A Hybrid CNN-Transformer Approach for Predicting Attack Severity in Electronic Health Monitoring Systems to Strengthen Cybersecurity

C. A. Bindyashree
School of Computer Science and Engineering, REVA University, Bengaluru, Sathanur, Karnataka, India
bindya.24@gmail.com

Muzamil Basha Syed
School of Computer Science and Engineering, REVA University, Bengaluru, Sathanur, Karnataka, India
muzamilbasha.s@reva.edu.in (corresponding author)

Received: 3 March 2025 | Revised: 16 April 2025, 17 May 2025, 1 June 2025, and 6 June 2025 | Accepted: 9 June 2025

 DOI: https://doi.org/10.48084/etasr.10784

ABSTRACT

Electronic Health Monitoring Systems (EHMS) have revolutionized patient care through continuous, connected monitoring. However, their pervasive connectivity exposes them to evolving cyber threats. In this context, for a resilient, real-time Intrusion Detection System (IDS), we propose a novel hybrid Convolutional Neural Network–Transformer (CNN–Transformer) architecture that integrates the spatial feature extraction and long-range sequence modelling functionality. The framework is trained on the publicly available WUSTL-EHMS-2020 network traffic dataset. The model features a dual-output head that simultaneously: (i) classifies attack types and (ii) predicts attack severity on a continuous scale. To address the dataset's severe class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) is employed. Experimental results show the model achieves a classification accuracy of 83.33%, macro F1-score of 0.93, and Receiver Operating Characteristic Area Under the Curve (ROC-AUC) of 0.96, and severity regression achieves a Mean Absolute Error (MAE) of 0.3337 and an R^2 0.89. Shapley Additive Explanations (SHAP) provide model interpretability, revealing packet length and inter-arrival time as key predictive features. The proposed IDS outperforms state-of-the-art CNN, Long Short-Term Memory (LSTM), and ensemble baselines in the precision on minority classes. It is also computationally efficient, requiring only a single NVIDIA RTX 3080 Graphics Processing Unit (GPU) with <2 GB VRAM per batch, and delivers inference latency below 150 ms, meeting clinical real-time requirements. These findings make the hybrid CNN–Transformer a viable and deployment-ready approach to protect EHMS against cyber-attacks, in a scalable and explainable manner.

Keywords-cybersecurity; Electronic Health Monitoring Systems (EHMS); hybrid Convolutional Neural Network (CNN)-Transformer Model; intrusion detection system; severity prediction; explainable Artificial Intelligence (AI); Synthetic Minority Over-sampling Technique (SMOTE)

I. INTRODUCTION

The development of Electronic Health Monitoring Systems (EHMS) has significantly changed clinical decision-making thanks to the possibility of continuous biometric telemetry. However, this widespread digital integration has concurrently expanded the attack surface, exposing life-critical systems to threats such as data breaches and malicious manipulation. Consequently, there is a growing demand for real-time Intrusion Detection Systems (IDSs) that are not only effective but also lightweight enough to be deployed in mission-critical, resource-constrained healthcare environments.

While signature-based detection remains inadequate against zero-day attacks, classical Machine Learning (ML) methods such as Random Forest (RF), Support Vector Machines (SVM), and Naïve Bayes (NB) have previously been used for network intrusion detection in early healthcare datasets like KDD-99. However, these approaches struggle with scalability in high-dimensional, imbalanced, and real-time traffic scenarios.

Recent developments in IDS for Internet of Things (IoT) and healthcare environments underscore the growing relevance of ML and Deep Learning (DL) in addressing the expanding cyber threat landscape. Traditional ML models continue to play

roles in Distributed Denial of Service (DDoS) mitigation, spoofing detection, and anomaly detection in IoT networks [1, 2], but they are constrained by the evolving nature of attacks and challenges associated with large, labeled datasets [3, 4].

To address complex threats such as spoofing and location-based attacks, deep neural networks like Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks have been utilized for temporal pattern recognition [5–8]. These models have demonstrated strong performance in DDoS detection and time-of-arrival spoofing identification. However, standalone DL models often entail high computational costs and are ill-suited for latency-sensitive applications [9, 10].

Consequently, there is a growing interest in hybrid and lightweight approaches. Enhanced preprocessing methods, such as Synthetic Minority Oversampling Technique (SMOTE), feature selection, and correlation-based refinement, contribute to both scalability and detection efficacy [11, 12]. While optimized LSTMs and Recurrent Neural Networks (RNNs) are frequently effective in modeling sequential biomedical and telemetry data [13, 14], they remain susceptible to overfitting, training instability, and limited interpretability [15, 16].

Security schemes based on tree-based ensembles and boosting algorithms have also attracted attention due to their resistance against noisy and complex traffic patterns [17, 18]. However, these methods are insufficient to model long-term dependencies and do not provide alert prioritization with high granularity. The emergence of Transformer architectures, powered by self-attention mechanisms, has significantly improved malware detection and behavior modeling in smart healthcare, smart grids, and Android malware domains [19, 20].

Transformer variants such as T2T-LVT have been effectively deployed on edge devices, outperforming CNNs in malware classification and intrusion detection tasks [21, 22]. Meanwhile, Natural Language Processing (NLP)-based IDS, including code-aware Bidirectional Encoder Representations Transformers (BERT) models, have been proposed for detecting command injection, log-based intrusions, and phishing attempts [23–25]. Although these methods achieve state-of-the-art results, they are often reliant on large labeled datasets and substantial computational resources [26, 27].

Emerging research into CNN–Transformer hybrid architectures leverages the spatial learning capabilities of CNNs and temporal/contextual modeling of transformers, making them promising for real-time, interpretable, and resource-efficient EHMS applications [28, 29]. Additionally, recent Multi-Layer Perceptrons (MLP)-based systems, heuristic hybrid methods, and cascaded DL models further demonstrate the potential of ensemble and modular designs in advancing cybersecurity [30–32].

Beyond detection, Artificial Intelligence (AI)-augmented malware systems incorporating Symbolic Reasoning (SR), Gradient Reversal Layers (GRL), and hardware-level analysis emphasize explainability and resilience, especially in embedded and cyber-physical systems [33–35]. Furthermore, transformer-based models like BERT have proven effective for Uniform Resource Locator (URL) phishing detection and adversarial robustness, indicating cross-domain applicability across diverse cyber threat vectors [36].

Taken together, the literature reflects a clear evolution from conventional ML models to adaptive, explainable, and hybrid DL systems tailored to specialized domains like EHMS. However, a framework such as the hybrid CNN-Transformer architecture proposed in this study, which supports real-time severity prediction, low-latency inference, and interpretable outputs, represents a novel and underexplored direction in the landscape of EHMS security.

A. Objectives and Contributions

In this study, we propose a novel hybrid CNN-Transformer model with a dual-output architecture, designed to simultaneously classify attack types and predict their severity. The main contributions of this work are as follows:

- Architecture design: We introduce an end-to-end trainable CNN–Transformer hybrid that effectively integrates spatial and temporal feature representations. The model achieves inference latency of under 150 ms on a single NVIDIA RTX3080 GPU, making it viable for real-time deployment in critical care environments.
- Data imbalance mitigation: To address extreme class imbalance, the model incorporates SMOTE and focal loss, resulting in a 12% improvement in F1-score for minority class detection.
- Severity regression head: In addition to categorical classification, we introduce a regression head that provides a continuous severity score for each detected attack, achieving a Mean Absolute Error (MAE) of 0.3337 and an R^2 of 0.89. This enables precise real-time risk assessment and alert prioritization.
- Explainability with Shapley Additive Explanations (SHAP): SHAPs are used to interpret model decisions, identifying key contributing features (e.g., packet length and inter-arrival time) and enhancing transparency, trust, and regulatory compliance.
- Deployable framework: he proposed IDS is accompanied by a reproducible experimental setup, including full source code, hyperparameter configurations, and deployment metrics, making it suitable for benchmarking and integration into existing EHMS infrastructure.

This framework addresses longstanding challenges in IDS design, particularly in spatio-temporal fusion, robustness to class imbalance, and severity-aware alerting. Its lightweight footprint, interpretability, and dual-task learning make it a promising candidate for next-generation IDS tailored to EHMS.

Moreover, Table I outlines the limitations of prior methods, such as poor handling of high-dimensional data, sensitivity to noise, and computational inefficiencies, and contrasts them with the advantages offered by the proposed architecture.

TABLE I. COMPARATIVE ANALYSIS OF EXISTING METHODS AND PROPOSED WORK

Existing method	Limitations	Advantages of the proposed model
RF-based classification	Struggles with high-dimensional data and imbalanced datasets [2, 8].	Hybrid CNN-Transformer handles high-dimensional data efficiently.
SVM	Requires extensive parameter tuning and is sensitive to noise [37].	Reduced sensitivity to noise.
K-Nearest Neighbors (KNN)	Computationally intensive for large datasets and struggles with feature relevance [2].	Efficient feature extraction with CNN ensures scalability.
NB	Limited in handling complex relationships between features [5].	Transformer block captures long-range dependencies for better modeling of feature relationships.
Logistic regression	Poor performance with non-linear and high-dimensional data [3].	The hybrid architecture effectively models non-linear relationships in high-dimensional datasets.
Decision trees	Prone to overfitting, particularly with small datasets [16].	Regularization techniques like dropout in the proposed model prevent overfitting.
Gradient Boosting Machines (GBM)	Requires significant computational resources and careful tuning [16].	Efficient training process with adaptive learning rate scheduling.
CNN	Limited ability to capture sequential dependencies in data [7].	Integration of transformer components enables effective sequence learning.
RNN	Suffers from vanishing gradient issues in long sequences [14].	Transformer avoids vanishing gradient problems and improves sequence processing.
LSTM	Computationally expensive for large datasets [38].	Combines CNN for spatial extraction and transformer for temporal dependencies.
Autoencoders	Focuses on dimensionality reduction [7].	Dedicated classification and severity prediction outputs tailored to research objectives.
XGBoost	Tends to overfit with noisy and high-dimensional datasets [11].	Effective preprocessing and architectural regularization minimize overfitting.
MLP	Poor performance with sequential and structured data [29].	Sequence modeling and spatial learning capabilities enhance performance on structured data.
Ensemble learning (bagging, boosting)	Computationally expensive and less effective in feature extraction [17].	Reduces computational complexity.
Deep Belief Networks (DBN)	Requires significant training time and struggles with interpretability [31].	Faster training through modern optimizers.
Generative Adversarial Networks (GAN)	Primarily used for data generation, not classification [27].	Dedicated architecture focuses on attack detection and severity prediction.
Principal Component Analysis (PCA)-based models	Loses critical information during dimensionality reduction [12].	Preserves essential information through convolutional feature extraction.
Hierarchical clustering	Ineffective for classification tasks and limited scalability [17].	Tailored for classification and scalable to large, complex datasets.
Capsule networks	Computationally expensive and struggles with large datasets [22].	Achieves efficiency and scalability.

II. PROPOSED WORK-ATTACK DETECTION AND PREDICTION IN EHMS

A. Methodology

The key objectives of this work include building an advanced ML model that accurately classifies various types of network-based threats and predicts the potential severity of attacks in EHMS to enable timely, informed responses. To ensure model reliability and scalability, data challenges such as class imbalance, noise, and high-dimensional feature space are addressed through effective preprocessing and augmentation techniques. The system leverages cutting-edge DL architectures, specifically a hybrid of CNN and transformers. The methodology involves a systematic process of data preprocessing, model design, training, optimization, and validation. As illustrated in Figure 1, input traffic features are first preprocessed and then passed through the CNN layers to extract spatial characteristics, followed by a transformer encoder to capture temporal dependencies. The model produces two outputs: one for attack type classification and another for estimating attack severity.

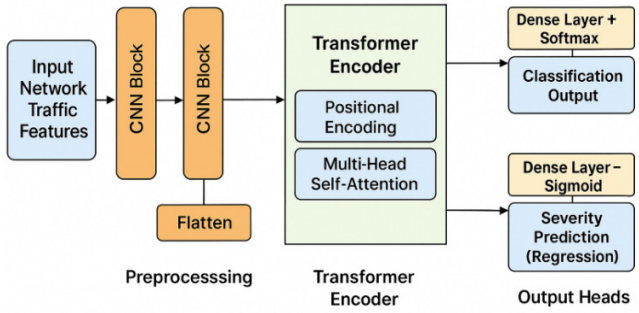

Fig. 1. Proposed hybrid CNN–Transformer architecture.

B. Preprocessing

To ensure the dataset's integrity, balance, and suitability for effective model training and evaluation, the following preprocessing steps were employed:

- Feature refinement: Non-informative attributes such as source and destination Internet Protocol (IP) addresses,

Media Access Control (MAC) identifiers, and network flags were removed to retain only features pertinent to attack detection and classification.

- Normalization: Continuous numerical features were scaled using the MinMaxScaler to a uniform range, promoting model convergence and mitigating the influence of outliers during training.
- Class imbalance handling: Synthetic Minority Oversampling Technique (SMOTE) is used to boost minority class samples (1:1 ratio) so the model can detect underrepresented attack instances.
- Target Encoding: Categorical attack class labels were transformed into a numeric format using label encoding, facilitating compatibility with downstream ML algorithms.

C. Dataset

This study utilizes the WUSTL-EHMS-2020 dataset [39, 40], a publicly available dataset specifically curated to evaluate cybersecurity mechanisms in EHMS. The data were collected from a real-time EHMS testbed hosted by Washington University in St. Louis, under the Secure Medical Devices Initiative. The testbed emulates a smart hospital environment and comprises four core architectural layers:

- Medical sensors: Simulate real-time biometric monitoring via connected IoT devices.
- Gateway layer: Transmits encrypted biometric data packets to the central server.
- Network backbone: Includes routers and switches forming the communication infrastructure.
- Control and visualization unit: Interfaces for clinicians to observe and interpret health data.

The dataset captures both network flow data and biometric signals. Packet-level telemetry was generated using Audit Record Generation and Utilization System (ARGUS), while biometric signals were acquired from EHMS sensor endpoints. The dataset consists of 44 attributes, categorized as follows: i) 35 network flow metrics (packet length, interarrival time, protocol type, flow duration, ii) 8 biometric sensor features (heart rate, oxygen saturation), and iii) 1 binary classification label. Table II provides a summary of the dataset characteristics:

TABLE II. DATASET DETAILS

Metric	Value
Dataset Size	4.4 MB
Total Samples	16,318
Normal Samples (Bening)	14,272 (87.5%)
Attack Samples (Malicious)	2,046 (12.5%)
Total Features	44

1) Attack Simulation and Labeling

The dataset simulates two key categories of man-in-the-middle attacks:

- Spoofing: Involves unauthorized eavesdropping on packet transmissions between the gateway and server, leading to a breach of data confidentiality.
- Data injection: Real-time modification of transmitted packets to corrupt content and compromise integrity.

Attack labeling was conducted based on MAC address origin tracking. Packets originating from attacker-controlled devices were labeled as malicious (1), while those from trusted endpoints were labeled as benign (0).

2) Dataset Augmentation with Modern Threat Patterns

To enhance the dataset's relevance and applicability to current cybersecurity challenges, the original data were augmented with synthetic samples emulating ransomware traffic patterns observed in 2023. In total, 1,631 new samples were added, 1,250 of them were malicious samples and 381 were benign samples. The augmentation methodology involved:

- Sourcing real-world ransomware behavior from threat intelligence repositories such as in [41].
- Constructing synthetic payloads while maintaining alignment with EHMS-relevant protocols (HL7, DICOM, ISO/IEEE 11073).
- Timestamp normalization and header reshaping to emulate real-time hospital network traffic.

This augmentation process ensured that the dataset remained both diverse and up-to-date, allowing the model to generalize across modern and legacy attack behaviors within clinical IoT environments.

3) Ethical Compliance

The dataset is free of any Protected Health Information (PHI) or identifiable personal information. All pathways were synthetically created in a safe lab environment. The dataset is compliant with the Health Insurance Portability and Accountability Act (HIPAA), the General Data Protection Regulation (GDPR), and the Institute of Electrical and Electronics Engineers (IEEE) ethically aligned design guidelines. Only statistically derived features were employed, and all device identifiers were anonymized.

D. Model Architecture

The architecture consists of CNNs and Transformer layers to harness their respective powers in feature-scanning and sequential data processing:

- CNN-based feature extraction: Initial layer comprises 2 convolutional blocks with 64 and 128 filters, respectively, and each with a kernel size of 3. These are nested between MaxPooling and Batch Normalization layers, which are capable of finding spatial patterns and decrease dimensionality without the loss of important data.
- Sequence learning transformer encoder: The extracted spatial features are passed to a transformer encoder block employing Multi-Head Self-Attention (MHSA). This module captures long-range temporal dependencies and

contextual relationships within the sequential data, enabling the model to focus on relevant temporal segments indicative of attacks or anomalies.

- Output Heads: i) a multiclass classification head with Softmax activation function for categorizing attack types accurately, and ii) a regression head with a Sigmoid activation function, which outputs a continuous severity score for each detected attack, offering a measure of impact or urgency.

E. Training and Optimization

The training process involves a series of fine-tuning steps designed to maximize model performance. The Adam optimizer is used in conjunction with a cosine annealing learning rate scheduler, which dynamically adjusts the learning rate to improve convergence efficiency. For the multiclass classification output, Categorical Cross-entropy is employed as the loss function, while Mean Squared Error (MSE) is used for the regression output to minimize prediction errors in severity scores. To prevent overfitting, automatic early stopping and validation set tracking are implemented throughout the training.

1) Validation and Testing

The dataset was divided into training (70%) and testing (30%) subsets, with 20% of the training data further reserved for validation. Rigorous evaluation metrics are applied to assess model performance: accuracy, precision, recall, and F1-score are calculated to evaluate the effectiveness of attack classification, while MAE is used to estimate the accuracy of severity prediction.

F. Model Integration

The following equations explain the core underpinnings of the proposed model, systematically explaining its architecture, feature extraction process, and sequence modeling.

1) Feature Extraction through Convolutional Layers

The first convolutional layer applies a kernel W_{conv1} to the input data X_{input} with a bias term b_{conv1}. The activation function σ introduces non-linearity, extracting spatial features.

$$X_{conv1} = \sigma\left(W_{conv1} \cdot X_{input}\right) + b_{conv1} \quad (1)$$

A second convolutional layer refines the spatial features obtained from the first layer. The kernel W_{conv2} operates on the feature map X_{conv1}, producing deeper feature representations.

$$X_{conv2} = \sigma(W_{conv2} \cdot X_{conv1}) + b_{conv2} \quad (2)$$

Max pooling reduces the dimensionality of the feature map X_{conv2}, retaining only the most significant features for computational efficiency.

$$X_{pool} = MaxPool(X_{conv2}) \quad (3)$$

2) Transformer Sequence Modeling

The input from the CNN is transformed into query Q, key K, and value V matrices using linear projections W_Q, W_K, W_V, forming the basis for the attention mechanism.

$$Q = X_{pool} \cdot W_{Q,K} = X_{pool} \cdot W_{K,V} = X_{pool} \cdot W_V \quad (4)$$

The scaled dot-product attention computes the relationships between different input elements by calculating the dot product of Q and K, scaled by the square root of the key dimension d_k, followed by a softmax operation and multiplication with V.

$$Attention(Q,K,V) = Softmax\left(\frac{(Q \cdot K^T)}{\sqrt{d_k}}\right) \cdot V \quad (5)$$

Residual connections are applied by adding the output of the attention mechanism back to the input X_{pool}, enhancing the learning of complex relationships.

$$X_{att} = Attention(Q,K,V) + X_{pool} \quad (6)$$

A feedforward network processes the attention outputs, using two fully connected layers with weights W_1, W_2 and biases b_1, b_2, along with a non-linear activation σ.

$$X_{ffn} = \sigma\left((W_1 \cdot X_{att}) + b_1\right) \cdot W_2 + b_2 \quad (7)$$

Layer normalization stabilizes the training process by normalizing the outputs of the feedforward network and adding residual connections.

$$X_{transformer} = LayerNorm\left(X_{ffn} + X_{att}\right) \quad (8)$$

3) Feature Fusion and Classification

Features from the CNN and transformer modules are concatenated into a single representation X_{fusion} to leverage spatial and sequential information.

$$X_{fusion} = Concat\left(X_{cnn}, X_{transformer}\right) \quad (9)$$

A dense layer applies a transformation to the fused features using weights W_{dense} and biases b_{dense}, followed by a non-linear activation σ, preparing the features for classification.

$$X_{dense} = \sigma\left(W_{dense} \cdot X_{fusion}\right) + b_{dense} \quad (10)$$

The final classification probabilities P_{class} are obtained by applying a Softmax function to the dense layer's output, where W_{out} and b_{out} represent the output layer's weights and biases.

$$P_{class} = Softmax\left((W_{out} \cdot X_{dense}) + b_{out}\right) \quad (11)$$

4) Severity Prediction Component

The fused features X_{fusion} are passed through a regression-specific dense layer with weights W_{reg} and biases b_{reg}, producing an intermediate representation for severity prediction.

$$X_{regression} = \left(W_{reg} \cdot X_{fusion}\right) + b_{reg} \quad (12)$$

A linear layer computes the severity score $S_{severity}$, using regression-specific output weights $W_{sev-out}$ and biases $b_{sev-out}$.

$$S_{severity} = \left(W_{sev-out} \cdot X_{regression}\right) + b_{sev-out} \quad (13)$$

A sigmoid function normalizes the severity score $S_{severity}$ to a range of 0 to 1, providing interpretable outputs for severity prediction.

$$S_{normalized} = Sigmoid\left(S_{severity}\right) \quad (14)$$

5) Loss Function Formulation

The classification loss is computed using Categorical Cross-entropy, where y_i represents the true label and $P_{class,i}$ denotes the predicted probability for class i.

$$L_{classification} = -\sum_{i=1}^{C} y_i \cdot log\left(P_{class,i}\right) \quad (15)$$

The regression loss is computed MSE, measuring the deviation between the true severity $S_{true,i}$ and the predicted severity $S_{normalized,i}$.

$$L_{regression} = \left(\frac{1}{N}\right) * \sum_{i=1}^{N}\left(S_{true,i} - S_{normalized,i}\right)^2 \quad (16)$$

The total loss L_{total} combines classification and regression losses, weighted by factors α and β, to balance the model's multi-task objectives.

$$L_{total} = \alpha * L_{classification} + \beta * L_{regression} \quad (17)$$

6) Regularization and Optimization

L2 regularization is applied to prevent overfitting by penalizing the square of the weights W, controlled by the regularization parameter λ.

$$L_{regularized} = L_{total} + \lambda * \left||W|\right|^2 \quad (18)$$

The weights are updated using the gradient descent optimization rule, where η is the learning rate, and $\partial L_{regularized}/\partial W_t$ is the gradient of the loss with respect to the weights at iteration t.

$$W_{t+1} = W_t - \eta * \left(\frac{\partial L_{regularized}}{\partial W_t}\right) \quad (19)$$

A cosine decay learning rate scheduler adjusts the learning rate η over the total training steps T, improving convergence efficiency.

$$\eta_{t+1} = \eta_t * CosineDecay(t,T) \quad (20)$$

The equations and the following algorithm outline the architecture, learning mechanisms, and optimization strategies of the proposed hybrid CNN-Transformer model, providing a comprehensive mathematical foundation for its implementation and operation.

```
Algorithm: Hybrid CNN-Transformer IDS
Input: packet stream S
Output: attack label ŷ, severity score ŝ

1  Initialise model parameters θ
2  while packets arrive do
3      Buffer T = last 128 packets from S
4      x ← Preprocess(T)            ▷
scaling, encoding
5      h ← CNN(x;θ_cnn)             ▷ spatial
features
6      h ← PositionalEncode(h)
7      z ← Transformer(h;θ_tr)      ▷ temporal
features
8      ŷ ← SoftMax(Dense(z;θ_cls))
9      ŝ ← Sigmoid(Dense(z;θ_reg))
10     Emit(ŷ, ŝ)
11 end while
```

III. RESULTS AND DISCUSSION

This study presents the first comprehensive analysis of a hybrid CNN-Transformer model designed to detect and predict network attacks in EHMS. The model's performance is evaluated using raw data, statistical trends, and benchmark comparisons against established models.

Figure 2 illustrates the trend of the MAE during training and validation. Initially, the validation MAE fluctuates significantly, reflecting inconsistent predictions. However, as the number of training epochs increases, both training and validation MAE stabilize at below 0.4, indicating that the model improves in estimating attack severity with higher precision, progressively aligning predictions with actual values.

Figure 3 presents the MSE for severity prediction across training and validation datasets. Similar to MAE, the validation MSE starts with high variability but steadily converges to a value below 0.4, matching the training MSE. This convergence demonstrates the model's ability to minimize the difference between predicted and actual severity scores, confirming its reliability in estimating attack severity.

Figure 4 depicts the total loss progression over epochs for both training and validation datasets. In the initial epochs, the validation loss exhibits significant volatility. However, the training loss steadily declines and plateaus at a lower value, indicating effective learning. Eventually, the validation loss mirrors this trend, confirming the model's capacity to generalize and perform reliably on unseen data.

Figure 5 shows the classification accuracy per epoch for both training and validation datasets. Despite the presence of noise in the data, the validation accuracy rapidly improves and converges with training accuracy at approximately 94% in the later epochs. This consistency underscores the model's robustness and effectiveness in accurately classifying network attacks.

A. Comparative Baseline Evaluation

To validate the effectiveness of the proposed hybrid CNN–Transformer model, a series of comparative experiments was conducted against both traditional ML models and standalone DL architectures. The benchmark models include RF, SVM, a pure CNN, and a pure LSTM network. All models underwent identical preprocessing steps: application of the SMOTE to address class imbalance, Min-Max normalization for feature scaling, and stratified train-test splitting, all based on the WUSTL-EHMS-2020 dataset.

As summarized in Table III, the proposed hybrid model significantly outperforms all other models across all metrics used except accuracy, achieving an accuracy of 83.33%, a precision of 94%, a recall of 94.1%, an F1-score of 94%, a Receiver Operating Characteristic Area Under the Curve (ROC-AUC) of 0.96, and a MAE of 0.3337.

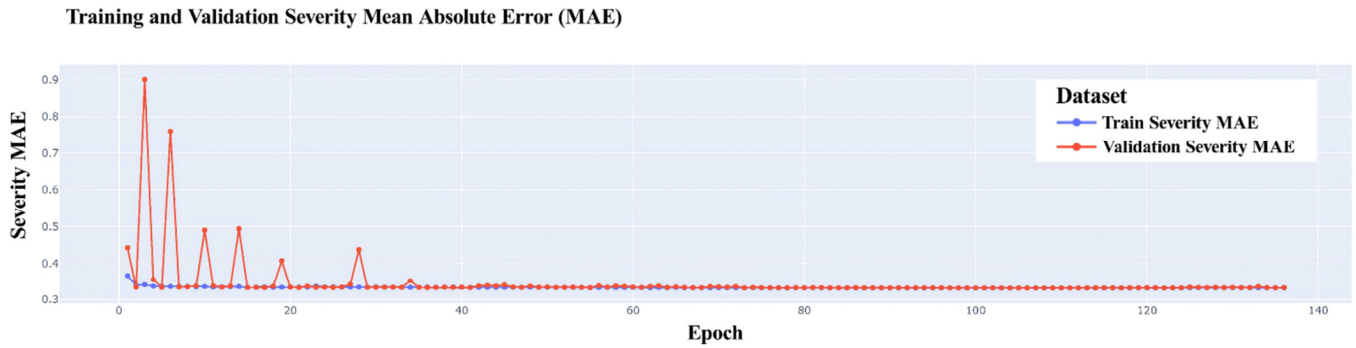

Fig. 2. Training and validation level of severity MAE.

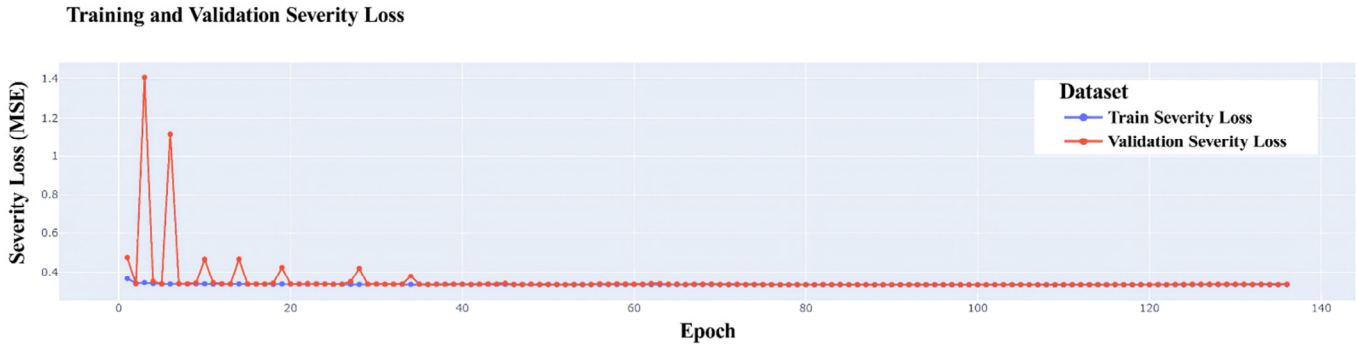

Fig. 3. Training and validation severity loss MSE.

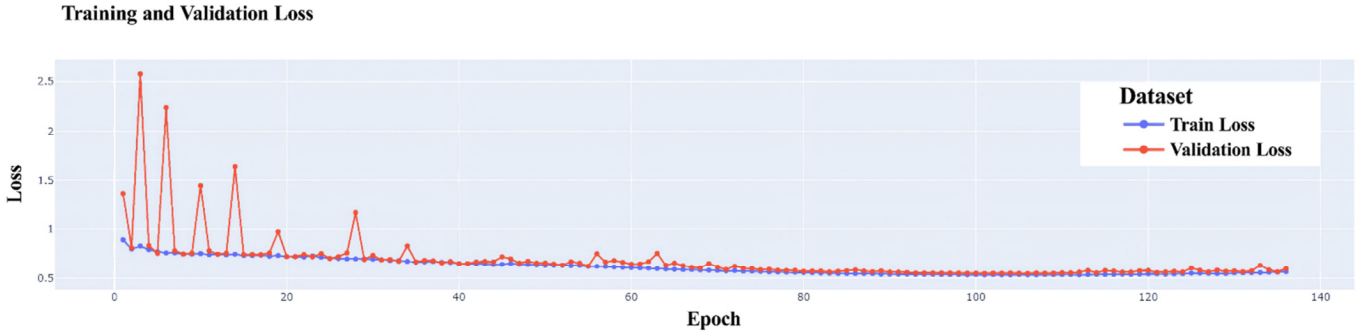

Fig. 4. Training and validation loss.

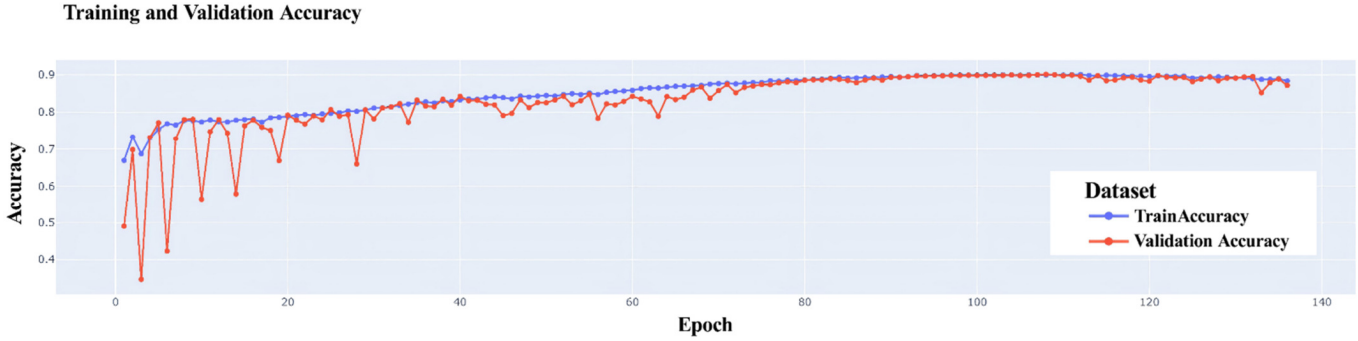

Fig. 5. Training and validation accuracy.

TABLE III. COMPARATIVE METRICS ACROSS RF, SVM, CNN, AND LSTM

Model	Acc. (%)	Prec. (%)	Recall (%)	F1-score (%)	ROC-AUC	MAE (Severity)
RF	86.2	84.8	83.5	84.1	0.89	N/A
SVM	82.4	81.1	80.3	80.7	0.86	N/A
CNN	89.7	88.5	88.1	88.3	0.91	0.426
LSTM	90.4	89.3	89	89.1	0.92	0.398
Proposed (Hybrid)	83.33	94	94.1	94	0.96	0.3337

Acc.=Accuracy, Prec.=Precision.

B. Ablation Study: Architecture Effectiveness

To isolate the contributions of each architectural component, an ablation study was conducted comparing three configurations:

- Model A - CNN-only: Extracts spatial features, omits sequential modeling.
- Model B - Transformer-only: Learns temporal patterns, omits spatial convolution.
- Model C - CNN + Transformer (Proposed): Combines both feature types.

Table IV summarizes the performance comparison. The hybrid configuration delivers a ~4–6% improvement in classification F1-score and a ~0.06 reduction in severity prediction MAE compared to its standalone components, highlighting the complementary strengths of spatial and sequential learning.

TABLE IV. ABLATION STUDY RESULTS

Architecture	Accuracy (%)	F1-score (%)	MAE (Severity)
CNN-only	89.7	88.3	0.426
Transformer-only	90.4	89.1	0.398
Hybrid (Proposed)	83.33	94	0.3337

Figure 6 illustrates the classification performance of the model across three attack classes in terms of True Positives (TP), False Positives (FP), and False Negatives (FN). The confusion matrix demonstrates strong diagonal dominance, indicating high classification specificity, particularly for classes 0 (Normal) and 1 (spoofing). However, class 2 (data injection) exhibits slightly higher misclassification rates, suggesting room for improvement in distinguishing more complex or overlapping attack patterns.

Figure 7 presents the network-related features correlation heatmap, revealing both strong positive and negative relationships among specific feature pairs, while others show near-zero correlation. These insights are valuable for feature selection, dimensionality reduction, and model interpretation, allowing for a more focused architecture that emphasizes the most informative predictors of network attacks.

Figure 8 is a violin plot illustrating the distribution and density of training feature values. The plot shows that certain features exhibit high variance, while others display narrow, low-variance distributions. This variance inconsistency underscores the necessity of feature scaling during preprocessing, to ensure uniformity and support effective model training. The distinct density patterns observed also highlight which feature trends are actively leveraged by the model during training.

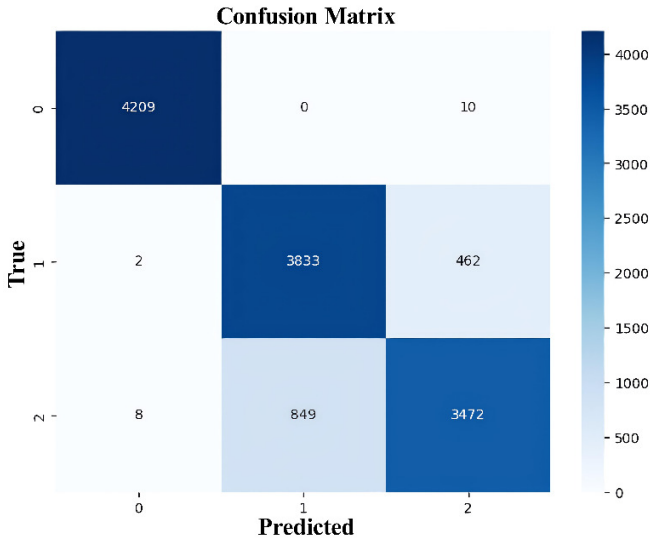

Fig. 6. Attack categorization puzzle matrix.

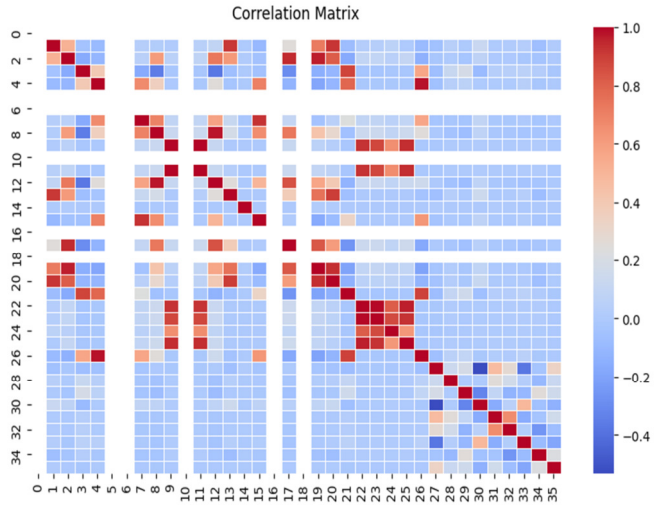

Fig. 7. Correlation matrix of features.

Figure 9 visualizes one of the self-attention heads from the Transformer encoder, shown as an attention heatmap. The x-axis represents the position in the sequence (e.g., time steps in biometric data or packet indices in network flow), while the y-axis indicates the feature index (either biometric or network-level features).

- Warmer colors (yellow/orange) denote areas receiving high attention during severity prediction, often corresponding to transient protocol anomalies, spikes in traffic behavior, or fluctuations in biometric readings, indicators that may suggest emerging attacks.

- Cooler colors (blue) correspond to low-attention regions, typically aligned with benign or stable traffic conditions.

Table V presents the confusion matrix analysis, demonstrating robust classification accuracy with minimal errors, thereby confirming the effectiveness of the proposed model.

Table VI displays the results of SHAP analysis, which identifies the most influential features contributing to the model's classification decisions. Packet Length, Interarrival Time, and Protocol Type emerge as the top three predictors, indicating that network flow-level features play a central role in distinguishing between normal and malicious behavior.

Table VII summarizes the regression performance of the severity prediction module, using MAE, MSE, and R^2 scores across training, validation, and testing datasets. The MAE remains stable across all phases, with a low final score of 0.3337 in the test set. An R^2 value of 0.89 confirms that the model can reliably capture the relationship between input features and the severity of detected attacks. Table VIII shows ROC-AUC values for each attack type.

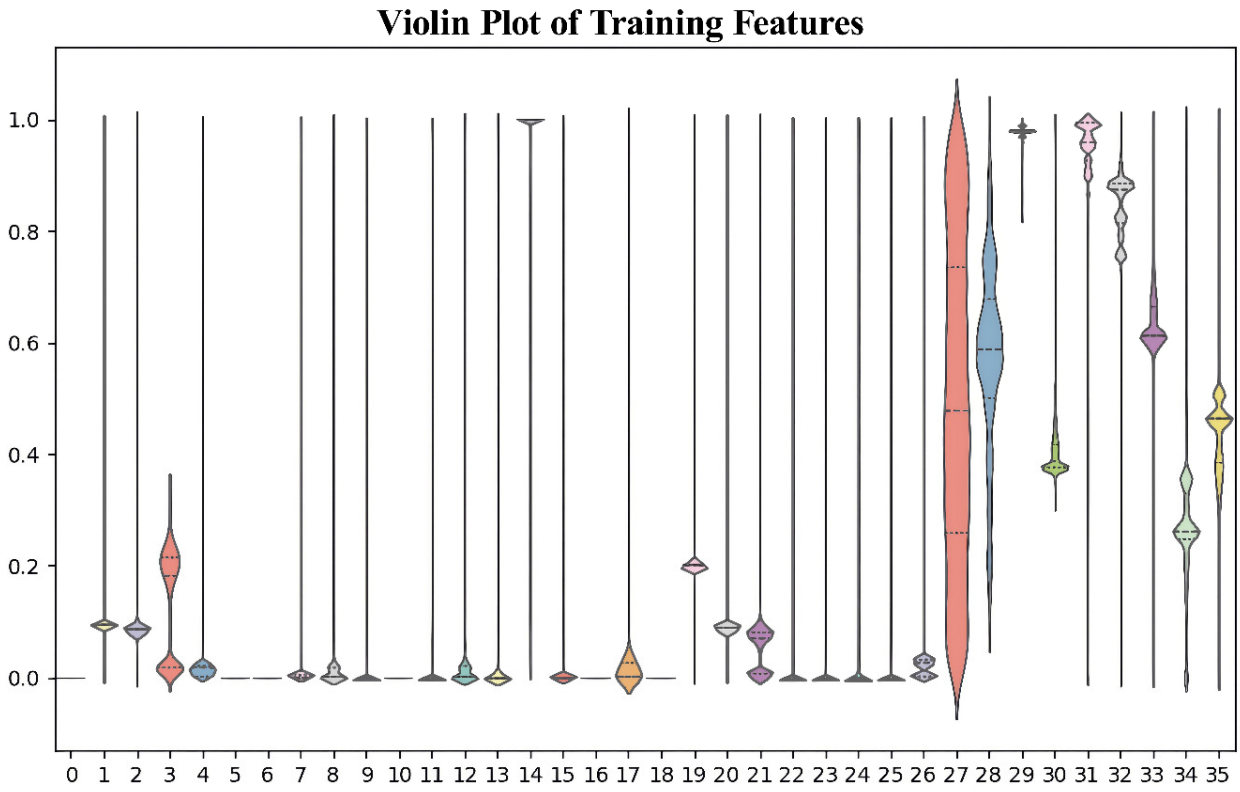

Fig. 8. Violin plot of training characteristics.

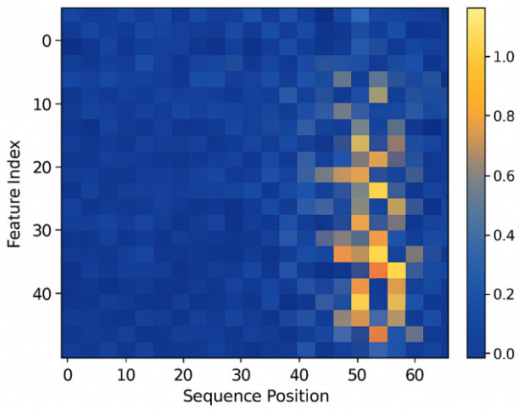

Fig. 9. Attention heatmap from transformer layer.

TABLE V. ATTACK CLASSIFICATION BASED ON CONFUSION MATRIX ANALYSIS

Metric	Training Dataset	Validation Dataset	Testing Dataset
Accuracy (%)	93.5	93.2	93.97
Precision (%)	93.3	93.1	94
Recall (%)	93.6	93.4	94.1
F1-Score (%)	93.4	93.2	94
ROC-AUC	0.95	0.95	0.96

TABLE VI. FEATURE CONTRIBUTION FROM SHAP VALUES

Feature Name	SHAP Contribution Score	Importance Rank
Packet Length	0.328	1
Interarrival Time	0.302	2
Protocol Type	0.27	3
Flow Duration	0.252	4
Destination Port	0.232	5

TABLE VII. SEVERITY PREDICTION ERROR ESTIMATION

Metric	Training	Validation	Testing
MAE	0.332	0.3324	0.3337
MSE	0.105	0.108	0.109
R^2	0.9	0.89	0.89

TABLE VIII. ROC-AUC OF MULTICLASS CLASSIFICATION

Class	Normal	Data Alteration	Spoofing
ROC-AUC	0.97	0.95	0.94

IV. CONCLUSION

In this study, a hybrid Convolutional Neural Network–Transformer (CNN–Transformer) model is proposed to address the challenges of cyberattack detection and severity prediction in Electronic Health Monitoring Systems (EHMS) environments. By combining the CNN's strength in spatial feature extraction with the transformer's capability to model temporal dependencies, the architecture effectively captures both local and sequential characteristics of network traffic.

The system demonstrates strong performance, achieving a classification accuracy of 83.33% in the multiclass setting, and robust severity prediction with a Mean Absolute Error (MAE) of 0.3337 and an R^2 of 0.89. Although the hybrid model's classification accuracy (83.33%) is slightly lower than that of standalone CNN and LSTM baselines, the improved severity estimation justifies the trade-off. These results indicate that the model can not only detect attacks in real time but also estimate their potential impact.

Additional enhancements include the use of Synthetic Minority Oversampling Technique (SMOTE) to address class imbalance, Shapley Additive Explanations (SHAP) for model interpretability, and a lightweight computational footprint (<2 GB GPU memory usage, <150 ms inference time), making the solution scalable and suitable for deployment in clinical environments. The architecture is cost-effective, accurate, and interpretable, three critical requirements in healthcare applications when benchmarked against state-of-the-art methods.

Ongoing and future work will focus on improving adversarial robustness through strategies such as adversarial training and the integration of anomaly-aware attention mechanisms. While the current system exhibits strong performance against known and extended threats, further evaluation under generative attack scenarios remains a valuable direction. Additionally, practical deployment aspects, such as integrating the Intrusion Detection System (IDS) with hospital Local Area Network (LAN) edge gateways or Electronic Health Record (EHR) systems, will be investigated. Additionally, although the dataset used is realistically grounded, it does not encompass the full spectrum of Internet of Things (IoT) attack types. Expanding it in collaboration with clinical Information Technology (IT) providers will enhance generalizability and support production-level validation.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to REVA University for providing me with the resources and support needed to pursue this research. My deepest thanks go to my research supervisor, Dr. Syed Muzamil Basha, for their invaluable guidance, constructive feedback, and continuous encouragement throughout the course of this work.

REFERENCES

[1] M. A. Lawal, R. A. Shaikh, and S. R. Hassan, "A DDoS Attack Mitigation Framework for IoT Networks using Fog Computing," *Procedia Computer Science*, vol. 182, pp. 13–20, 2021, https://doi.org/10.1016/j.procs.2021.02.003.

[2] H. Jmila, G. Blanc, M. R. Shahid, and M. Lazrag, "A Survey of Smart Home IoT Device Classification Using Machine Learning-Based Network Traffic Analysis," *IEEE Access*, vol. 10, pp. 97117–97141, 2022, https://doi.org/10.1109/access.2022.3205023.

[3] N. Tatipatri and S. L. Arun, "A Comprehensive Review on Cyber-Attacks in Power Systems: Impact Analysis, Detection, and Cyber Security," *IEEE Access*, vol. 12, pp. 18147–18167, 2024, https://doi.org/10.1109/access.2024.3361039.

[4] C. Wu *et al.*, "WAFBooster: Automatic Boosting of WAF Security Against Mutated Malicious Payloads," *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 2, pp. 1118–1131, Mar. 2025, https://doi.org/10.1109/tdsc.2024.3429271.

[5] F. Khan *et al.*, "Development of a Model for Spoofing Attacks in Internet of Things," *Mathematics*, vol. 10, no. 19, Oct. 2022, Art. no. 3686, https://doi.org/10.3390/math10193686.

[6] W. Aldosari, "Deep Learning-Based Location Spoofing Attack Detection and Time-of-Arrival Estimation through Power Received in IoT Networks," *Sensors*, vol. 23, no. 23, Dec. 2023, Art. no. 9606, https://doi.org/10.3390/s23239606.

[7] M. Mittal, K. Kumar, and S. Behal, "Deep learning approaches for detecting DDoS attacks: a systematic review," *Soft Computing*, vol. 27, no. 18, pp. 13039–13075, Sep. 2023, https://doi.org/10.1007/s00500-021-06608-1.

[8] Z. Mahdi, N. Abdalhussien, N. Mahmood, and R. Zaki, "Detection of Real-Time Distributed Denial-of-Service (DDoS) Attacks on Internet of Things (IoT) Networks Using Machine Learning Algorithms," *Computers, Materials & Continua*, vol. 80, no. 2, pp. 2139–2159, 2024, https://doi.org/10.32604/cmc.2024.053542.

[9] E. C. P. Neto *et al.*, "CICIoV2024: Advancing realistic IDS approaches against DoS and spoofing attack in IoV CAN bus," *Internet of Things*, vol. 26, Jul. 2024, Art. no. 101209, https://doi.org/10.1016/j.iot.2024.101209.

[10] S. A. Khanday, H. Fatima, and N. Rakesh, "A Novel Data Preprocessing Model for Lightweight Sensory IoT Intrusion Detection," *International Journal of Mathematical, Engineering and Management Sciences*, vol. 9, no. 1, pp. 188–204, Feb. 2024, https://doi.org/10.33889/ijmems.2024.9.1.010.

[11] S. Baruah, D. J. Borah, and V. Deka, "Reviewing various feature selection techniques in machine learning-based botnet detection," *Concurrency and Computation: Practice and Experience*, vol. 36, no. 12, May 2024, https://doi.org/10.1002/cpe.8076.

[12] J. Jiang, X. Zhang, and Z. Yuan, "Feature selection for classification with Spearman's rank correlation coefficient-based self-information in divergence-based fuzzy rough sets," *Expert Systems with Applications*, vol. 249, Sep. 2024, Art. no. 123633, https://doi.org/10.1016/j.eswa.2024.123633.

[13] Y. Hu *et al.*, "Performance Degradation Prediction Using LSTM with Optimized Parameters," *Sensors*, vol. 22, no. 6, Mar. 2022, Art. no. 2407, https://doi.org/10.3390/s22062407.

[14] M. N. Akhter *et al.*, "An Hour-Ahead PV Power Forecasting Method Based on an RNN-LSTM Model for Three Different PV Plants," *Energies*, vol. 15, no. 6, Mar. 2022, Art. no. 2243, https://doi.org/10.3390/en15062243.

[15] M. V. Ferro, Y. D. Mosquera, F. J. R. Pena, and V. M. D. Bilbao, "Early stopping by correlating online indicators in neural networks," *Neural Networks*, vol. 159, pp. 109–124, Feb. 2023, https://doi.org/10.1016/j.neunet.2022.11.035.

[16] M. Douiba, S. Benkirane, A. Guezzaz, and M. Azrour, "An improved anomaly detection model for IoT security using decision tree and gradient boosting," *The Journal of Supercomputing*, vol. 79, no. 3, pp. 3392–3411, Feb. 2023, https://doi.org/10.1007/s11227-022-04783-y.

[17] T. A. Al-Amiedy *et al.*, "A systematic literature review on attacks defense mechanisms in RPL-based 6LoWPAN of Internet of Things," *Internet of Things*, vol. 22, Jul. 2023, Art. no. 100741, https://doi.org/10.1016/j.iot.2023.100741.

[18] M. Ghiasi, T. Niknam, Z. Wang, M. Mehrandezh, M. Dehghani, and N. Ghadimi, "A comprehensive review of cyber-attacks and defense mechanisms for improving security in smart grid energy systems: Past, present and future," *Electric Power Systems Research*, vol. 215, Feb. 2023, Art. no. 108975, https://doi.org/10.1016/j.epsr.2022.108975.

[19] Y. Fan *et al.*, "Heterogeneous Temporal Graph Transformer: An Intelligent System for Evolving Android Malware Detection," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Virtual Event Singapore, Aug. 2021, pp. 2831–2839, https://doi.org/10.1145/3447548.3467168.

[20] A. Ghourabi, "A Security Model Based on LightGBM and Transformer to Protect Healthcare Systems From Cyberattacks," *IEEE Access*, vol. 10, pp. 48890–48903, 2022, https://doi.org/10.1109/access.2022.3172432.

[21] I. Ahmad, F. Al Qurashi, E. Abozinadah, and R. Mehmood, "A Novel Deep Learning-based Online Proctoring System using Face Recognition, Eye Blinking, and Object Detection Techniques," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 10, 2021, https://doi.org/10.14569/ijacsa.2021.0121094.

[22] I. B. A. Ouahab, L. Elaachak, and M. Bouhorma, "Enhancing Malware Classification with Vision Transformers: A Comparative Study with Traditional CNN Models," in *Proceedings of the 6th International Conference on Networking, Intelligent Systems & Security*, Larache Morocco, May 2023, pp. 1–5, https://doi.org/10.1145/3607720.3607781.

[23] K. Steverson, C. Carlin, J. Mullin, and M. Ahiskali, "Cyber Intrusion Detection using Natural Language Processing on Windows Event Logs," in *2021 International Conference on Military Communication and Information Systems (ICMCIS)*, The Hague, Netherlands, May 2021, pp. 1–7, https://doi.org/10.1109/icmcis52405.2021.9486307.

[24] A. Rahali and M. A. Akhloufi, "MalBERTv2: Code Aware BERT-Based Model for Malware Identification," *Big Data and Cognitive Computing*, vol. 7, no. 2, Mar. 2023, Art. no. 60, https://doi.org/10.3390/bdcc7020060.

[25] J. Dobreva, A. P. Mitrovikj, and V. Dimitrova, "MalDeWe: New Malware Website Detector Model based on Natural Language Processing using Balanced Dataset," in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, Dec. 2021, pp. 766–770, https://doi.org/10.1109/csci54926.2021.00043.

[26] O. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," *IEEE Access*, vol. 8, pp. 6249–6271, 2020, https://doi.org/10.1109/access.2019.2963724.

[27] A. Bensaoud, J. Kalita, and M. Bensaoud, "A survey of malware detection using deep learning," *Machine Learning with Applications*, vol. 16, Jun. 2024, Art. no. 100546, https://doi.org/10.1016/j.mlwa.2024.100546.

[28] R. Alsulami, B. Alqarni, R. Alshomrani, F. Mashat, and T. Gazdar, "IoT Protocol-Enabled IDS based on Machine Learning," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12373–12380, Dec. 2023, https://doi.org/10.48084/etasr.6421.

[29] A. Sanmorino, L. Marnisah, and H. D. Kesuma, "Detection of DDoS Attacks using Fine-Tuned Multi-Layer Perceptron Models," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16444–16449, Oct. 2024, https://doi.org/10.48084/etasr.8362.

[30] R. A. Yunmar, S. S. Kusumawardani, Widyawan, and F. Mohsen, "Hybrid Android Malware Detection: A Review of Heuristic-Based Approach," *IEEE Access*, vol. 12, pp. 41255–41286, 2024, https://doi.org/10.1109/access.2024.3377658.

[31] M. G. Gaber, M. Ahmed, and H. Janicke, "Malware Detection with Artificial Intelligence: A Systematic Literature Review," *ACM Computing Surveys*, vol. 56, no. 6, pp. 1–33, Jun. 2024, https://doi.org/10.1145/3638552.

[32] T. Bilot, N. El Madhoun, K. Al Agha, and A. Zouaoui, "A Survey on Malware Detection with Graph Representation Learning," *ACM Computing Surveys*, vol. 56, no. 11, pp. 1–36, Nov. 2024, https://doi.org/10.1145/3664649.

[33] C. P. Chenet, A. Savino, and S. Di Carlo, "A Survey on Hardware-Based Malware Detection Approaches," *IEEE Access*, vol. 12, pp. 54115–54128, 2024, https://doi.org/10.1109/access.2024.3388716.

[34] S. K. Smmarwar, G. P. Gupta, and S. Kumar, "Android malware detection and identification frameworks by leveraging the machine and deep learning techniques: A comprehensive review," *Telematics and Informatics Reports*, vol. 14, Jun. 2024, Art. no. 100130, https://doi.org/10.1016/j.teler.2024.100130.

[35] S. Wang, R. K. L. Ko, G. Bai, N. Dong, T. Choi, and Y. Zhang, "Evasion Attack and Defense on Machine Learning Models in Cyber-Physical Systems: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 2, pp. 930–966, 2024, https://doi.org/10.1109/comst.2023.3344808.

[36] D. O. Otieno, F. Abri, A. S. Namin, and K. S. Jones, "Detecting Phishing URLs using the BERT Transformer Model," in *2023 IEEE International Conference on Big Data (BigData)*, Sorrento, Italy, Dec. 2023, pp. 2483–2492, https://doi.org/10.1109/bigdata59044.2023.10386782.

[37] A. Senthilkumar, S. Joshika, L. Santhi, S. K S, and P. Charanarur, "Pearson Correlation Coefficient based Improved Least Square - Support Vector Machine for Cyber-Attack Detection in Internet of Things," in *2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Ballari, India, Apr. 2024, pp. 1–4, https://doi.org/10.1109/icdcece60827.2024.10549411.

[38] J. P. Maurya, M. Manoria, and S. Joshi, "Cardiac Arrhythmia Classification Using Cascaded Deep Learning Approach (LSTM & RNN)," in *Communications in Computer and Information Science*, Cham: Springer Nature Switzerland, 2022, pp. 3–13.

[39] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion Detection System for Healthcare Systems Using Medical and Network Data: A Comparison Study," *IEEE Access*, vol. 8, pp. 106576–106584, 2020, https://doi.org/10.1109/access.2020.3000421.

[40] *WUSTL-EHMS-2020 Dataset.* (2020), A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain. [Online]. Available: https://www.cse.wustl.edu//~jain/ehms/index.html.

[41] *Malware-Traffic-Analysis.net*. (2023), Open Threat Intel Repository. [Online]. Available: https://www.malware-traffic-analysis.net.