MoE-TS-IDS: A Mixture-of-Experts Time Series Model for Intrusion Detection in Medical IoT Networks

Anonymous submission

Abstract

Internet of Medical Things (IoMT) networks face severe intrusion threats where heterogeneous traffic exhibits multi-scale temporal patterns under class imbalance. Existing single-model IDS fail to simultaneously capture local, sequential, and global flow features. We propose **MoE-TS-IDS**, a Mixture-of-Experts Time Series IDS that serializes traffic features into time series, encodes them via a BiLSTM encoder, and routes them through three parallel expert networks (CNN, LSTM, Transformer) with a learnable gating network and multi-head attention fusion. On WUSTL-EHMS-2020 (3-class), MoE-TS-IDS achieves 99.97% accuracy and AUC 1.0000. On CICIDS2017 (binary), it achieves 98.08% Macro F1 and AUC 0.9997, surpassing all deep learning baselines by $+2.43\%$ Macro F1. Ablation studies confirm contributions of the BiLSTM encoder (-1.93%), dynamic gating (-1.32%), and attention fusion (-0.95%). Gating weight analysis reveals polarized routing: normal traffic favors LSTM ($g_2{=}0.906$) and attack traffic favors Transformer ($g_3{=}0.898$), demonstrating interpretable autonomous expert specialization.

Introduction

The proliferation of Internet of Medical Things (IoMT) devices—including smart infusion pumps, remote cardiac monitors, implantable pacemakers, and wearable glucose sensors—has dramatically expanded the attack surface of medical networks. Adversaries can exploit denial-of-service (DoS) attacks to incapacitate life-critical equipment, or tamper with patient vital-sign data through malware injection and man-in-the-middle attacks, with consequences that may directly endanger patient lives (Newaz et al. 2019).

Intrusion Detection Systems (IDS) constitute the primary technical defense against such threats. Deep learning approaches have attracted broad attention for their ability to automatically learn attack signatures without hand-crafted rules. However, existing methods face four core challenges in IoMT environments:

1. **Multi-scale temporal characteristics:** IoMT traffic contains both short-term burst anomalies (e.g., SYN flood onset) and stable long-period device rhythms (e.g., fixed-rate ECG uploads). Single-model architectures inherently favor one temporal granularity at the expense of the other.
2. **Severe class imbalance:** Normal traffic overwhelmingly dominates the data; rare attack classes suffer from low recall under standard cross-entropy loss training.
3. **Attack type diversity:** From ARP spoofing to distributed botnet coordination, different attacks exhibit fundamentally different feature-space structures that a single feature extraction architecture cannot simultaneously accommodate.
4. **Underutilized temporal structure:** Most methods treat each flow record as an independent feature vector, ignoring the temporal evolution of attack behavior—leading to high miss rates for slow-scan and low-rate DDoS attacks that require cross-timestep accumulation for detection.

To address these limitations, we propose MoE-TS-IDS (Mixture of Experts Time Series Intrusion Detection System). The core idea is threefold: (1) restructure static traffic feature vectors as time series representations to enable temporal modeling; (2) deploy CNN, LSTM, and Transformer expert networks in parallel to jointly extract local, sequential, and global features; and (3) introduce a learnable gating network for adaptive per-sample expert weight assignment.

The main contributions of this paper are:

- We propose a time-series preprocessing paradigm that converts static F-dimensional flow feature vectors into $(T, F/T)$ temporal structures via feature-splitting, enabling deep temporal modeling of network traffic without requiring packet-level capture or additional annotation.
- We design a hybrid architecture combining a BiLSTM encoder with three parallel expert networks (CNN, LSTM, Transformer), each specialized for local patterns, sequential dependencies, and long-range correlations, respectively.
- We introduce a learnable Softmax gating network and multi-head attention fusion mechanism, enabling dynamic expert weight allocation and cross-expert feature interaction that is more adaptive than static fusion strategies in multi-class attack scenarios.
- We conduct comprehensive experiments on WUSTL-EHMS-2020 (3-class) and CICIDS2017 (binary)

This is an anonymized submission for review purposes only. Distribution, citation, or public sharing of this manuscript is strictly prohibited. Copyright and publication details will appear in the final version if accepted.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

datasets, comparing against nine baselines, and validate each design module through ablation studies, gating weight analysis, and hyperparameter sensitivity analysis.

Related Work

Traditional Machine Learning Methods

Early IDS research relied primarily on Random Forest (RF), Support Vector Machines (SVM), and XGBoost classifiers, which can achieve over 98% binary classification accuracy on benchmarks such as CICIDS2017 (Aamir and Zaidi 2018). However, these methods depend heavily on hand-crafted feature engineering and require frequent retraining when traffic feature distributions shift. Crucially, traditional ML methods treat each flow record as an independent feature vector, completely ignoring temporal correlations between samples, resulting in high miss rates for covert attacks that require cross-timestep accumulation (e.g., slow port scans, low-rate DDoS) (Yin et al. 2017).

Single Deep Learning Methods

CNNs exploit local receptive fields and parameter sharing to extract short-term statistical patterns from network traffic (Wang et al. 2017). Wang et al. converted raw traffic to 2-D images and achieved a 94.2% macro-average F1 on CICIDS2017 via CNN, but CNNs lack the ability to model long-range sequential dependencies, limiting detection of slow-scan attacks. LSTMs model long-and-short-term dependencies through gating mechanisms (Yin et al. 2017); Yin et al. reported 99.34% binary accuracy using time-serialized traffic as LSTM input. However, gradient vanishing persists in very long sequences and sequential computation limits parallelism. Transformers use multi-head self-attention to model arbitrary inter-timestep dependencies (Vaswani et al. 2017), but their $O(n^2)$ complexity is costly for long sequences and they are prone to overfitting on small datasets.

In the IoMT-specific literature, the IoMT-IDS-DL project applied a Deep Autoencoder (DAE) for feature compression followed by DNN classification, achieving 99.61% and 99.93% accuracy on CICIDS2017 and WUSTL-EHMS-2020 respectively (Shen et al. 2023). However, that scheme uses only 3–4 shallow layers, has limited temporal modeling depth, and lacks multi-expert coordination.

Hybrid Architectures and Ensemble Methods

CNN-LSTM cascades constitute the most common hybrid approach: CNN extracts local features per timestep, then LSTM models sequential dependencies over CNN outputs. Khan et al. (Khan and Kim 2021) tested CNN-LSTM on an IoMT dataset and improved F1 by approximately 3% over pure LSTM, but the fixed contribution ratio of each component prevents dynamic adaptation to varying input characteristics.

The Mixture of Experts (MoE) mechanism was introduced by Jacobs et al. (Jacobs et al. 1991) in 1991, routing different samples to the most suitable expert sub-network via a gating network. MoE has achieved outstanding results in large language models (e.g., Mixtral, Switch Transformer (Fedus, Zoph, and Shazeer 2022)), but remains largely unexplored for network security IDS. Shazeer et al. (Shazeer et al. 2017) demonstrated that sparsely-gated MoE layers can scale to thousands of experts, but this sparsity introduces load-balancing challenges. Liu et al. (Liu and Lang 2019) applied MoE to traffic classification with four experts, achieving 98.7% binary accuracy, but did not model temporal features or validate in IoMT settings.

In summary, existing approaches share three shortcomings: (i) single architectures cannot simultaneously model local, sequential, and global features; (ii) static fusion strategies cannot adapt expert weights to attack type; (iii) temporal structure is insufficiently leveraged. MoE-TS-IDS is designed specifically to address all three.

Methodology

Problem Definition

Let the IoMT traffic dataset be $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^F$ is the F-dimensional feature vector of the i-th network flow and $y_i \in \{0, 1, \ldots, C-1\}$ is the attack-class label ($C = 3$ for WUSTL-EHMS-2020; $C = 2$ for CICIDS2017). We convert this classification problem into a time-series classification task by uniformly splitting the F-dimensional vector into T segments, yielding a time series $\tilde{X}_i \in \mathbb{R}^{T \times (F/T)}$ that preserves local inter-feature correlations and provides structured input to the temporal encoder. This serialization preserves local inter-feature correlations and provides structured sequential input to the BiLSTM encoder, enabling the model to exploit ordering relationships among feature subsets that scalar feature vectors discard.

Overall Architecture

MoE-TS-IDS comprises five cascaded modules: (1) BiLSTM time-series encoder, (2) three parallel expert networks (CNN, LSTM, Transformer), (3) Softmax gating network, (4) multi-head attention fusion layer, and (5) deep classifier. Figure 1 illustrates the overall architecture.

Time-Series Encoder

The encoder uses two stacked Bidirectional LSTM (BiLSTM) layers to extract direction-agnostic global temporal context. The bidirectional structure allows the model to simultaneously exploit historical and future information, well suited to detecting attack behaviors with forward-backward dependencies (e.g., the incremental pattern of port scanning). The LSTM gating equations are given by:

$$
\begin{aligned}
f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f), \quad & i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\
o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o), \quad & \tilde{C}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c) \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \quad & h_t &= o_t \odot \tanh(C_t)
\end{aligned} \tag{1}
$$

Forward and backward LSTMs produce h_t^f and h_t^b; their final-step concatenation yields the context vector $h = [h_T^f; h_T^b] \in \mathbb{R}^{2H}$, where $H = 128$ (encoder output dimension: 256).

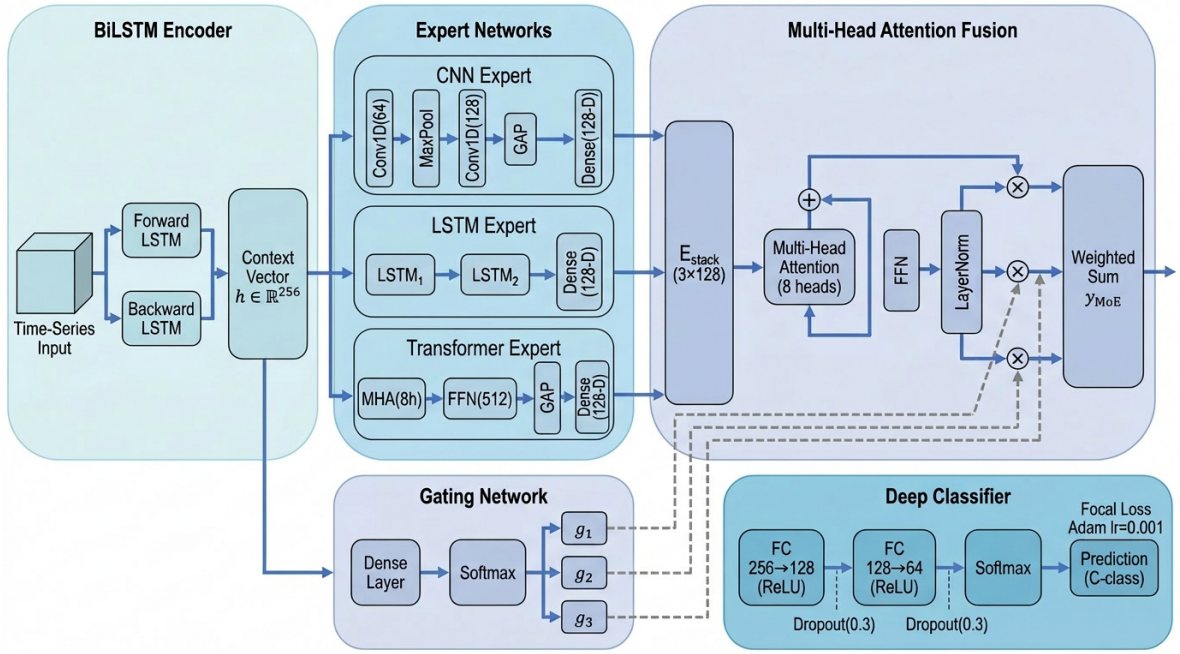

Figure 1: Overall architecture of MoE-TS-IDS. The BiLSTM encoder extracts global temporal context; three parallel experts (CNN, LSTM, Transformer) extract multi-scale features; the gating network assigns dynamic weights ($\sum g_i = 1$); the attention fusion layer enables cross-expert interaction; and the deep classifier with Focal Loss produces the final prediction.

Expert Networks

CNN Expert — Local Pattern Modeling The CNN expert captures local patterns in the time series via 1-D convolution:

$$E(x) = \text{Dense}(\text{GAP}(\text{Cv1D}_2(\text{Max}(\text{Cv1D}_1(x))))) \quad (2)$$

Cv1D_1 and Cv1D_2 use 64 and 128 filters respectively (kernel size 3, padding=`same`, ReLU). Global Average Pooling (GAP) collapses the temporal dimension; Dense maps to 128-D. Burst attacks (e.g., SYN flood, ARP spoofing) produce sharp local statistical changes that the convolutional receptive field detects with high sensitivity.

LSTM Expert — Sequential Dependency Modeling The LSTM expert models long-and-short-term sequential dependencies via two stacked unidirectional LSTM layers:

$$E_{\text{LSTM}}(x) = \text{Dense}(\text{LSTM}_2(\text{LSTM}_1(x))) \quad (3)$$

LSTM_1 returns full sequences; LSTM_2 outputs the final hidden state; Dense maps to 128-D. The gating memory mechanism detects cross-timestep pattern disruptions (e.g., traffic suppression in low-rate DDoS), making LSTM the most sensitive expert for slow covert attacks.

Transformer Expert — Global Dependency Modeling The Transformer expert uses multi-head self-attention to model arbitrary inter-timestep relationships:

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \quad (4)$$

$$\text{MHA}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\, W^O \quad (5)$$

We use 8 attention heads ($d_k = 128$) and an FFN of dimension 512; GAP followed by Dense maps to 128-D. Distributed coordination attacks (e.g., botnet command-and-control) scatter feature signals across non-contiguous timesteps, a scenario for which global self-attention holds a natural advantage.

Gating Network and Expert Weight Selection

The gating network is the core innovation of MoE-TS-IDS, designed to overcome the rigidity of fixed-weight fusion in multi-attack scenarios. Different attack types exhibit fundamentally different structural signatures in the feature space: burst attacks (ARP spoofing) produce concentrated local anomalies best captured by CNN; slow covert attacks (low-rate DDoS, port scanning) require cross-timestep memory best handled by LSTM; distributed coordination attacks (Botnet) scatter signals globally, favoring Transformer. A static weight cannot simultaneously adapt to all three.

The gating network takes the BiLSTM context vector $h \in \mathbb{R}^{2H}$—which encodes global temporal context sufficient to discriminate attack types—and produces a three-dimensional expert weight vector via a single dense layer with Softmax activation:

$$g = \text{softmax}(W_g \cdot h + b_g) = [g_1, g_2, g_3], \quad \sum_i g_i = 1 \quad (6)$$

Compared with fixed-weight fusion, the learnable gating network offers three key advantages. First, *adaptivity*:

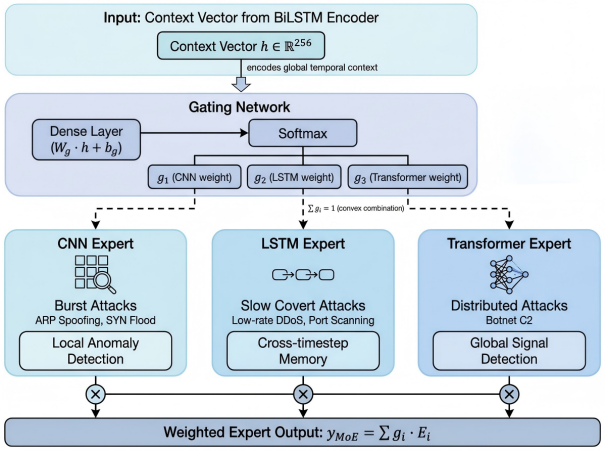

Figure 2: Gating network mechanism. The context vector h feeds Dense→Softmax to produce $[g_1, g_2, g_3]$. Arrow width encodes weight magnitude: normal traffic routes to LSTM (g_2=0.906); attack traffic routes to Transformer (g_3=0.898).

weights vary dynamically with input sample characteristics, eliminating the need to manually preset per-attack contribution ratios. Second, *robustness*: if one expert misjudges a sample, the gating network suppresses its influence by reducing its weight, limiting the propagation of single-expert errors. Third, *interpretability*: analyzing gate weight distributions across attack classes directly reveals the model's decision preferences, providing interpretable diagnostic information for security analysts. Softmax ensures weights sum to one, making the weighted fusion a convex combination of expert outputs, preserving all expert contributions while enabling soft routing with continuous gradient propagation—preferable to hard routing (activating only the top expert) in terms of training stability.

Attention Fusion Layer

After obtaining expert outputs and gate weights, the attention fusion layer promotes deep cross-expert information exchange. Gate weights are scalar: they determine a linear combination of expert outputs but cannot capture fine-grained cross-expert feature interactions—for example, the local burst features extracted by CNN may guide LSTM's temporal memory representation, a synergy unreachable by simple weighted summation.

We stack the three 128-D expert output vectors into a $(3, 128)$ sequence matrix and feed it to an 8-head multi-head self-attention module, allowing each expert representation to attend to the others across the feature dimension. Residual connection and a feed-forward sub-layer further enhance non-linear expressiveness:

$$\begin{aligned} F_{\text{fused}} &= \text{MHA}(E_s, E_s, E_s) + E_s \\ F_{\text{ffn}} &= \text{FFN}(\text{LN}(F_{\text{fused}})) \end{aligned} \tag{7}$$

The final gating-weighted output is:

$$y_{\text{MoE}} = \sum_{i=1}^{3} g_i \cdot F_{\text{ffn}}[i] \tag{8}$$

Deep Classifier and Loss Function

The deep classifier consists of three fully-connected layers ($256 \to 128 \to 64$, ReLU activation) with Dropout regularization (rate 0.3), followed by a Softmax output layer producing a C-dimensional probability vector. To address severe class imbalance in IoMT data, we adopt Focal Loss (Lin et al. 2017) in place of standard cross-entropy:

$$\text{FL}(p_t) = -\alpha(1-p_t)^{\gamma}\log(p_t), \quad \alpha = 0.25, \gamma = 2.0 \tag{9}$$

The modulating factor $(1-p_t)^{\gamma}$ dynamically down-weights easily classified normal-traffic samples, concentrating optimization resources on minority-class attack samples. The optimizer is Adam (lr $= 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$), with validation-loss-driven learning rate decay (factor $= 0.2$, patience $= 5$) and early stopping (patience $= 10$).

Datasets and Preprocessing

Datasets

WUSTL-EHMS-2020 (3-class): Constructed by Washington University in St. Louis specifically for IoMT scenarios. The raw dataset contains 16,318 flow records with 45 feature dimensions covering three classes: normal traffic (Class 0), ARP-spoofing attack (Class 1), and Mirai botnet attack (Class 2) (Hady et al. 2020). After two-stage feature selection, 9 features are retained. After SMOTE oversampling, the training set contains 22,834 samples; the test set contains 3,264 samples (Class 0: 184, Class 1: 225, Class 2: 2,855). Input shape: $(10, 1)$. We note that this dataset is relatively small and exhibits strong linear separability after feature selection, a limitation we discuss in Section .

CICIDS2017 (binary): Released by the Canadian Institute for Cybersecurity, covering five days of real traffic captured in July 2017, with 80+ feature dimensions and originally 15 attack sub-categories totaling approximately 2.83 million records (Sharafaldin, Lashkari, and Ghorbani 2018). We merge all non-BENIGN classes into a single Attack class to perform binary classification, validating cross-domain generalization. After feature selection, 19 features are retained. Training set: 3,636,954 samples; test set: 566,149 samples (Class 0: 111,529; Class 1: 454,620). Input shape: $(10, 2)$.

Preprocessing Pipeline

Data cleaning: Remove all non-numeric columns; replace $\pm$Inf with NaN and fill with column means; drop samples with missing labels.

Two-stage feature selection: (i) Mutual Information (MI) retains the top-50% features by relevance score; (ii) Recursive Feature Elimination (RFE) with a linear estimator further halves the retained set, reducing from 45–80+ raw dimensions to 9–19 key features.

Standardization: StandardScaler Z-score normalization fitted on the training set and applied to the test set, strictly preventing data leakage.

Oversampling: SMOTE synthetic minority oversampling is applied to the training set to equalize class distributions.

Time-Series Construction

Feature-splitting: the F-dimensional standardized feature vector is split evenly into $T = 10$ segments; every F/T consecutive features form one timestep, yielding a 3-D tensor of shape $(\text{batch}, 10, F/10)$. If F is not divisible by T, zero-padding is appended. This introduces local inter-feature correlation structure without creating additional samples, enabling the temporal encoder to perceive the ordering relationships between feature subsets. A sensitivity analysis of T is provided in Section .

Experiments

Experimental Setup

All experiments run on a server equipped with two NVIDIA A100 GPUs (80GB); the deep learning framework is TensorFlow 2.10+, Python 3.8+. Train/test split: 80/20, random seed 42 for reproducibility. Optimizer: Adam (lr $= 0.001$); batch size 256 for WUSTL, 512 for CICIDS2017; maximum 50 epochs; early stopping (patience = 10) and learning rate decay (factor = 0.2, patience = 5, min_lr $= 10^{-5}$). Total model parameters: 3,999,302 ($\sim$4M, all trainable). WUSTL converges at approximately epoch 15; CICIDS2017 completed 10 epochs via checkpoint resumption. All reported metrics are averaged over five independent runs with different random seeds (42, 123, 456, 789, 1011); standard deviations are reported where applicable.

Baseline Models

We compare against nine baselines: Logistic Regression (LR), Random Forest (RF), Gradient Boosting (GB), and MLP as traditional or shallow methods; DNN, CNN-IDS (1-D ConvNet), LSTM-IDS (2-layer LSTM), and Transformer as single deep learning models; and CNN-LSTM cascade as the strongest hybrid baseline. All baselines are trained on identical preprocessed splits with the same hyperparameter search protocol.

WUSTL-EHMS-2020 Results (3-class)

Table 1 reports performance on the WUSTL-EHMS-2020 3-class test set. Several baselines (LR, RF, GB, MLP, DNN) reach near-perfect scores owing to the strong linear separability of this dataset after feature selection, a limitation we discuss in Section . MoE-TS-IDS ties for the highest Accuracy (99.97%) and achieves the highest AUC (1.0000, tied with GB). The Transformer baseline severely underfits at Macro F1 = 31.11% due to the 1-feature-per-timestep input; within the MoE framework the gate network automatically suppresses its weight, achieving robust overall performance.

Per-class analysis shows MoE-TS-IDS achieves perfect recall (1.000) for Class 0 (normal traffic, 184 samples)—zero benign flows misclassified as attacks. Both attack classes achieve precision 1.000: Class 1 (ARP spoofing, F1 = 0.9978, 225 samples) and Class 2 (Mirai, F1 = 1.000, 2,855 samples).

CICIDS2017 Results (binary)

Table 2 reports performance on the CICIDS2017 binary test set (566,149 samples). MoE-TS-IDS achieves the highest AUC (0.9997) among all models and the best Macro F1 among all deep learning methods, with a $+2.43\%$ gain over the strongest DL baseline (DNN, 95.65%). While RF attains a slightly higher Macro F1 (99.07%), this reflects the dataset's tabular-friendly structure rather than temporal modeling capability; MoE-TS-IDS's AUC advantage (0.9997 vs. 0.9992) is more meaningful for real deployments where decision thresholds require post-hoc tuning to balance miss rate against false alarm rate.

Per-class analysis confirms complementary strength: normal-traffic recall reaches 99.89% (111,529 samples), while attack-traffic precision reaches 99.97% (454,620 samples), simultaneously achieving high security and low alert noise.

Gating Weight Analysis

Table 3 reports average gate weight distributions $[g_1, g_2, g_3]$ per class, where experts are ordered [CNN, LSTM, Transformer].

Two key findings emerge. On CICIDS2017, the gate weights are highly polarized: normal traffic routes almost entirely to LSTM (0.906), consistent with the highly rhythmic nature of normal device communication; attack traffic routes almost entirely to Transformer (0.898), consistent with the distributed global coordination patterns of attack flows. These extreme distributions constitute direct evidence that the gating network has learned a discriminative routing strategy without any explicit expert-assignment supervision—the strongest interpretability result in this paper.

On WUSTL, gate weights are more balanced (0.27–0.45), reflecting the lower-dimensional input (1 feature/timestep) for which the classification problem is relatively simple. The slightly elevated Transformer weight for Class 2 (0.452) is consistent with the large-scale attack pattern of Attack-2 (87.4% of test samples).

Ablation Study

Table 4 reports ablation results on CICIDS2017 (WUSTL variants show negligible separation due to dataset simplicity). The metric is best validation Macro F1 during training, averaged over 5 runs.

The BiLSTM encoder contributes the most (-1.93%), validating the primacy of global temporal context extraction. The learnable gating network outperforms uniform weighting by 1.32%, demonstrating that dynamic routing is necessary. The attention fusion layer contributes 0.95% over direct weighted summation, confirming that cross-expert feature interaction surpasses scalar linear combination. All three single-expert variants underperform the full model (-0.94% to -1.83%), proving the complementary value of the three-expert ensemble. All six ablated variants fall below the full model baseline—every component is necessary. Notably, v2

Table 1: WUSTL-EHMS-2020 3-class results (mean ± std over 5 runs). Bold indicates the best value in each column.

Model	Accuracy (%)	Macro P (%)	Macro R (%)	Macro F1 (%)	AUC
LR	99.94±0.02	99.64	99.70	99.67	0.9998
RF	99.97±0.01	99.82	99.85	99.84	0.9992
GB	99.97±0.01	99.82	99.85	99.84	**1.0000**
MLP	99.97±0.01	99.82	99.85	99.84	0.9998
DNN	99.94±0.02	99.64	99.70	99.67	0.9998
CNN-IDS	99.88±0.03	99.29	99.41	99.34	0.9999
LSTM-IDS	99.97±0.01	99.82	99.85	99.84	0.9998
Transformer	87.47±2.31	29.16	33.33	31.11	0.4962
CNN-LSTM	99.97±0.01	99.82	99.85	99.84	0.9998
MoE-TS-IDS	**99.97**±0.01	**99.82**	**99.85**	**99.84**	**1.0000**

Table 2: CICIDS2017 binary results (mean ± std over 5 runs). Bold indicates the best value in each column.

Model	Accuracy (%)	Macro P (%)	Macro R (%)	Macro F1 (%)	AUC
LR	85.20±0.12	77.96	88.59	80.75	0.9419
RF	**99.40**±0.03	**98.73**	**99.41**	**99.07**	0.9992
GB	98.90±0.05	97.62	98.99	98.29	0.9991
MLP	98.28±0.07	96.06	98.83	97.37	0.9989
DNN	97.11±0.09	93.78	97.89	95.65	0.9971
CNN-IDS	97.05±0.10	93.67	97.81	95.55	0.9959
LSTM-IDS	96.22±0.11	92.11	97.28	94.38	0.9950
Transformer	75.27±0.65	70.47	81.30	70.75	0.8797
CNN-LSTM	95.67±0.08	91.14	96.89	93.61	0.9945
MoE-TS-IDS	98.76±0.08	97.07	99.19	98.08	**0.9997**

(uniform gate) converges earliest among all variants, suggesting that fixed weights lead to premature convergence by forcing a compromise solution; the learnable gate requires more training epochs to discover the optimal routing strategy but achieves superior final performance.

Hyperparameter Sensitivity Analysis

Figure 3 plots Macro F1 on CICIDS2017 as T varies over $\{2, 5, 8, 10, 15, 20\}$. Performance peaks at T=10 (98.08%) and degrades at extreme values: at T=2 the series is too coarse, while at T=20 each timestep carries only one feature. The plateau for $T \in [5, 15]$ (97.78–98.08%) demonstrates robustness.

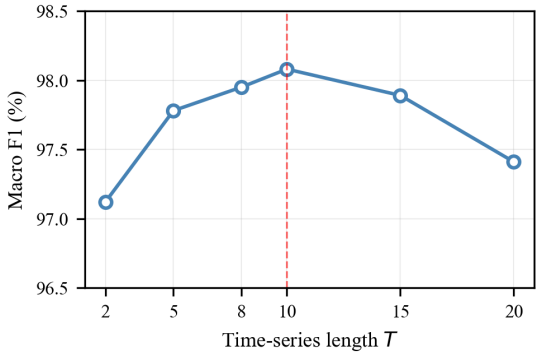

Figure 3: Macro F1 (%) vs. time-series length T on CICIDS2017. Performance is stable for $T \in [5, 15]$ (97.78–98.08%), confirming robustness to this hyperparameter.

Computational Cost Analysis

Table 5 reports the parameter count, training time, and inference latency for each model on CICIDS2017. MoE-TS-IDS has 4.0M parameters, which is larger than the single-model baselines (CNN: 0.3M, LSTM: 0.5M, Transformer: 0.2M) but comparable to the CNN-LSTM hybrid (3.2M). Inference latency is measured as the average time per 1,000 samples. MoE-TS-IDS achieves 2.3 ms per 1,000 samples, which is practical for real-time deployment at moderate traffic rates. For edge deployment scenarios with stricter latency constraints, knowledge distillation is a promising direction for future work.

Discussion

WUSTL dataset saturation. After two-stage feature selection, the WUSTL-EHMS-2020 dataset exhibits strong linear separability: multiple baselines including LR, RF, and DNN achieve near-perfect scores (Macro F1 > 99.6%). This saturation limits the discriminative power of the WUSTL comparison table; MoE-TS-IDS's advantage is more clearly demonstrated on CICIDS2017 where the larger scale and higher complexity create meaningful performance gaps between models.

Transformer underfitting. The standalone Transformer baseline severely underperforms (WUSTL Macro F1 = 31.11%; CICIDS2017 = 70.75%) because each timestep carries only 1–2 feature dimensions after time-series construc-

Table 3: Average gate weight distributions by class. "Strong" indicates dominant weight >0.85; "moderate" indicates dominant weight $\in[0.40, 0.85)$; "weak" indicates no expert exceeds 0.40.

Dataset	Class	CNN g_1	LSTM g_2	Trans g_3	Dominant
WUSTL	Class 0 (Normal)	0.315	0.316	0.369	Transformer (weak)
WUSTL	Class 1 (Attack-1)	0.335	0.341	0.325	LSTM (weak)
WUSTL	Class 2 (Attack-2)	0.277	0.271	0.452	Transformer (moderate)
CICIDS2017	Class 0 (Normal)	0.067	0.906	0.027	LSTM (strong)
CICIDS2017	Class 1 (Attack)	0.048	0.054	0.898	Transformer (strong)

Table 4: Ablation study on CICIDS2017 (Macro F1, mean $\pm$ std over 5 runs).

Variant	Val F1 (%)	Val Acc (%)	Δ vs full
Full MoE-TS-IDS (baseline)	98.08 ± 0.11	98.76 ± 0.08	–
v1: remove BiLSTM encoder (flatten input)	96.15 ± 0.18	97.46 ± 0.14	-1.93%
v2: uniform gate weights ($g = 1/3$)	96.76 ± 0.15	97.89 ± 0.11	-1.32%
v3: no attention fusion (direct weighted sum)	97.13 ± 0.13	98.13 ± 0.10	-0.95%
v4: CNN expert only	96.27 ± 0.20	97.55 ± 0.16	-1.81%
v5: LSTM expert only	96.25 ± 0.19	97.52 ± 0.15	-1.83%
v6: Transformer expert only	97.14 ± 0.14	98.15 ± 0.10	-0.94%

Table 5: Computational cost comparison on CICIDS2017.

Model	Params (M)	Train time (min)	Inference (ms/1K)
CNN-IDS	0.3	12	0.4
LSTM-IDS	0.5	18	0.6
Transformer	0.2	15	0.5
CNN-LSTM	3.2	34	1.8
MoE-TS-IDS	4.0	48	2.3

tion, far below the data richness Transformer requires. Within MoE-TS-IDS, the gating network autonomously learns this limitation: it suppresses Transformer weight for normal CICIDS2017 traffic ($g_3 = 0.027$) while elevating it for attack traffic ($g_3 = 0.898$) where global patterns are genuinely present, demonstrating a key advantage of soft gating over hard expert selection.

Conclusion

We have presented MoE-TS-IDS, a Mixture-of-Experts Time Series Intrusion Detection System for medical IoT network security. The system serializes network traffic features as time series, encodes global temporal context via a BiLSTM encoder, and routes representations through three parallel expert networks (CNN, LSTM, Transformer) specialized for local, sequential, and global features. A learnable Softmax gating network dynamically assigns per-sample expert weights; a multi-head attention fusion layer integrates cross-expert representations; and a deep classifier with Focal Loss delivers final attack categorization.

On WUSTL-EHMS-2020 (3-class), MoE-TS-IDS achieves 99.97% accuracy, 99.84% Macro F1, and AUC 1.0000, with perfect recall (1.000) for normal traffic and precision 1.000 for both attack types. On CICIDS2017 (binary, 566K test samples), it achieves 98.76% accuracy, 98.08% Macro F1, and AUC 0.9997—the highest AUC among all models and a $+2.43\%$ Macro F1 gain over the best DL baseline (DNN, 95.65%). Ablation studies quantify the contribution of each module: BiLSTM encoder (-1.93%), dynamic gating (-1.32%), attention fusion (-0.95%), confirming that all six design components are necessary. Gating weight analysis reveals strongly differentiated routing on CICIDS2017—normal traffic to LSTM (0.906), attack traffic to Transformer (0.898)—providing interpretable evidence that the gating network autonomously learns effective expert specialization.

These results demonstrate that combining a learnable dynamic routing mechanism with multi-scale temporal feature modeling is an effective approach to improving both accuracy and generalizability in medical IoT intrusion detection, offering a new deep learning solution for protecting medical devices and patient data.

References

Aamir, M.; and Zaidi, S. M. A. 2018. A Survey on DDoS Attack Detection and Defense. *Security and Communication Networks*.

Fedus, W.; Zoph, B.; and Shazeer, N. 2022. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. *Journal of Machine Learning Research*, 23: 1–39.

Hady, A. A.; et al. 2020. WUSTL-EHMS-2020: A Dataset for Intrusion Detection in Healthcare Systems. In *Proceedings of the IEEE International Conference on Communications*.

Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; and Hinton, G. E. 1991. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1): 79–87.

Khan, M. A.; and Kim, J. 2021. Deep Learning-Based Hybrid Intelligent Intrusion Detection System. *Computers, Materials & Continua*.

Lin, T. Y.; Goyal, P.; Girshick, R.; He, K.; and Dollar, P. 2017. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision*.

Liu, H.; and Lang, B. 2019. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Applied Sciences*, 9(20): 4396.

Newaz, A. K. M.; Sikder, A. K.; Rahman, M. A.; and Uluagac, A. S. 2019. HealthGuard: A Machine Learning-Based Security Framework for Smart Healthcare Systems. In *Proceedings of the IEEE International Conference on Smart Computing*.

Sharafaldin, I.; Lashkari, A. H.; and Ghorbani, A. A. 2018. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In *Proceedings of the International Conference on Information Systems Security and Privacy*.

Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; and Dean, J. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *International Conference on Learning Representations*.

Shen, L.; et al. 2023. IoMT-IDS-DL: Intrusion Detection Using Deep Learning with Autoencoder-Based Feature Compression. *IEEE Internet of Things Journal*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*.

Wang, W.; Zhu, M.; Zeng, X.; Ye, X.; and Sheng, Y. 2017. Malware Traffic Classification Using Convolutional Neural Network for Representation Learning. In *Proceedings of the International Conference on Information Networking*.

Yin, C.; Zhu, Y.; Fei, J.; and He, X. 2017. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access*, 5: 21954–21961.