, timestamps, labels, and raw list-valued attributes, only the 71 numerical sensor/log and network features were retained for subsequent processing.

3.2.2 Numerical Data Cleaning

The retained attributes were converted into numerical form to obtain a consistent input representation. Missing and infinite values were replaced with zero, and the resulting feature vector was stored in 32-bit floating-point format are presented in Equations (5), and (6), respectively [21]:

$$\tilde{x}_{ij} = \begin{cases} x_{ij}, & \text{if } x_{ij} \text{ is finite} \\ 0, & \text{if } x_{ij} \in \{\text{NaN}, +\infty, -\infty\} \end{cases} \tag{5}$$

$$\tilde{\mathrm{x}}_i = \text{Float32}([c(x_{i1}), c(x_{i2}), \dots, c(x_{i71})]) \in \mathbb{R}^{71} \tag{6}$$

where x_{ij} is feature j of sample i, j is the feature index, NaN denotes a missing numerical value, $\pm\infty$ denote positive and negative infinity, $\tilde{x}_i$ is the cleaned numerical feature vector of sample i, $c(.)$ is the cleaning operation, and $Float32(.)$ is the conversion to 32-bit floating-point representation.

This preprocessing step provides a uniform numerical representation while reducing memory requirements for subsequent model training.

3.2.3 Fine-Grained Label Normalization

Port-specific variants of the same attack type were normalized to avoid treating identical attacks on different ports as separate classes [8]. The fine-grained label is directly constructed as presented in Equation (7):

$$y_i^{(C)} = \begin{cases} \text{Benign}, & \text{if } label2_i = \text{Benign}, \\ label2_i \oplus \text{Norm}(label3_i), & \text{otherwise}, \end{cases} \qquad |\mathcal{Y}_C| = 50 \tag{7}$$

where $label2_i$ is the attack-family field, $label3_i$ is the raw fine-grained attack-type field, $Norm(.)$ removes only the terminal -port-number suffix, and $\oplus$ is concatenation.

This procedure produces 49 attack classes and one benign class, while preserving distinctions between attack families such as DoS and DDoS.

3.2.4 Timestamp Processing

Timestamps were standardized to ensure consistent alignment of observations across different temporal resolutions. The starting timestamps were converted to UTC, and records with invalid timestamps were excluded as presented in Equation (8):

$$t_i = \text{UTC}(timestamp_start_i), \qquad \mathcal{D}^* = \left\{(\tilde{\mathrm{x}}_i, \mathrm{y}_i, t_i) \in \mathcal{D}_{pre} \mid t_i \neq \text{NULL}\right\} \tag{8}$$

where t_i is the standardized UTC timestamp, $\text{UTC}(.)$ is conversion to Coordinated Universal Time, $\mathcal{D}_{pre}$ is the preprocessed dataset before timestamp filtering, and $\mathcal{D}^*$ is the subset containing valid timestamps.

The retained timestamps are subsequently used to align observations within the corresponding 10-second macro-contexts.

3.2.5 Sensor and Network Feature Grouping

The cleaned feature vector was divided into complementary sensor/log and network groups to enable modality-specific processing through the dedicated HARMONIC-IDS encoding branches [22]. This decomposition is defined as presented in Equation (9):

$$\tilde{\mathrm{x}}_i = \left[\mathrm{x}_i^{(S)}, \mathrm{x}_i^{(N)}\right], \qquad \mathrm{x}_i^{(S)} \in \mathbb{R}^7, \qquad \mathrm{x}_i^{(N)} \in \mathbb{R}^{64} \tag{9}$$

where $x_i^{(S)}$ and $x_i^{(N)}$ are the sensor/log and network feature vectors, respectively, while superscripts S and N identify the two modalities.

The two feature groups are subsequently aggregated independently within each temporal resolution using their mean and standard deviation before constructing the multi-resolution tokens in Section 3.3.

3.3 Multi-Resolution Macro-Context Construction

The cleaned observations from the ten temporal resolutions were aligned within common 10-second macro-contexts to preserve their temporal relationships and construct unified multi-resolution samples. Each resolution was statistically summarized and organized as a separate token, while a binary mask was used to indicate the availability of each temporal scale.

3.3.1 Ten-Second Macro-Context Generation

Each valid timestamp was assigned to a 10-second temporal bucket, and the corresponding macro-context identifier was then constructed using the device, attack scenario, and temporal bucket, as presented in Equation (10) [23]:

$$b_i = \left\lfloor \frac{t_i}{10} \right\rfloor, \quad k_i = (v_i, s_i, b_i) \tag{10}$$

where b_i is the 10-second macro-context index, and $\lfloor . \rfloor$ is the floor operation, k_i is the macro-context sample identifier, v_i is the device identifier, and s_i is the attack-scenario identifier.

3.3.2 Resolution-Specific Statistical Aggregation

Within each macro-context and temporal resolution, the numerical observations were summarized using their mean and standard deviation to capture both central tendency and variability, as presented in Equation (11)[24]:

$$\mu_{ij}^{(r)} = \frac{1}{n_{ir}} \sum_{q=1}^{n_ir} x_{ijq}^{(r)}, \quad \sigma_{ij}^{(r)} = \sqrt{\left(\frac{1}{n_{ir}} \sum_{q=1}^{n_{ir}} \left(x_{ijq}^{(r)} - \mu_{ij}^{(r)} \right)^2 \right)} \tag{11}$$

where $\mu_{ij}^{(r)}$ is the mean of feature j for sample i at resolution r, n_{ir} is the number of contributing observations; $x_{ijq}^{(r)}$ is the q^{th} contributing value, and q is a context-dependent index whose range is specified in each equation, and $\sigma_{ij}^{(r)}$ is the corresponding feature-wise standard deviation.

3.3.3 Multi-Resolution Token Construction

The mean and standard deviation of the 71 retained features were concatenated to form one resolution token is presented in Equation (12)[7]:

$$z_i^{(r)} = \left[\mu_i^{(r)}, \sigma_i^{(r)} \right] \in \mathbb{R}^{142} \tag{12}$$

where $z_i^{(r)}$ denotes the resolution token, while $\mu_i^{(r)}$ and $\sigma_i^{(r)}$ are the vectors of feature-wise means and standard deviations.

Each temporal resolution is represented by a 142-dimensional feature token. Because the 71 base features contain 7 sensor/log and 64 network features, mean-and-standard-deviation aggregation produces 14 sensor/log and 128 network components, respectively, yielding 142 features per resolution token.

3.3.4 Resolution Availability Mask

A binary availability mask was constructed to indicate which temporal resolutions are present for each macro-context, thereby preventing unavailable resolution tokens from influencing subsequent model computations is presented in Equation (13)[25]:

$$M_i = \left[m_i^{(1)}, m_i^{(2)}, \ldots, m_i^{(10)} \right] m_i^{(r)} = \begin{cases} 1 & if\ z_i^{(r)}\ is\ available \\ 0 & otherwise \end{cases} \tag{13}$$

Where M_i is the complete resolution-availability mask of sample i, $m_i^{(r)}$ is the availability indicator for resolution r of sample i.

Missing resolution tokens are subsequently prevented from influencing the model state.

3.3.5 Final Multi-Resolution Sample Construction

The ten resolution-specific tokens were arranged in ascending temporal order and combined with the availability mask and hierarchical labels to form the final multi-resolution sample, as presented in Equation (14)[26]:

$$X_i = \left[z_i^{(1)}, z_i^{(2)}, \dots, z_i^{(10)}\right] \in \mathbb{R}^{10\times 142}, S_i = \left(X_i, M_i, y_i^{(B)}, y_i^{(F)}, y_i^{(C)}\right) \quad (14)$$

where X_i is the ordered 10-resolution representation of macro-context sample i, and S_i is the complete multi-resolution sample tuple.

This procedure generated 23,939 macro-context samples. The resulting sample tuple S_i is passed directly to the stratified cross-validation and fold-local normalization stage in Section 3.4.

3.4 Stratified Cross-Validation and Fold-Local Normalization

A stratified five-fold cross-validation strategy was employed to preserve the distribution of the 50 fine-grained classes across the folds. To prevent data leakage, normalization parameters were estimated exclusively from the training samples within each fold and then applied to the corresponding validation and test subsets.

3.4.1 Stratified Five-Fold Cross-Validation

The complete set of macro-context samples was partitioned into five stratified folds according to the fine-grained class labels, ensuring approximately consistent class proportions across all folds is presented in Equation (15) [27]:

$$\mathcal{D} = \bigcup_{k=1}^{5} \mathcal{D}_{te}^{(k)}, \qquad \mathcal{D}_{trval}^{(k)} = \mathcal{D} \backslash \mathcal{D}_{te}^{(k)}, \qquad k = 1, \dots, 5 \quad (15)$$

where D is the complete set of constructed macro-context samples, $\mathcal{D}_{\text{te}}^{(k)}$ is the test subset of fold k, k is the cross-validation fold index, and $D_{\text{trainval}}^{(k)}$ denotes the train-validation portion of fold k.

This stratified partitioning preserves the fine-grained class distribution as closely as possible across the five folds.

3.4.2 Inner Training–Validation Split

The train-validation portion within each fold was further divided using stratified sampling into training and validation subsets, while preserving the fine-grained class distribution, as presented in Equation (16) [28]:

$$\mathcal{D}_{trval}^{(k)} = \mathcal{D}_{tr}^{(k)} \cup \mathcal{D}_{val}^{(k)}, \qquad \left|\mathcal{D}_{val}^{(k)}\right| = 0.12\left|\mathcal{D}_{trval}^{(k)}\right| \quad (16)$$

where $\mathcal{D}_{\text{tr}}^{(k)}$ is the training subset of fold k, respectively.

The validation subset was used for model selection, learning-rate adjustment, and early stopping.

3.4.3 Scale-Specific Fold-Local Normalization

Normalization statistics were independently estimated for each temporal resolution using only the available training samples within the corresponding fold, thereby preventing information leakage from the validation and test subsets. The fold-local mean and standard deviation are defined as presented in Equation (17):

$$\mu_j^{(r,k)} = \frac{1}{N_r^{(k)}} \sum_{i \in D_{tr}^{(k)}} z_{ij}^{(r)}, \qquad \sigma_j^{(r,k)} = \sqrt{\frac{1}{N_r^{(k)}} \sum_{i \in D_{tr}^{(k)}} \left(z_{ij}^{(r)} - \mu_j^{(r,k)}\right)^2} \quad (17)$$

where $\mu_j^{(r,k)}$ is the fold-local training mean of feature j at resolution r, $N_r^{(k)}$ is the number of available training samples at that resolution in fold k, $z_{ij}^{(r)}$ is feature j of the aggregated resolution token $z_{ij}^{(r)}$, and $\sigma_j^{(r,k)}$ is the corresponding fold-local training standard deviation.

3.4.4 Normalized Fold Input Generation

Using the statistics estimated from the corresponding training fold, each resolution-token feature was standardized, clipped to limit the influence of extreme values, and masked according to resolution availability[21]. The complete transformation is defined as presented in Equation (18):

$$\tilde{z}_{ij}^{(r,k)} = m_i^{(r)} Clip\left(\frac{z_{ij}^{(r)} - \mu_j^{(r,k)}}{\sigma_j^{(r,k)}}, -8, 8\right) \quad (18)$$

where $\tilde{z}_{ij}^{(r,k)}$ is the final normalized and masked value of feature j for sample i at resolution r in fold k, and $\text{Clip}(.,-8,8)$ limits standardized values to the interval [-8, 8].

This produces the fold-specific normalized multi-resolution tokens supplied to the proposed HARMONIC-IDS architecture in Section 3.5, while ensuring that unavailable resolutions do not contribute to subsequent computations.

3.5 Proposed HARMONIC-IDS Architecture

The proposed Hierarchical Adaptive Resolution-aware Multimodal Orchestration Network for Industrial Cybersecurity – Intrusion Detection System (HARMONIC-IDS) architecture jointly processes sensor/log and network information across ten temporal resolutions. Each resolution token is independently encoded through dual MLP branches, adaptively fused using a scale-aware gating mechanism, and enriched with temporal-resolution embeddings. The resulting sequence is modeled by stacked Bidirectional Selective State-Space (BiSSM) blocks to capture cross-resolution dependencies, followed by scale-aware attention and masked mean pooling. The final shared embedding is then used for hierarchical binary, attack-family, and fine-grained classification, while learnable class prototypes further improve discrimination among similar fine-grained IIoT attacks.

3.5.1 Dual Sensor–Network Feature Encoding

The fold-normalized token at each temporal resolution was divided into sensor/log and network streams, which were then independently projected into a common latent space using dedicated MLP encoders[5]. The fold index r for readability is omitted from the architecture equations. The modality-specific split is defined as presented in Equation (19):

$$\tilde{z}_i^{(r)} = \left[x_{i,S}^{(r)}, x_{i,N}^{(r)}\right], \qquad x_{i,S}^{(r)} \in \mathbb{R}^{14}, \qquad x_{i,N}^{(r)} \in \mathbb{R}^{128} \quad (19)$$

where $\tilde{z}_i^{(r)}$ is the normalized resolution token, while $x_{i,S}^{(r)}$ and $x_{i,N}^{(r)}$ are the sensor/log and network components, respectively.

The two streams are independently encoded as presented in Equation (20):

$$h_{i,S}^{(r)} = E_S\left(x_{i,S}^{(r)}\right), \qquad h_{i,N}^{(r)} = E_N\left(x_{i,N}^{(r)}\right), \qquad h_{i,S}^{(r)}, h_{i,N}^{(r)} \in \mathbb{R}^{128} \quad (20)$$

where $E_S(.)$ and $E_N(.)$ are the sensor/log and network MLP encoders, respectively.

3.5.2 Scale-Aware Gated Sensor–Network Fusion

An adaptive gate determines the relative contribution of sensor and network information at each resolution. The corresponding formulation is presented in Equation (21) [29]:

$$g_i^{(r)} = \sigma\left(W_g\left[h_{i,N}^{(r)}; h_{i,S}^{(r)}\right] + b_g\right) \quad (21)$$

where $g_i^{(r)}$ denotes the adaptive fusion gate; $\sigma(.)$ is the sigmoid function; W_g and b_g denote trainable weight matrices and bias vectors of the indicated layers; and $[.;.]$ denotes vector concatenation.

The fused representation is presented in Equation (22):

$$h_i^{(r)} = g_i^{(r)} \odot h_{i,N}^{(r)} + \left(1 - g_i^{(r)}\right) \odot h_{i,S}^{(r)} \quad (22)$$

where $h_i^{(r)}$ denotes the fused sensor-network representation, and $\odot$ denotes element-wise multiplication.

This mechanism enables the model to adaptively emphasize the most informative modality.

3.5.3 Temporal-Resolution Embedding

A learnable embedding was added to each fused representation to distinguish the ten temporal resolutions, followed by layer normalization and availability masking[30]. The resulting representation is defined as presented in Equation (23)

$$\tilde{u}_i^{(r)} = m_i^{(r)} LN\left(h_i^{(r)} + e_r\right) \quad (23)$$

where e_r is the learnable embedding associated with temporal resolution r, LN(.) is layer normalization, $m_i^{(r)}$ is the resolution-availability indicator, and $\tilde{u}_i^{(r)}$ is the final masked resolution-enriched representation.

This operation preserves the identity of each temporal scale while preventing unavailable resolutions from contributing to subsequent sequence modeling.

3.5.4 Bidirectional Selective State-Space Modeling

The multi-resolution sequence is processed by two stacked Bidirectional Selective State-Space Modeling (BiSSM) blocks to capture dependencies among resolutions in both directions.

3.5.4.1 Forward Selective State-Space Modeling

The forward selective state-space mechanism processes the multi-resolution sequence in ascending temporal order. At each resolution r, an input-dependent step size and a stable decay coefficient are first computed as presented in Equation (24) [31]:

$$\Delta_i^{(r)} = Softplus\left(W_\Delta \tilde{u}_i^{(r)} + b_\Delta\right) + \epsilon, A = -exp(a), \lambda_i^{(r)} = exp\left(A \odot \Delta_i^{(r)}\right) \quad (24)$$

where $\Delta_i^{(r)}$ is the input-dependent selective step size, A is the stable state-transition parameter, a is its unconstrained trainable parameterization, $\lambda_i^{(r)}$ is the selective decay coefficient, and ϵ is a small positive stability constant.

The candidate state is computed from the current resolution representation, and the recurrent state is subsequently updated using the selective decay coefficient, as presented in Equations (25) and (26):

$$c_i^{(r)} = tanh\left(W_B \tilde{u}_i^{(r)} + b_B\right) \quad (25)$$

$$s_i^{(r)} = \lambda_i^{(r)} \odot s_i^{(r-1)} + \left(1 - \lambda_i^{(r)}\right) \odot c_i^{(r)} \quad (26)$$

where $c_i^{(r)}$ is the candidate state, $s_i^{(r)}$ denotes the recurrent selective state, and $s_i^{(r-1)}$ is the state from the preceding temporal resolution.

To prevent unavailable resolutions from modifying the recurrent state, the availability mask is applied as presented in Equation (27):

$$\hat{s}_i^{(r)} = m_i^{(r)} s_i^{(r)} + \left(1 - m_i^{(r)}\right)\hat{s}_i^{(r-1)} \quad (27)$$

where $\hat{s}_i^{(r)}$ is the mask-preserved recurrent state.

Finally, the forward selective output is obtained as presented in Equation (28):

$$o_i^{(r)} = W_C \hat{s}_i^{(r)} \odot \sigma\left(W_q \tilde{u}_i^{(r)} + b_q\right) \quad (28)$$

where $o_i^{(r)}$ is the selective state-space output at temporal resolution r.

3.5.4.2 Backward Selective State-Space Modeling

The same selective state-space operation is applied to the reversed resolution sequence is presented in Equation (29) [31]:

$$H_{i,B} = SSM_B\left(\left[\tilde{u}_i^{(10)}, \ldots, \tilde{u}_i^{(1)}\right]\right) \quad (29)$$

where $H_{i,B}$ denotes the backward state-space sequence representation, and $SSM_B(.)$ denotes the backward selective state-space operator.

The resulting sequence is reversed again to restore the original scale order.

3.5.4.3 Bidirectional State Fusion

The forward and backward representations are concatenated and projected into a unified representation, as presented in Equation (30):

$$H_i^{(r)} = LN\left(\tilde{u}_i^{(r)} + W_M\left[H_{i,F}^{(r)}; H_{i,B}^{(r)}\right] + b_M\right) \quad (30)$$

where $H_i^{(r)}$ denotes the bidirectionally fused token representation, while $H_{i,F}^{(r)}$ and $H_{i,B}^{(r)}$ denote the aligned forward and backward outputs.

This allows each temporal-resolution token to incorporate information from both shorter and longer scales.

3.5.5 Scale-Aware Attention Pooling

A scale-aware attention mechanism was employed to estimate the relative importance of each available temporal resolution and assign normalized attention weights accordingly is presented in Equation (31):

$$e_i^{(r)} = w_a^T tanh\left(W_a H_i^{(r)} + b_a\right), \;\; \alpha_i^{(r)} = \frac{m_i^{(r)} exp\left(e_i^{(r)}\right)}{\sum_{q=1}^{10} m_i^{(q)} exp\left(e_i^{(q)}\right)} \tag{31}$$

where $e_i^{(r)}$ denotes the unnormalized scale-attention score, W_a denotes the trainable attention vector, and $\alpha_i^{(r)}$ denotes the normalized attention weight assigned to resolution r.

The attention-pooled representation is computed in Equation (32):

$$p_{i,A} = \sum_{r=1}^{10} \alpha_i^{(r)} H_i^{(r)} \tag{32}$$

where $p_{i,A}$ denotes the attention-pooled multi-resolution representation.

3.5.6 Hybrid Global Representation

To preserve both discriminative and global multi-resolution information, attention pooling is combined with masked mean pooling, and the combined representation is defined as presented in Equations (33):

$$p_{i,M} = \frac{\sum_{r=1}^{10} m_i^{(r)} H_i^{(r)}}{\sum_{r=1}^{10} m_i^{(r)}}, \;\; p_i = p_{i,A} + p_{i,M} \tag{33}$$

where $p_{i,M}$ is the masked mean-pooled representation, and p_i is the hybrid pooled representation.

3.5.7 Shared Latent Embedding

The pooled representation is transformed into the final shared latent embedding as presented in Equation (34)[6]:

$$z_i = E_z(p_i) \tag{34}$$

where z_i denotes the shared latent embedding, and $E_z(.)$ denotes the shared embedding head.

The embedding is shared by all hierarchical classification heads.

3.5.8 Hierarchical Classification Heads

Three parallel heads predict the hierarchical labels from the shared representation: Binary Classification Head, Attack-Family Classification Head, and Fine-Grained Classification Head. The corresponding formulation is presented in Equation (35):

$$l_i^{(B)} = W_B z_i + b_B, \quad l_i^{(F)} = W_F z_i + b_F, \;\; l_i^{(C)} = W_C z_i + b_C \tag{35}$$

where $l_i^{(B)} \in R^2$, $l_i^{(F)} \in R^8$, and $l_i^{(B)} \in R^{50}$ are the binary, attack-family, and fine-grained classification logit vectors, respectively, W_B, W_F, and W_C their corresponding bias terms are trainable classification parameters, and b_B, b_F, and b_C are the corresponding trainable bias vectors.

The corresponding formulation is presented in Equation (36):

$$p_i^{(q)} = Softmax\left(l_i^{(q)}\right), \qquad q \in \{B, F, C\}. \tag{36}$$

where $p_i^{(q)}$ denotes the class-probability vector at the indicated classification level q, and Softmax(.) denotes the softmax transformation.

3.5.9 Fine-Class Prototype Learning

To improve separation among similar fine-grained attacks, a learnable prototype p_c is assigned to each of the 50 classes.

The sample embedding and class prototypes are normalized as presented in Equation (37) [10]:

$$\bar{z}_i = \frac{z_i}{\|z_i\|_2}, \qquad \bar{p}_c = \frac{p_c}{\|p_c\|_2}. \tag{37}$$

where $\bar{z}_i$ denotes the L2-normalized sample embedding; p_c is the learnable prototype of class c, $\bar{p}_c$ is its L2-normalized form, c is the class index, and $\|.\|_2$ denotes the Euclidean norm.

The prototype similarity logits are then calculated using in Equation (38):

$$l_{i,c}^{(P)} = \frac{\bar{z}_i^T \bar{p}_c}{\tau}, \qquad c = 1, \dots, 50, \tag{38}$$

where $l_{i,c}^{(P)}$ denotes the prototype-similarity logit for class c, τ is the prototype temperature, and superscript T denotes transpose.

The prototype representation encourages compact intra-class structures and stronger separation between confusing attacks. The hierarchical logits and prototype-similarity logits produced by the architecture are then supplied to the learning objective in Section 3.6.

3.6 Hierarchical Learning Objective

The proposed HARMONIC-IDS is optimized using a joint hierarchical objective that combines fine-grained, attack-family, binary, prototype-based, and hierarchical-consistency losses. To address class imbalance, class-specific weights are computed from the training subset as presented in Equation (39) [32]:

$$w_c = \left(\frac{N_{max}}{N_c}\right)^{\gamma} \tag{39}$$

where w_c denotes the weight assigned to class c, N_c is the number of training samples in that class; N_{max} is the largest class count, and γ controls the weighting strength.

The three hierarchical classification losses are expressed using a unified weighted cross-entropy formulation, while the prototype branch is supervised using the fine-grained labels, as presented in Equation (40):

$$\mathcal{L}_q = -\sum_{c=1}^{K_q} w_c^{(q)} y_{ic}^{(q)} log p_{ic}^{(q)}, \quad \mathcal{L}_P = -\sum_{c=1}^{50} w_c^{(C)} y_{ic}^{(C)} log\, p_{ic}^{(P)} \tag{40}$$

where $q \in \{B, F, C\}$ is the hierarchical classification level, respectively, $\mathcal{L}_q$ is the classification loss at level q, $\mathcal{L}_P$ is the prototype loss, $y_{ic}^{(q)}$ is the corresponding target indicator, and $p_{ic}^{(q)}$ and $p_{ic}^{(P)}$ are the hierarchical and prototype-derived probabilities.

To enforce consistency among the hierarchical outputs, the fine-grained probabilities are first aggregated into their corresponding families, as presented in Equation (41):

$$\tilde{p}_{if}^{(F)} = \sum_{c \in \mathcal{C}_f} p_{ic}^{(C)} \tag{41}$$

where $\tilde{p}_{if}^{(F)}$ denotes the family probability aggregated from fine-grained predictions, and $\mathcal{C}_f$ denotes the set of fine-grained classes belonging to family f.

while the family probabilities are converted into binary probabilities as presented in Equation (42):

$$\tilde{p}_i^{(B)} = \left[p_{i,benign}^{(F)}, \sum_{f=2}^{8} p_{if}^{(F)}\right] \tag{42}$$

where $\tilde{p}_i^{(B)}$ denotes the binary probability vector induced from family probabilities, and p_i, $benign^{(F)}$ is the benign-family probability.

The hierarchical consistency loss is presented in Equation (43):

$$L_H = MSE\left(p_i^{(F)}, \tilde{p}_i^{(F)}\right) + MSE\left(p_i^{(B)}, \tilde{p}_i^{(B)}\right) \tag{43}$$

where L_H denotes the hierarchical consistency loss, and $MSE(.)$ denotes mean squared error.

Finally, the overall learning objective is presented in Equation (44):

$$L_{total} = \lambda_C L_C + \lambda_F L_F + \lambda_B L_B + \lambda_P L_P + \lambda_H L_H \tag{44}$$

where L_{total} is the overall learning objective, and $\lambda_C = 1.00$, $\lambda_F = 0.35$, $\lambda_B = 0.15$, $\lambda_P = 0.15$, and $\lambda_H = 0.10$ are the weighting coefficients of its five loss components.

This joint objective simultaneously promotes fine-grained discrimination, hierarchical consistency, and prototype-based separation. The resulting total loss L_{total} is minimized using the training strategy described in Section 3.7.

3.7 Training Strategy

The proposed HARMONIC-IDS was trained independently within each cross-validation fold using AdamW optimization, adaptive learning-rate scheduling, mixed-precision computation, and early stopping. The validation 50-class Macro-F1 was used to adjust the learning rate and select the best model checkpoint. Table 1 shows the training configuration of the proposed HARMONIC-IDS.

TABLE I. TRAINING CONFIGURATION OF THE PROPOSED HARMONIC-IDS.

Parameter	Setting
Cross-validation	Stratified 5-fold
Maximum epochs	25
Batch size	512
Optimizer	AdamW
Initial learning rate	8×10^{-4}
Weight decay	1×10^{-4}
Learning-rate scheduler	ReduceLROnPlateau
Scheduler monitoring metric	Validation 50-class Macro-F1
LR reduction factor	0.5
Scheduler patience	2 epochs
Minimum learning rate	1×10^{-6}
Early-stopping patience	5 epochs
Best-model criterion	Highest validation 50-class Macro-F1
Gradient clipping	1.0
Label smoothing	0.03
Dropout	0.15
Validation fraction	0.12 of train-validation set
Mixed precision	Enabled on CUDA
Random seed	42

For each fold, the checkpoint with the highest validation 50-class Macro-F1 is retained and passed to the inference stage in Section 3.8.

3.8 Model Inference and Final Prediction

During inference, the best checkpoint from each fold was applied to the corresponding unseen test subset. The shared latent representation was passed through the three hierarchical classification heads to obtain binary, attack-family, and fine-grained predictions.

The predicted probability vector for each classification level, and the final predicted class at each level, are obtained as presented in Equation (45) and (46):

$$p_i^{(q)} = Softmax\left(l_i^{(q)}\right) \tag{45}$$

$$\hat{y}_i^{(q)} = \underset{c}{argmax}\ p_{i,c}^{(q)} \tag{46}$$

where $\hat{y}_i^{(q)}$ denotes the predicted class at level q, and $argmax$ denotes selection of the class with maximum probability.

The corresponding formulation is presented in Equation (47):

$$\hat{y}_i^{(B)} \in \{1,2\}, \qquad \hat{y}_i^{(F)} \in \{1,\dots,8\}, \qquad \hat{y}_i^{(C)} \in \{1,\dots,50\} \tag{47}$$

The prediction confidence for the fine-grained output is defined in Equation (48):

$$Conf_i = \underset{c}{max} p_{i,c}^{(C)} \tag{48}$$

where $Conf_i$ denotes the fine-grained prediction confidence of sample i.

HARMONIC-IDS simultaneously produces binary, attack-family, and fine-grained predictions together with fine-grained confidence scores. These outputs are used directly for the performance evaluation described in Section 3.9.

3.9 Performance Evaluation

The performance of HARMONIC-IDS was evaluated on the unseen test samples of each fold at the binary, 8-class family, and 50-class fine-grained levels. A multiclass confusion matrix was first constructed to analyze the relationship between the actual and predicted classes, while the quantitative evaluation metrics and their mathematical formulations are summarized in Table 2.

For EV classes, the confusion matrix is represented in Equation (49):

$$CM = \begin{bmatrix} CM_{1,1} & CM_{1,2} & \cdots & CM_{1,EV} \\ CM_{2,1} & CM_{2,2} & \cdots & CM_{2,EV} \\ \vdots & \vdots & \ddots & \vdots \\ CM_{EV,1} & CM_{EV,2} & \cdots & CM_{EV,EV} \end{bmatrix} \tag{49}$$

where CM denotes the confusion matrix, EV is the number of evaluated classes, $CM_{a,b}$ is the entry at actual-class row a and predicted-class column b, and a and b are class indices.

The corresponding formulation is presented in Equation (50) [33],[34]:

$$CM_{a,b} = \sum_{i=1}^{N} I\left(y_i = a \wedge \hat{y}_i = b\right) \tag{50}$$

where N denotes the number of evaluated test samples, and $I(.)$ denotes the indicator function, which equals 1 when its condition is true and 0 otherwise.

The diagonal elements represent correct classifications, whereas the off-diagonal elements represent misclassifications. For fine-grained evaluation, a 50 × 50 confusion matrix was generated.

TABLE II. EVALUATION METRICS AND MATHEMATICAL FORMULATIONS

Metrics	Mathematical Formula
Accuracy	$\text{Accuracy} = \frac{\sum_{c=1}^{K} TP_c}{N}$
Precision	$P_c = \frac{TP_c}{TP_c + FP_c}$
Recall	$R_c = \frac{TP_c}{TP_c + FN_c}$
F1-score	$F1_c = \frac{2P_cR_c}{P_c + R_c}$
Macro precision	$P_{\text{Macro}} = \frac{1}{K}\sum_{c=1}^{K} P_c$
Macro recall	$R_{\text{Macro}} = \frac{1}{K}\sum_{c=1}^{K} R_c$
Macro F1-score	$F1_{\text{Macro}} = \frac{1}{K}\sum_{c=1}^{K} F1_c$
Balanced accuracy	$BA = \frac{1}{K}\sum_{c=1}^{K} R_c$

where TP_c, FP_c, and FN_c denote the true-positive, false-positive, and false-negative counts for class c, respectively, P_c, R_c, and $F1_c$ denote class-wise precision, recall, and F1-score; P_{Macro}, R_{Macro}, and $F1_{\text{Macro}}$ denote their macro-averaged values; Accuracy denotes overall classification accuracy; and BA denotes balanced accuracy. Symbols N, K, and c were defined previously.

4. RESULTS AND DISCUSSION

4.1 Overall Hierarchical Performance

Table 3 summarizes the performance of HARMONIC-IDS across three hierarchical classification levels using stratified five-fold cross-validation on 23,939 macro-context samples. All results are reported as mean ± standard deviation.

For binary intrusion detection, HARMONIC-IDS achieved an accuracy of 99.36 ± 0.07%, a Macro-F1 score of 99.35 ± 0.08%, and an MCC of 0.9870 ± 0.0015, indicating strong discrimination between benign and malicious observations. At the eight-class attack-family level, the model maintained an accuracy of 97.90 ± 0.32% and a Macro-F1 score of 93.70 ± 1.08%. The lower Macro-F1 reflects differences in recognition performance across attack families, particularly Bruteforce and MITM.

Fine-grained classification across 50 classes was more challenging, achieving an accuracy of 86.05 ± 0.42%, a Macro-F1 score of 79.75 ± 1.99%, and an MCC of 0.7910 ± 0.0062. The gap between accuracy and Macro-F1 indicates uneven performance across individual attack subtypes, especially those with similar characteristics.

Overall, the results demonstrate strong performance in binary detection and attack-family classification, while highlighting the greater difficulty of fine-grained attack recognition. These differences are further examined through class-level and misclassification analyses in the following sections.

TABLE III. FIVE-FOLD CROSS-VALIDATION PERFORMANCE OF HARMONIC-IDS AT THE THREE HIERARCHICAL LEVELS (MEAN ± STD; ALL VALUES IN % EXCEPT MCC).

Metric	Binary (2-class)	Attack family (8-class)	Fine-grained (50-class)
Accuracy	99.36 ± 0.07	97.90 ± 0.32	86.05 ± 0.42
Balanced accuracy	99.28 ± 0.09	93.88 ± 0.78	79.61 ± 1.89
Macro precision	99.42 ± 0.06	93.66 ± 1.50	80.72 ± 1.91
Macro recall	99.28 ± 0.09	93.88 ± 0.78	79.61 ± 1.89
Macro F1-score	99.35 ± 0.08	93.70 ± 1.08	79.75 ± 1.99
Weighted F1-score	99.36 ± 0.07	97.90 ± 0.30	85.88 ± 0.39
MCC	0.9870 ± 0.0015	0.9665 ± 0.0050	0.7910 ± 0.0062

4.2 Fold-Wise Stability

Table 4 presents the fold-wise performance of HARMONIC-IDS to assess its stability across different data partitions. Binary classification accuracy remained highly consistent, ranging from 99.27% to 99.46%, while attack-family accuracy varied between 97.45% and 98.22%. Similarly, fine-grained accuracy ranged from 85.34% to 86.47%, indicating relatively limited variation across the five folds.

In contrast, fine-grained Macro-F1 exhibited greater variability, ranging from 77.04% in Fold 3 to 81.45% in Fold 1, with a mean of 79.75 ± 1.99%. This variation may be partly associated with the limited number of samples in certain attack classes and differences in class-level recognition across test partitions. Overall, the results indicate consistent classification accuracy across the five folds, although fine-grained performance remains more sensitive to class-level differences.

TABLE IV. PER-FOLD TEST PERFORMANCE OF HARMONIC-IDS.

Fold	Binary Acc. (%)	Family Acc. (%)	Family Macro-F1 (%)	Fine Acc. (%)	Fine Macro-F1 (%)	Fine MCC	Best epoch
1	99.31	97.83	93.56	86.15	81.45	0.7926	25
2	99.39	97.81	93.44	86.11	78.24	0.7917	22
3	99.37	97.45	92.22	85.34	77.04	0.7805	21
4	99.46	98.22	94.10	86.47	81.08	0.7972	22
5	99.27	98.18	95.19	86.19	80.97	0.7929	24
Mean ± std	99.36 ± 0.07	97.90 ± 0.32	93.70 ± 1.08	86.05 ± 0.42	79.75 ± 1.99	0.7910 ± 0.0062	22.8 ± 1.6

4.3 Binary and Attack-Family Detection

Pooling the five test folds, the binary head misclassifies 153 of 23,939 macro-contexts: 20 benign samples are flagged as attacks and 133 attack samples are missed. This corresponds to a false-positive rate of 0.15% and an attack recall of 98.70%. The low false-positive rate indicates that the proposed framework generates relatively few false alarms under the evaluated conditions. Nevertheless, the 133 missed attack observations emphasize the importance of considering both false-positive and false-negative rates when assessing the operational reliability of IIoT IDSs.

Table 5 and Figure 2 report the attack-family results. Six of the eight families exceed an F1-score of 0.93, and Recon, DDoS, and DoS all exceed 0.95. The largest single source of family-level error is the DDoS/DoS pair, which accounts for 95 of the 503 family errors (18.9%); these two families share the same flooding mechanisms and differ mainly in the number of sources. Bruteforce is the weakest family (F1 = 0.803), which is expected given that it is the smallest (194 samples) and its login traffic resembles the interception traffic of MITM (13 samples confused). The most security-relevant errors are the 59 Recon and 26 MITM samples classified as benign, since both are low-rate attacks that blend into normal traffic.

TABLE V. ATTACK-FAMILY (8-CLASS) PERFORMANCE POOLED OVER THE FIVE TEST FOLDS.

Family	Support	Precision	Recall	F1-score
Benign	13,680	0.9916	0.9977	0.9947
Bruteforce	194	0.7778	0.8299	0.8030
DDoS	2,071	0.9822	0.9599	0.9709
DoS	2,132	0.9627	0.9554	0.9590
Malware	794	0.9448	0.9484	0.9466
MITM	898	0.9062	0.9143	0.9102
Recon	3,885	0.9794	0.9678	0.9736
Web	285	0.9336	0.9368	0.9352

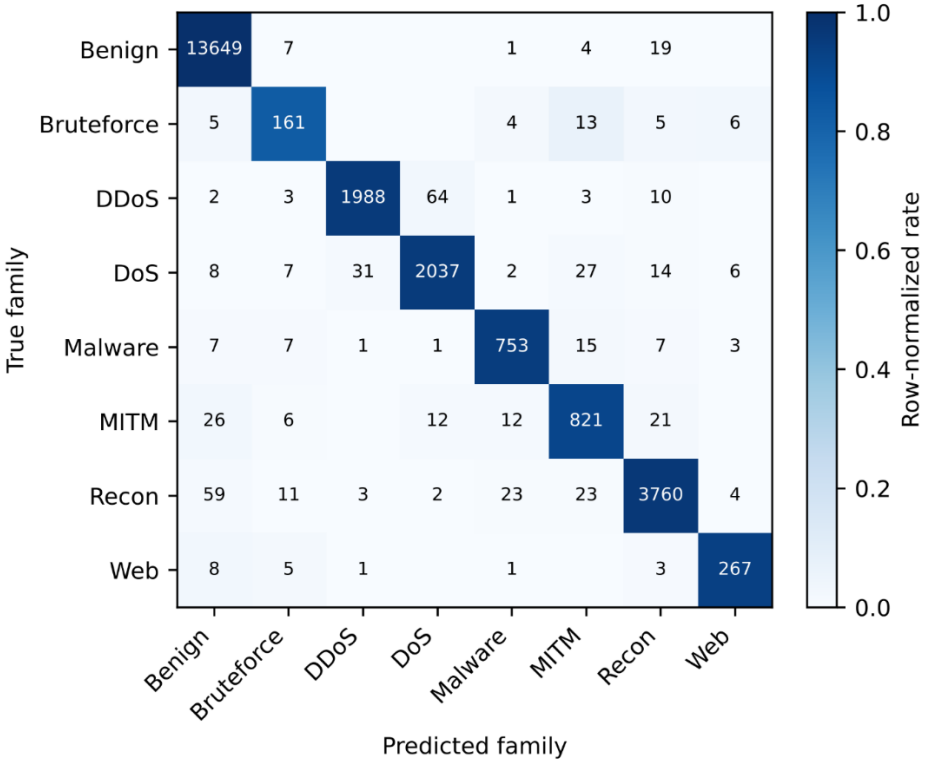


Fig. 2. Attack-family confusion matrix pooled over the five test folds. Cell values are sample counts; color denotes the row-normalized rate.

4.4 Fine-Grained Classification

Table 6 and Figure 3 present the fine-grained classification performance across 50 classes, with detailed class-level results provided in Table S1. Performance varied considerably across attack categories, with 31 classes achieving an F1-score of at least 0.90 and six additional classes scoring between 0.80 and 0.90. The DDoS and DoS families demonstrated strong fine-grained recognition, achieving exact-class recall values of 94.74% and 94.42%, respectively. Similarly, MITM and Web attacks achieved 91.31% and 91.23%.

TABLE VI. FINE-GRAINED (50-CLASS) CLASSIFICATION PERFORMANCE GROUPED BY ATTACK FAMILY, AGGREGATED ACROSS FIVE TEST FOLDS.

Family	Classes	Support	Fine-grained accuracy (%)	Mean class F1
Benign	1	13,680	99.71	0.995
Bruteforce	2	194	74.23	0.758
DDoS	14	2,071	94.74	0.951
DoS	14	2,132	94.42	0.920
Malware	2	794	69.14	0.692
MITM	3	898	91.31	0.926
Recon	9	3,885	31.17	0.336
Web	5	285	91.23	0.853

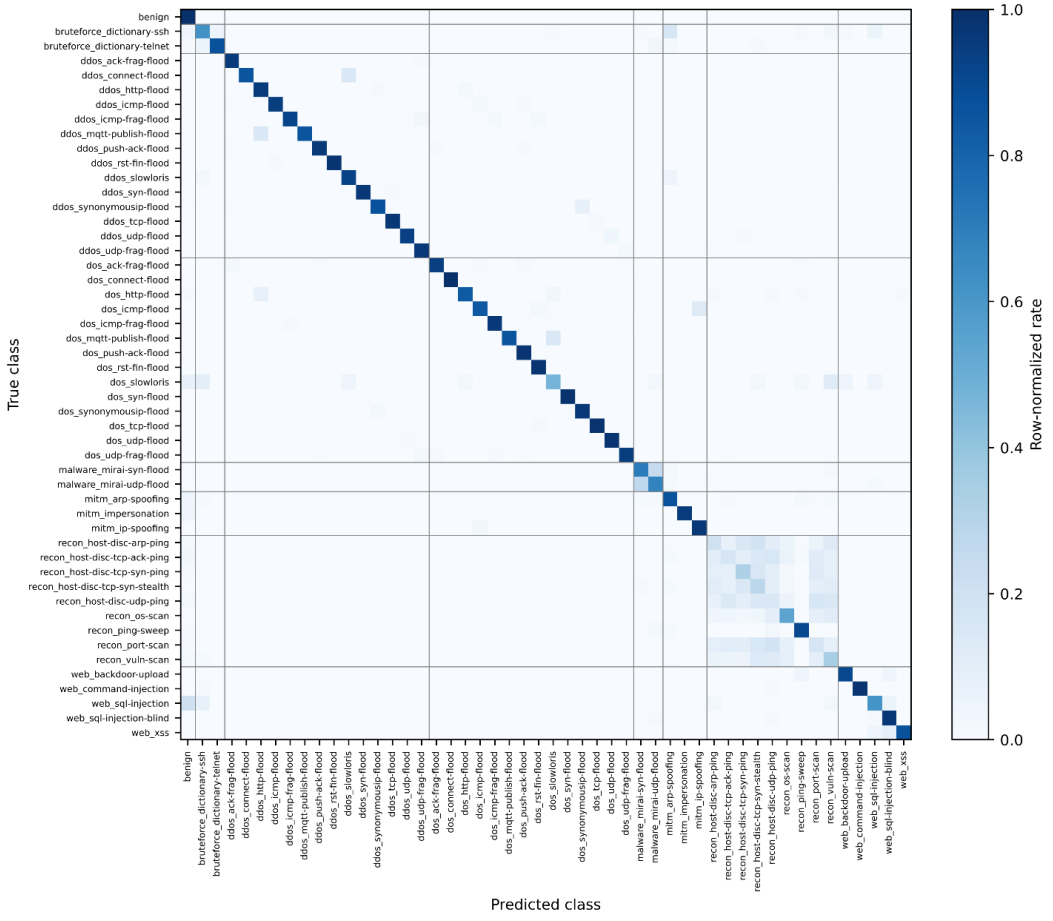


Fig. 3. Row-normalized confusion matrix for 50-class classification, aggregated across five test folds. Gray lines separate the eight attack families.

In contrast, Malware and Reconnaissance exhibited greater difficulty at the subtype level. Although Malware achieved a family-level F1-score of 0.947, its two Mirai variants (SYN flood and UDP flood) obtained fine-grained F1-scores of 0.689 and 0.695, respectively. The largest performance gap occurred in Reconnaissance, where family-level recall reached 96.78%, while exact-class recall declined to 31.17%. Excluding Reconnaissance, fine-grained accuracy increased to 96.68% across the remaining 20,054 samples. These findings indicate that the main classification challenges are concentrated in distinguishing closely related attack subtypes rather than identifying their broader attack families.

4.5 Error Analysis

Fine-grained classification errors were primarily concentrated in the Reconnaissance family, accounting for 2,674 of the 3,339 errors (80.1%). Among these, 2,551 involved confusion between Reconnaissance subtypes. Seven closely related classes, including host discovery, port scanning, and vulnerability scanning, achieved only 23.0% accuracy and a mean F1-score of 0.229. In contrast, the remaining 43 classes achieved a mean F1-score of 0.897.

These difficulties may be associated with the statistical aggregation of features within 10-second macro-contexts, which could obscure discriminative characteristics of similar attack patterns. However, further experiments are needed to confirm this explanation. Additional errors occurred between the Mirai SYN and UDP flood variants, with 204 mutual misclassifications, and between corresponding DDoS and DoS variants, with 71 cross-family errors.

Furthermore, four attack classes contained only seven samples each, contributing to variability in class-level performance. This limited support may partly explain the greater variation in fine-grained Macro-F1 compared with overall accuracy across the five folds.

4.6 Hierarchical Consistency and Prediction Confidence

HARMONIC-IDS demonstrated strong agreement across its three classification levels, with 99.22% of test samples producing hierarchically consistent predictions. Agreement between the fine-grained and attack-family heads reached 99.34%, while the binary and attack-family heads agreed in 99.87% of cases.

Prediction confidence was also closely associated with classification correctness. Correct fine-grained predictions had a mean confidence of 0.904, compared with 0.369 for incorrect predictions. As shown in Table 7, applying higher confidence thresholds improved classification accuracy among retained samples, although this reduced prediction coverage. These findings suggest that confidence scores could help identify uncertain fine-grained predictions and support alternative decisions at the attack-family level. However, confidence thresholds and the proposed hierarchical fallback strategy require further validation before practical deployment.

TABLE VII. FINE-GRAINED CLASSIFICATION ACCURACY AND PREDICTION COVERAGE AT DIFFERENT CONFIDENCE THRESHOLDS, AGGREGATED ACROSS FIVE TEST FOLDS.

Confidence threshold	Coverage (%)	Fine-grained accuracy (%)
None	100.0	86.05
≥ 0.5	86.2	96.54
≥ 0.7	83.0	98.13
≥ 0.9	75.0	99.39

4.7 Training Behavior and Model Complexity

Figure 4 shows the learning curves of the five folds. The loss values refer to the total composite objective with label smoothing, which explains why they remain above 1.0. Training and validation losses dropped sharply in the first five epochs and behaved consistently across folds. After about epoch 12, validation loss levelled off at 1.35–1.40 while training loss kept falling, indicating mild overfitting. Validation accuracy stabilized at 85–86% after epoch 10, whereas Macro-F1 remained noisier (0.70–0.82), reflecting the few validation samples in some fine-grained classes. The best checkpoints were reached between epochs 21 and 25.

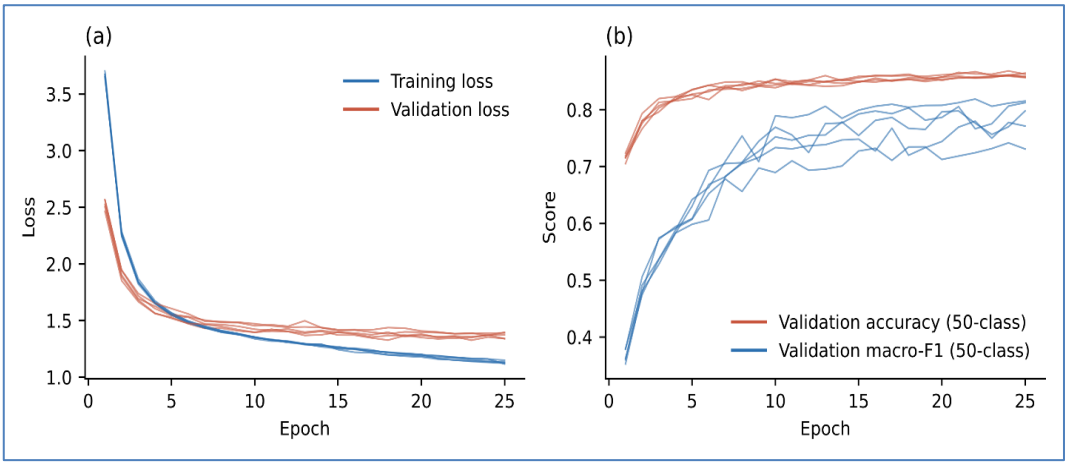


Fig. 4. Learning curves of the five folds: (a) training and validation loss; (b) validation 50-class accuracy and macro F1-score.

4.8 Comparison with the DataSense Benchmark

Table 8 compares HARMONIC-IDS with the dataset paper by Firouzi et al. [11], which benchmarked 22 ML and DL models on DataSense IIoT 2025. Since no single model in [11] leads at all levels, each level is compared with its best reported model: RNN for binary and Random Forest for the 8-class and 50-class tasks. Because [11] reports weighted metrics only, the comparison uses the weighted F1 of HARMONIC-IDS, and its Macro-F1 is reported separately in Table 8.

TABLE VIII. COMPARISON WITH THE BEST RESULTS REPORTED IN [11] (ORIGINAL FEATURE SET, %).

Level	Model in [11]	Acc. [11]	W-F1 [11]	HARMONIC-IDS Acc	HARMONIC-IDS W-F1	ΔAcc	ΔW-F1
Binary	RNN	98.97	98.82	99.36 ± 0.07	99.36 ± 0.07	+0.39	+0.54
8-class	Random Forest	97.81	97.80	97.90 ± 0.32	97.90 ± 0.30	+0.09	+0.10
50-class	Random Forest	84.82	84.39	86.05 ± 0.42	85.88 ± 0.39	+1.23	+1.49

Values for [11] are from a single stratified 80/20 split with weighted metrics; HARMONIC-IDS values are 5-fold means, so the comparison is a benchmark reference rather than a paired test.

HARMONIC-IDS exceeds the best binary result in [11] (RNN) by 0.39 points in accuracy. At the family level, it is comparable to Random Forest, since the +0.09 point difference is below the fold standard deviation (0.32). The largest gain is at the fine-grained level, where it exceeds Random Forest by 1.23 points in accuracy and 1.49 points in weighted F1, about three times the fold standard deviation. Against the strongest deep baseline in [11] (CNN, 76.75% weighted F1), the improvement exceeds 9 points, making HARMONIC-IDS the only deep model in this comparison to surpass the tree-based baseline at 50 classes. We attribute this to its multi-resolution design, which combines all ten temporal windows rather than relying on a single one.

4.9 Limitations and Future Directions

Several limitations remain. First, closely related attack variants, particularly reconnaissance attacks, remain difficult to separate at the fine-grained level. Second, stratified cross-validation may place temporally adjacent samples in both training and test folds, so device- or session-grouped evaluation is needed. Third, rare classes yield less stable per-class estimates. Fourth, the comparison with [11] uses a different protocol and is indicative only. Finally, the study uses a single dataset without on-device measurements. Future work will address these through grouped evaluation, cross-dataset validation and edge deployment.

5. CONCLUSION

This study introduced HARMONIC-IDS, a hierarchical multi-resolution intrusion detection framework designed for heterogeneous IIoT environments. The proposed framework integrates adaptive sensor–network fusion, bidirectional selective state-space modeling, and prototype-enhanced learning to support intrusion detection at three classification levels. Experimental evaluation using stratified five-fold cross-validation demonstrated strong performance, achieving accuracies of 99.36 ± 0.07%, 97.90 ± 0.32%, and 86.05 ± 0.42% for binary, eight-class attack-family, and 50-class fine-grained classification, respectively. The corresponding Macro-F1 scores reached 99.35%, 93.70%, and 79.75%. Further analysis revealed consistent performance across folds and strong attack-family recognition, although distinguishing closely related reconnaissance and malware subtypes remained challenging. Overall, the findings highlight the potential of hierarchical multi-resolution modeling for identifying IIoT attacks at different levels of granularity. Future work will focus on improving fine-grained attack discrimination, conducting component-level ablation studies, and evaluating generalization through temporally independent and cross-scenario testing.

Acknowledgment

None.

Funding

None.

CRediT Authorship Contribution Statement

Ahmed Adil Nafea: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing – original draft, Writing – review & editing, Project administration. The author has read and approved the final manuscript.

Conflicts of Interest

The author declares no conflicts of interest.

Ethical Approval

This study did not involve human or animal subjects, and ethical approval was not required.

Data Availability Statement

The DataSense IIoT Dataset 2025 is publicly available from the Canadian Institute for Cybersecurity, University of New Brunswick, at https://www.unb.ca/cic/datasets/iiot-dataset-2025.html [11]. The code and fold assignments are available from the corresponding author upon reasonable request.

Use of Generative AI and AI-Assisted Technologies in Writing

As a non-native English speaker, I used ChatGPT and Claude to improve the language and structure of this manuscript, and ChatGPT to refine the layout of Figure 1, which I designed in PowerPoint. All ideas, methods, and results are my own, and I take full responsibility for the content.

References

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Trans. Ind. informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[2] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A systematic survey of industrial Internet of Things security: Requirements and fog computing opportunities," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 4, pp. 2489–2520, 2020.

[3] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1802–1831, 2017.

[4] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Comput. Secur.*, vol. 86, pp. 147–167, 2019.

[5] R. Canonico, G. Esposito, A. Navarro, S. Pietro Romano, G. Sperlì, and A. Vignali, "Empowered Cyber–Physical Systems security using both network and physical data," *Comput. Secur.*, vol. 152, p. 104382, 2025.

[6] F. Nie, W. Liu, G. Liu, and B. Gao, "M2VT-IDS: A multi-task multi-view learning architecture for designing IoT intrusion detection system," *Internet of Things*, vol. 25, p. 101102, 2024.

[7] S. Saidane, F. Telch, K. Shahin, and F. Granelli, "Multi-scale attention fusion for enhanced transformer models in intrusion detection systems," *Comput. Networks*, p. 111985, 2026.

[8] M. A. Uddin, S. Aryal, M. R. Bouadjenek, M. Al-Hawawreh, and M. A. Talukder, "Hierarchical classification for intrusion detection system: Effective design and empirical analysis," *Ad Hoc Networks*, p. 103982, 2025.

[9] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv Prepr. arXiv2312.00752*, 2023.

[10] S. Shoukat, T. Gao, D. Javeed, and M. Adil, "From evidence to decision: Concept–prototype reasoning for intrusion detection in SDN-based industrial networks," *Comput. Networks*, p. 112495, 2026.

[11] A. Firouzi, S. Dadkhah, S. A. Maret, and A. A. Ghorbani, "Datasense: A real-time sensor-based benchmark dataset for attack analysis in iiot with multi-objective feature selection," *Electronics*, vol. 14, no. 20, p. 4095, 2025.

[12] O. M. Obeten and O. H. Aliu, "Network Dominance and the MITM Exception: A Modality-Aware Analysis of IIoT Intrusion Detection on the DataSense Benchmark," 2026.

[13] F. Wang, L. Wang, M. Lv, H. Chen, Z. Zhang, and Y. Tao, "AD-FIT: industrial anomaly detection via fusion of IoT sensing and network traffic data," *Intell. Robot.*, vol. 6, no. 3, pp. 729–749, 2026.

[14] A. Pinto, L.-C. Herrera, Y. Donoso, and J. A. Gutierrez, "Cyber-physical anomaly detection a deep adversarial fusion of sensor and network data," *Discov. Comput.*, vol. 29, no. 1, p. 183, 2026.

[15] D. Zhou, L. Bo, S. Liu, Y. Xin, J. Cao, and X. Fu, "A Multi-View Hierarchical Intrusion Detection Model with Integrating Spatiotemporal Feature for Industrial Control System," *Tsinghua Sci. Technol.*, vol. 31, no. 3, pp. 1778–1801, 2026.

[16] M. Zhong, M. Lin, and Z. He, "Dynamic multi-scale topological representation for enhancing network intrusion detection," *Comput. Secur.*, vol. 135, p. 103516, 2023.

[17] J. Sun, P. Jin, Y. Wang, and S. Jin, "MeeDet: Efficient Malicious Traffic Detection Method via Mamba-Based Early-Exit Mechanism in IIoT Scenarios," *Electronics*, vol. 15, no. 5, p. 1017, 2026.

[18] S. Ben Atitallah, M. Driss, and W. Boulila, "IDS–GraphMamba: A Markov-Enhanced Graph Mamba Framework for Real-Time Intrusion Detection in IoMT Edge Networks," *Comput. Networks*, p. 111933, 2025.

[19] M. Verkerken *et al.*, "A novel multi-stage approach for hierarchical intrusion detection," *IEEE Trans. Netw. Serv. Manag.*, vol. 20, no. 3, pp. 3915–3929, 2023.
[20] A. A. Aburomman, M. Alakhras, and M. B. I. Reaz, "Leakage-safe IoT intrusion detection with White Shark Optimizer-based feature selection," *Array*, p. 100903, 2026.
[21] M. A. Hossain, "Deep learning-based intrusion detection for IoT networks: a scalable and efficient approach," *EURASIP J. Inf. Secur.*, vol. 2025, no. 1, p. 28, 2025.
[22] R. Canonico, G. Esposito, A. Navarro, S. Pietro Romano, G. Sperli, and A. Vignali, "An anomaly-based approach for cyber–physical threat detection using network and sensor data," *Comput. Commun.*, vol. 234, p. 108087, 2025.
[23] S. Martin-Reizabal, A. Caballero-Quiroga, B. Gil-Arroyo, N. Basurto, and R. Ruiz-Gonzalez, "Transformer-based classification of IoT network traffic with flow-to-window aggregation," *Internet of Things*, p. 101879, 2026.
[24] S. Baldoni and F. Battisti, "Histogram-based network traffic representation for anomaly detection through PCA," *Comput. Networks*, vol. 265, p. 111276, 2025.
[25] S. Qian and C. Wang, "COM: Contrastive Masked-attention model for incomplete multimodal learning," *Neural Networks*, vol. 162, pp. 443–455, 2023.
[26] C. Xi, H. Wang, and X. Wang, "A novel multi-scale network intrusion detection model with transformer," *Sci. Rep.*, vol. 14, no. 1, p. 23239, 2024.
[27] A. Al Mazroa, "FORT-IDS: a federated, optimized, robust and trustworthy intrusion detection system for IIoT security" *Sci. Rep.*, vol. 16, no. 1, p. 1483, 2025.
[28] M. A. Bilal, I. Ul Islam, S. Idrees, M. Qasim, M. J. Khan, and J. Khan, "Dataset-centric evaluation of federated intrusion detection models in IoT networks," *Sci. Rep.*, vol. 16, no. 1, p. 2683, 2026.
[29] L. Xiong, L. Wan, and J. Ren, "Two-Branch Intrusion Detection Method Based on Fusion of Deep Semantic and Statistical Features," *Comput. Mater. Contin.*, vol. 87, no. 3, 2026.
[30] F. Deng and J. Sun, "PSAF-Net: a position-aware sparse encoding and structure-aware fusion network for open-set network intrusion detection," *Comput. J.*, p. bxag046, 2026.
[31] D. Tian, C. Zhao, D. Yu, Q. Zhang, and Y. Zhang, "iMamba: An improved bidirectional selective state space model for accurate lithium-ion battery state-of-health estimation under diverse charging protocols," *J. Power Sources*, vol. 665, p. 239085, 2026.
[32] K. P. Ghosh, M. Hasan, M. T. I. Robin, M. A. Hossain, and M. S. Islam, "A novel deep learning framework with temporal attention convolutional networks for intrusion detection in IoT and IIoT networks," *Sci. Rep.*, vol. 15, no. 1, p. 44624, 2025.
[33] D. A. Q. Shakir, A. A. Nafea, S. A. G. Jassim, and A. K. Kareem, "Attention-Based CNN-BiLSTM Framework for Intrusion Detection in RPL-Based IoT Networks," in *2026 International Conference on Smart Multidomain Integrated Learning Environments (ICSMILE)*, IEEE, 2026, pp. 1–6.
[34] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.