A Deep Feature Ensemble Framework for Intrusion Detection in Internet of Medical Things

Chitty Avula
Department of Computer Science and Technology, Sri Krishnadevaraya University, Ananthapuramu, India
avulamaheswari@gmail.com (corresponding author)

Sathyanarayana Bachala
Department of Computer Science and Technology, Sri Krishnadevaraya University, Ananthapuramu, India
bachalasatya@gmail.com

Received: 19 May 2025 | Revised: 9 June 2025 and 27 June 2025 | Accepted: 23 July 2025

 DOI: https://doi.org/10.48084/etasr.12242

ABSTRACT

The Internet of Medical Things (IoMT) has transformed healthcare through real-time monitoring, remote diagnostics, and intelligent analytics. However, the rapid proliferation of connected medical devices introduces significant cybersecurity threats that endanger patient safety and data privacy. To address these challenges, this study proposes SIDHELNet (Semantically Improved Deep feature ensemble-driven Heterogeneous Ensemble Learning Network), a novel intrusion detection framework tailored for the IoMT environment. SIDHELNet introduces a novel fusion of semantic Word2Vec embeddings with parallel Bi-LSTM and Bi-GRU layers to extract enriched temporal-spatial features, followed by a nine-classifier heterogeneous ensemble. This design distinguishes SIDHELNet from prior hybrid IDS by enabling high adaptability and robustness across heterogeneous IoMT traffic. These features are classified using a Heterogeneous Ensemble Learning (HEL) model that fuses predictions from multiple learners to improve accuracy and robustness. Experimental evaluations on the WUSDL-EHMS-2020 and UNSW-NB15 datasets show that SIDHELNet achieves a detection accuracy of 99.61%, precision of 99.58%, recall of 99.62%, F1-score of 99.59%, and an AUC of 0.998, outperforming existing methods and validating SIDHELNet's high reliability, generalizability, and real-time capability.

Keywords-segmentation; cybersecurity; medical devices; IoMT; healthcare; accuracy; precision

I. INTRODUCTION

The Internet of Medical Things (IoMT) enables seamless integration of medical devices, sensors, and applications, facilitating continuous patient monitoring, early diagnosis, and personalized care [1]. By enabling the transmission of real-time health data and intelligent decision-making, IoMT systems significantly improve clinical efficiency and results. Despite these advantages, the interconnected nature and heterogeneity of IoMT devices make the infrastructure highly vulnerable to attacks [2]. These intrusions can compromise patient confidentiality, disrupt healthcare delivery, and damage institutional reputation. Traditional rule-based security mechanisms are often inadequate against these sophisticated and evolving threats, highlighting the need for intelligent and adaptive Intrusion Detection Systems (IDS) [3]. Recent technologies have gained traction in intrusion detection due to their ability to uncover hidden patterns and adapt to new attack types [4]. In particular, deep neural networks, such as CNNs, LSTMs, and GRUs, have shown promise in capturing complex temporal and spatial dependencies in IoMT traffic. IoMT systems generate sparse, heterogeneous, and latency-sensitive data, rendering traditional IDS insufficient due to their reliance on static features, poor adaptability, and limited real-time inference capability. To overcome existing IoMT security limitations, this study proposes SIDHELNet, a robust intrusion detection framework that combines multiple innovations.

In [5], a Deep Neural Network (DNN) achieved a prediction accuracy of 93% on the NSL-KDD dataset. Building on this, variations in the number of hidden layers were explored in [6], and heuristic-based hyperparameter selection was applied to optimize computational efficiency and detection performance. In [7], a more compact DNN was designed with one input layer, three ReLU-activated hidden layers, and a single output layer, achieving an accuracy of 92%. In [8], multiple DNN structures were examined by altering the size of input, hidden, and output layers, reporting an improved IDS

detection accuracy of 99.59% on the UNSW-NB15 dataset. In [9], a hybrid approach combined a DNN with a Stacked Autoencoder (SAE), utilizing ReLU activation to boost IDS performance. Another similar hybrid architecture implemented the SAE-DNN model with a tanh activation function for intrusion prediction [10]. In [11], the LuNet model was implemented using a deep neural architecture tailored for IDS applications, but performance results were limited. In [12], a hybrid model integrated CNN and RNN for efficient intrusion detection within large-scale data environments. In [13], a CNN-based hybrid IDS was paired with Spark to enhance binary classification performance. In [14], a Recurrent Neural Network (RNN)-based framework handled sequential IoT traffic data effectively.

Enhanced models such as LSTM used memory gates to regulate feature retention, leading to high detection accuracy [15]. In [16], an RNN was applied with gating mechanisms to reduce overfitting and improve intrusion classification. In [17], a two-stage deep learning framework combined LSTM with Autoencoders (AE) to compress data dimensionality and improve IDS accuracy. In [18], Bi-LSTM architectures were utilized to improve temporal feature learning, achieving higher performance than traditional RNN models. Bi-LSTM was also adopted in [19] to efficiently model bidirectional dependencies in network traffic data, achieving an accuracy of 97.01% and outperforming conventional deep models such as CNN and GRU [20]. In [21], a Bi-LSTM deep model was used to capture intricate patterns in network intrusion data, achieving significant improvements in recall and F-measure scores. In [22], a GRU-based lightweight IDS model was proposed for real-time detection, highlighting efficiency gains through reduced complexity. In [23], a Deep Recurrent Neural Network (DRNN) was proposed for SDN-based networks, enabling the detection of sophisticated intrusion attempts with high accuracy on sequential network flows. Sparse Autoencoders were employed in [24], combining feature dimensionality reduction with Logistic Regression to deliver improved IDS performance. Finally, in [25], an AE was combined with Random Forest to form a robust IDS model. However, despite these advances, a unified framework that combines semantic embedding, dual-deep feature extraction (Bi-LSTM and Bi-GRU), and heterogeneous ensemble classification tailored for IoMT environments remains missing.

The key contributions of this study are:

- A dual-stage feature enhancement using semantic Word2Vec embedding.
- Hybrid temporal modeling using Bi-LSTM and Bi-GRU for contextual learning.
- A Heterogeneous Ensemble Learning (HEL) classifier for robust intrusion prediction across IoMT datasets.

II. PROPOSED SYSTEM

This research method, as shown in Figure 1, aimed to design a semantically improved deep feature ensemble-driven heterogeneous ensemble learning model for intrusion detection in IoMT (SIDHELNet). The novelty of SIDHELNet lies in its integration of semantic feature embedding with Bi-LSTM and Bi-GRU models, which allows enriched temporal pattern recognition. This dual-deep framework captures bidirectional dependencies, improving intrusion detection accuracy across diverse IoMT traffic patterns. The initial stage of the SIDHELNet method involves a dual-step preprocessing strategy designed to refine the IoMT network traffic data. First, outlier removal is performed to eliminate extreme and anomalous values that can skew feature distribution and learning outcomes. This is achieved by applying a statistical thresholding technique based on the mean and standard deviation of the dataset, where any data point that falls outside the range $\mu \pm \sigma$ is discarded. This ensures that only statistically consistent samples are retained for downstream processing. Following outlier filtering, active period segmentation is applied to isolate relevant traffic bursts from idle or noise periods in the data stream.

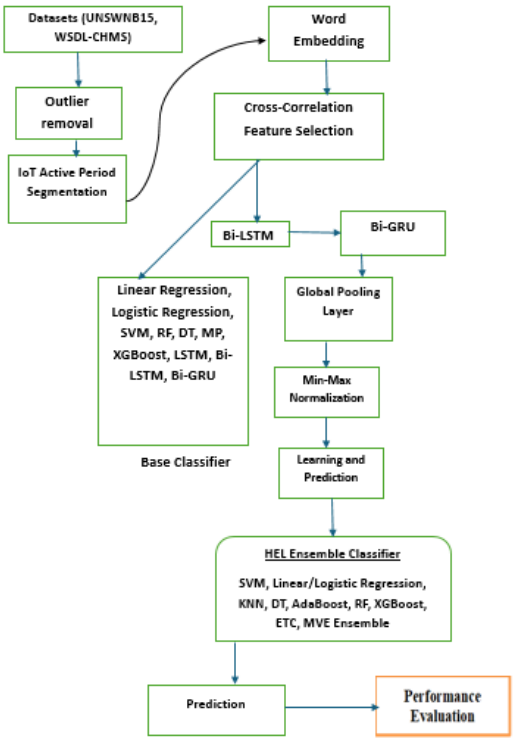

Fig. 1. Proposed method.

The WUSDL-EHMS-2020 dataset contains 10,000 samples with 44 features, where approximately 87.5% belong to normal traffic and 12.5% represent attack traffic [4]. The UNSW-NB15 dataset includes 175,341 samples with 49 features, consisting of 63.1% normal and 36.9% malicious instances spread across multiple attack categories [26-30]. The dataset also contains feature-level insight into the WUSTL EHMS-2020 dataset, categorizing attributes into node traffic metrics and patient-specific sensor data. Key transmission statistics, such as IP addresses, port numbers, jitter, packet sizes, delays, and inter-packet times, are captured alongside biometric parameters such as temperature, SpO_2, heart rate, and blood pressure.

All features were normalized using min-max scaling to the [0,1] range, and low-variance attributes were filtered out to retain informative dimensions. For class imbalance, targeted undersampling of dominant normal traffic was applied, preserving attack patterns through active period segmentation. Both datasets were split into 80% training and 20% testing sets using stratified sampling to maintain the original class distributions. Active period segmentation refers to isolating bursts of meaningful traffic by filtering out idle intervals, enabling the model to focus on relevant communication patterns indicative of normal or malicious behavior.

In the Semantic Feature Embedding Using Word2Vec phase, each data instance of IoMT traffic was semantically enriched using the Word2Vec model. By treating each traffic flow as a "sentence" and features as "words," Word2Vec captures contextual similarities between traffic patterns. A hybrid deep model combined Bi-LSTM and Bi-GRU layers to capture long-range dependencies in IoMT traffic. Bi-LSTM extracts bidirectional temporal features, while Bi-GRU accelerates training with simplified gating. This fusion ensures context-aware learning with reduced gradient issues and lower computational overhead. After segmentation, Word2Vec embeddings are generated and fed in parallel to the Bi-LSTM and Bi-GRU layers. Their outputs are concatenated into a unified feature vector that is passed to the HEL ensemble for final classification. Word2Vec was chosen for its lower computational cost and effectiveness in sparse IoMT traffic. Unlike BERT, it requires no pre-training on medical corpora and adapts better to mixed network-biometric features. The various gate elements and matching outputs can then be obtained using the configurations in LSTM using the following equations.

$$f_t = sigmoid\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right) \quad (1)$$

$$i_t = sigmoid(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \quad (2)$$

$$c_t = c_{t-1} \odot f_t + i_t \odot tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \quad (3)$$

$$O_t = sigmoid(W_{ox}x_t + W_{ch}h_{t-1} + b_0) \quad (4)$$

$$h_t = O_t tanh(c_t) \quad (5)$$

where $x_t \in R^n$ refers to the input vector, with $W \in R^{v*n}$, $b \in R^v$. The produced real-valued feature vectors are stacked in this way to produce an embedding matrix:

$$R = \begin{matrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ r_{3,1} & r_{3,2} & \cdots & r_{3,n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{v-1,1} & r_{v-1,2} & \cdots & r_{v-1,n} \\ r_{v,1} & r_{v,1} & \cdots & r_{v,n} \end{matrix} \quad (6)$$

The hidden layer yielding the output feature is defined as:

$$\vec{h}_t = LSTM(x_t, \vec{h}_{t-1}) \quad (7)$$

The input is processed by an LSTM designed for backward feature extraction from right to left, yielding the hidden layer information:

$$\overleftarrow{h}_t = LSTM(x_t, \overleftarrow{h}_{t+1}) \quad (8)$$

Thus, applying both sub-RNN components, the Bi-LSTM features are used to yield the contextual feature as:

$$h_{t,Bi-LSTM} = [\vec{h}_t, \overleftarrow{h}_t] \quad (9)$$

The model extracts features at various layers by utilizing the following equations:

$$z_t = \sigma(W^z x_t + U^z h_{-1}) \quad (10)$$

$$r_t = \sigma(W^r x_t + U^r h_{-1}) \quad (11)$$

$$\tilde{h}_t = tanh\left(W^h x_t + U^h(h_{-1} \odot r_t)\right) \quad (12)$$

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{-1} \quad (13)$$

where W and U state the GRU weights, and σ states the logical sigmoid function. In these equations, the update gate, denoted by z_t is employed to specify the degree of activation value associated with the functional GRU as shown in Figure 2, while $\odot$ represents the element-wise multiplication function. The proposed design of the network with two GRUs, as shown in Figure 3, works simultaneously to model traffic patterns toward the two directions of the feature sequence (13) for further learning and prediction.

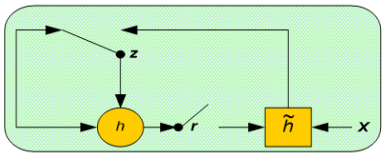

Fig. 2. The GRU architecture.

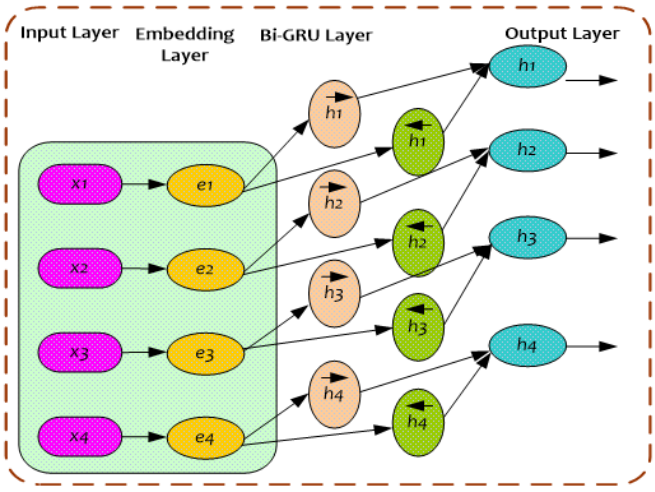

Fig. 3. The Bi-GRU architecture.

The models are used to provide features as follows:

$$\overleftarrow{h}_t = GRU(X_t, \overleftarrow{h}_{t+1}),\ t \in [1, L] \quad (14)$$

$$\vec{h}_t = GRU(X_t, \vec{h}_{t-1}),\quad t \in [L, 1] \quad (15)$$

$$H_{t\_Bi\_GRU} = [\vec{h}_t, \overleftarrow{h}_t] \quad (16)$$

$$H_{t\_Bi\_GRU} = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{-1} \quad (17)$$

In (14-16), the values of h_0 and h_{L+1} are initialized with zero vectors. Bi-LSTM captures long-term bidirectional dependencies, while Bi-GRU offers faster convergence and reduced computational load. In the SIDHELNet model, the instance feature representation vector V_s is produced by fusing the features obtained by Bi-LSTM and Bi-GRU. The resultant concatenated feature is:

$$H = concat\left(h_{t,Bi-LSTM}, H_{t\_Bi\_GRU}\right) \quad (18)$$

Consider B be the number of convolutional kernels in Bi-LSTM. The proposed model applies a GAP layer, averaging the input vector H (18) to derive a feature vector $V_s \in R^{2d}$ and avoid the overfitting problem.

$$V_s = GAP(H) \quad (19)$$

The final feature is obtained as:

$$V_s = concat\left(V_{s1,}\ V_{s2,\ldots,}V_{sN,}\right) \quad (20)$$

In SIDHELNet, the feature vector V_s is mapped as input to the output layer to perform learning and prediction by using the loss function:

$$Loss_{Function} = -\frac{1}{m}\sum_i^m \left(y_i * log\left(p(y_i)\right) + (1 - y_i) * log\left(1 - p(y_i)\right)\right) \quad (21)$$

where m signifies the total samples, y_i refers to the true labels, and $p(y_i)$ states the likelihood of the true labels. In this work, both deep models applied the Adam adaptive learning method with a learning rate of 0.0001.

To overcome inconsistences, a HEL model integrates multiple classifiers, each producing individual predictions. The final decision is derived through a majority voting or consensus-based aggregation of these outputs. The ensemble includes nine diverse base classifiers: SVM, Naïve Bayes, k-NN, Decision Tree, Random Forest, Extra Trees, XGBoost, Linear Regression, and Logistic Regression.

SVM is used due to its ability to construct optimal hyperplanes that separate different classes. The decision function used for multi-class intrusion prediction is defined as:

$$Y' = w * \phi(x) + b \quad (22)$$

where $\phi(x)$ is a nonlinear transform function and w and b are weights and bias values, respectively. In this case, the regression-risk parameter is decreased to quantify the output Y':

$$R_{reg}(Y') = C * \sum_{i=0}^{l} \gamma(Y_i' - Y_i) + \frac{1}{2} * \|w\|^2 \quad (23)$$

where C and γ refer to the penalty and cost functions, respectively. The w in (22-23) is derived by:

$$w = \sum_{j=1}^{l}\left(\alpha_j - \alpha_j^*\right)\phi(x_j) \quad (25)$$

where α and α^* refer to non-zero values.

The most probable class is given by:

$$Y' = \sum_{j=1}^{l}\left(\alpha_j - \alpha_j^*\right)\phi\left(x_j\right) * \phi(x) + b = \sum_{j=1}^{l}\left(\alpha_j - \alpha_j^*\right) * K\left(x_j, x\right) + b \quad (25)$$

where $K\left(x_j, x\right)$ signifies the kernel function. The class likelihood $e^* = argmax_d P(d|x)$ is applied to the x as per the Bayes′ rule:

$$P(d|x) = \frac{P(x|d)P(d)}{P(x)} \quad (26)$$

where the class probability is expressed as $P(d)$, and the likelihood of the data element x is expressed as $P(d|x)$. According to (27), the component $P(x)$ indicates the previous likelihood related to the predictor.

$$P(x|d) = \prod_{l=1}^{m} P(x_l|d) \quad (27)$$

In the proposed model, composite hybrid deep features from Bi-LSTM and Bi-GRU are recursively divided to form these nodes, enabling effective decision-making for classification. The input is denoted as x, and any noise in the data is represented as I. Equation (28) enhances information gain, with the samples in P_d, LC_d and RC_d.

$$InformationGain(P_d x) = I(P_d) - \frac{LC_n}{P_n} I(L.C_d) - \frac{RC_n}{P_n} I(R.C_d) \quad (28)$$

where I is obtained by using the Gini-Index I_G, entropy I_H, or the classification error I_E:

$$I_H(n) = -\sum_{i=1}^{c} p(c|n) \log_2 p(c|n) \quad (29)$$

$$I_G(n) = 1 - \sum_{i=1}^{c} p(c|n)^2 \quad (30)$$

$$I_E(n) = 1 - max\{p(c|n)\} \quad (31)$$

The HEL model integrates SVM, k-NN, Decision Tree, Random Forest, Extra Trees, XGBoost, Naïve Bayes, Logistic Regression, and Linear Regression. SIDHELNet employs a majority voting strategy, in which each base learner makes a prediction, and the class with the highest number of votes is selected as the final output. The base learners were selected for their diverse learning paradigms: SVM for margin maximization, k-NN for instance-based learning, trees for rule-based generalization, and logistic models for probabilistic interpretation, improving robustness via learner diversity.

When a labelled pattern is compared to an unlabeled pattern, k-NN can identify it by assigning a corresponding class label. In the deployed k-NN model, the Euclidean distance method was used to calculate:

$$D(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_n - q_n)^2} \quad (32)$$

The Random Forest algorithm uses multiple tree-based classifiers to perform voting-based learning and classification. The basis of the level of significance for the tree formation is:

$$\{h(x, \theta_k), k = 1,2, \ldots i \ldots\} \quad (33)$$

The data was divided into 80% data for training and 20% as out-of-bag samples for further inner cross-validation to retrieve the classification results.

Ensemble learning using the decision tree base classifier $X = \{x_i\}_{i=1}^{n}, x_i \in R^m$ yields the prediction labels $y = \{y_i\}_{i=1}^{n}, y_i \in \{\omega_j \in (1,2,\dots,c)\}$. The ensemble function can be defined as:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F \quad (34)$$

The CART space is defined as $F = \{f(x) = w_q(x)\}(q: R^m \to T, w \in R^T)$, where q refers to the tree's structure and T signifies the total leaves. Each tree f_k signifies a part of an autonomous q, where the leaf weight is given by w. A regularization function is used to calculate the set of functions designed for ensemble learning:

$$\ell = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k), \quad (35)$$

where $\Omega(f_k) = \xi T + \frac{1}{2}\xi \|w\|^2$, l states a differential convex loss function, while the prediction result is $\hat{y}_i$. $\Omega(f)$ defines the intricacy of the tree f_k, whereas ξT and $\xi \|w\|^2$ penalise the extreme weights and each involved tree, respectively.

$$\ell^{(t)} = \sum_{i=1}^{n} l\left(\hat{y}_i^{(t-1)} + f_t(x_i), y_i\right) + \Omega(f_t) \quad (36)$$

An objective function can be defined as in (37) by using the Taylor expansion function over (30), particularly with the 1st and 2nd order gradients of the cost function.

$$\ell^{(t)} \cong \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) \quad (37)$$

where:

$$g_i = \delta_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right) \text{ and } h_i = \delta^2_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right).$$

A decision tree classifier predicts fixed values within a leaf $w_q(x)$, where $q(x)$ maps the data instance x to a leaf and w signifies a score vector for each leaf, defining the tree $f_k(x)$. Extending the second term of (37) yields a sum over the tree leaves, where the regularization is found to be:

$$\ell^{(t)} \cong \sum_{j=1}^{T} \left[G_j w_i + \frac{1}{2}\left(H_j + \xi\right) w_j^2\right] + \lambda \mathrm{T},$$

$$\text{where } G_j = \sum_{j \in I_j} g_i, \; H_j = \sum_{j \in I_j} h_i \quad (38)$$

In (38), $I_j = \{i | q(x_i) = j|\}$ states the data point at leaf j. For a fixed structure tree, the cost function tries to reduce $\frac{\delta \ell^{(t)}}{\delta w_i} = G_j + \left(H_j + \lambda\right) w_i = 0$, and the optimal weight value of leaf node j is measured by using:

$$w^* = -\frac{G_j}{H_j + \xi} \quad (39)$$

Replacing the value of (39) into (36), the objective function to measure the best tree structure yields:

$$\ell^{(t)} = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \xi} + \xi T \quad (40)$$

During the m^{th} training round, XGBoost uses the DART booster by dropping k trees. The cost function is redefined as in (41) in case it is hypothesized that $D = \sum_{k \in K} F_k$:

$$\ell^{(t)} = \sum_{i=1}^{n} l\left(\hat{y}_i^{(m-1)} - D_i + \tilde{F}_m, y_i\right) + \Omega\left(\tilde{F}_m\right) \quad (41)$$

where the overshooting parameters D and $\tilde{F}_m$ are normalized. Tree- and forest-based normalization methods are supported by XGBoost. The following equation was used to perform a linear regression over the chosen characteristics.

$$logit[\pi(x)] = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots \dots . + \beta_m X_m \quad (42)$$

with x_i being the independent variable, or the final characteristics chosen for each traffic data, and the dependent variable is $logit[\pi(x)]$. The acquisition of each query's incursion probability is given as:

$$\pi(x) = \frac{e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots \dots + \beta_m X_m}}{1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots \dots + \beta_m X_m}} \quad (43)$$

In this manner, Logistic Regression classifies every input query as Normal Traffic (i.e., 0) or Intrusion (1) by using the above-derived probability function. Once the predicted score is obtained using each of the base ensemble classifiers, the proposed method applies the MVE ensemble technique or the consensus-driven decision model to predict intrusion in IoMT traffic and the type of attack(s).

III. RESULTS AND ANALYSIS

SIDHELNet enhances feature informativeness and classification robustness through active period segmentation, Word2Vec embedding, hybrid RNN feature extraction (Bi-LSTM and Bi-GRU), and HEL. The proposed method was evaluated on two benchmark datasets, WUSDL-EHMS-2020 (binary classification) and UNSW-NB15 (multi-class classification). SIDHELNet outperformed both traditional ML/DL models, achieving up to 99.77% accuracy with a minimal error rate (MAE = 0.0064). In an ablation study, removing semantic embedding reduced accuracy to 96.02%, excluding Bi-GRU dropped it to 96.77%, and omitting the HEL ensemble resulted in 95.31%, highlighting each component's importance in enhancing overall detection performance. Table I shows the performance of various models on the WSDL-EHMS-2020 dataset. Traditional ML methods performed well, with XGBoost achieving 95.2% accuracy. However, SIDHELNet outperformed, achieving 97.58% accuracy and a near-perfect F-measure of 0.99. Figure 4 visualizes the results on WSDL-EHMS-2020, confirming SIDHELNet's superior precision and recall across compared models.

TABLE I. ASSESSMENT ON WSDL-EHMS-2020

Technique	Model	Accuracy (%)	Precision (%)	Recall (%)	F-score
ML methods	Linear Regression	93.00	92.00	99.00	0.96
	Logistic Regression	93.00	92.00	99.00	0.96
	SVM	92.83	92.01	97.00	0.97
	k-NN	92.00	93.00	99.00	0.96
	Random Forest	94.8	93.00	100.00	0.96
	Decision Tree	94.00	97.00	97.00	0.97
	ETC	94.80	92.00	100.0	0.96
	XGBoost	95.20	93.00	100.0	0.96
DL methods	LSTM	93.16	93.92	97.00	0.95
	Bi-LSTM	94.44	94.21	94.47	0.94
	Bi-GRU	93.92	93.11	98.71	0.97
HEL	SIDHELNet	97.58	97.99	99.21	0.99

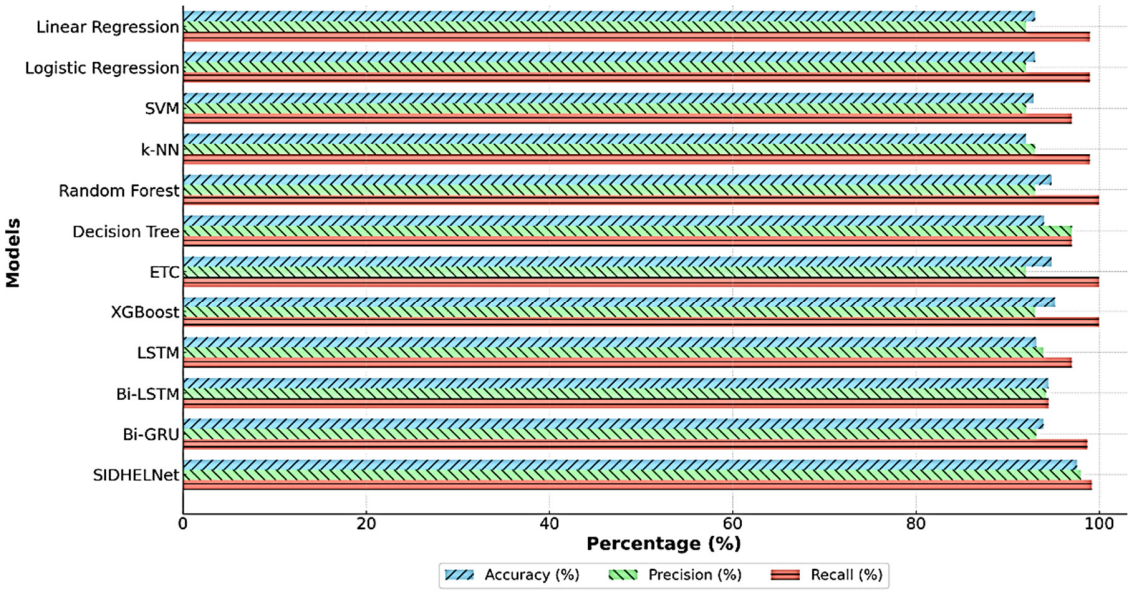

Fig. 4. Simulation results on the WUSDL-EHMS-2020 dataset.

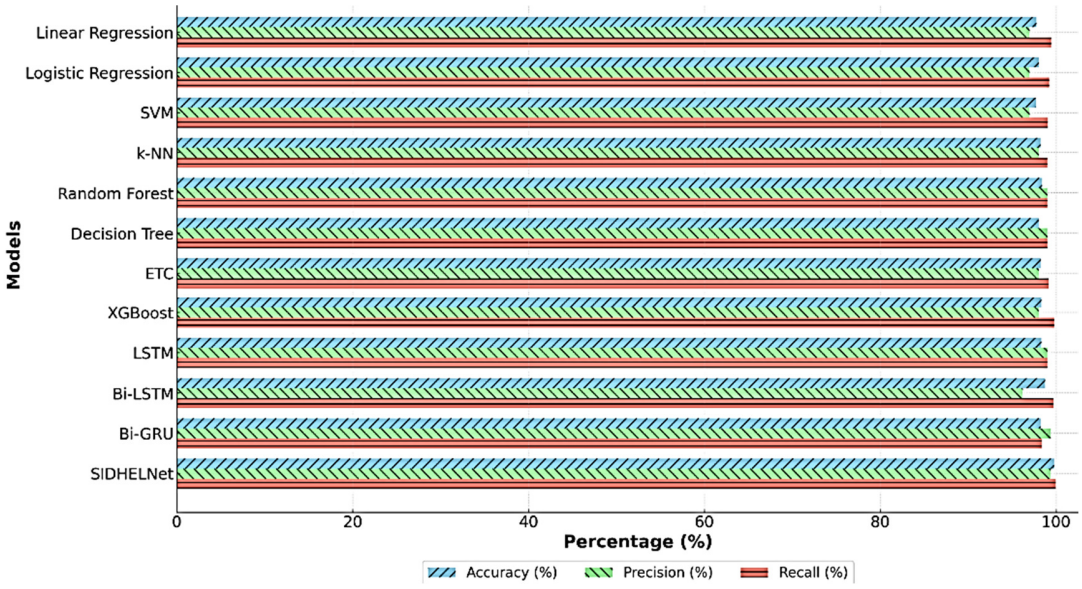

Fig. 5. Simulation results on the UNSW-NB15 dataset

In 5-fold cross-validation, SIDHELNet achieved average accuracies of 97.41% on WUSDL-EHMS-2020 and 99.62% on UNSW-NB15, confirming its consistent performance across multiple data subsets. Stratified sampling, early stopping, and 5-fold cross-validation were applied to mitigate overfitting and class imbalance, confirming that SIDHELNet's high accuracy reflects genuine learning rather than dataset bias.

The results on UNSW-NB15 (Table II) show high accuracy for all models, with XGBoost and Bi-LSTM achieving 98.32% and 98.76%, respectively. SIDHELNet once again led with 99.77% accuracy and 99.94% recall, demonstrating strong generalizability. Figure 5 also shows that SIDHELNet outperformed others in all evaluation metrics on UNSW-NB15.

Table III summarizes the prediction error metrics. Although models such as XGBoost and Bi-LSTM performed well, SIDHELNet recorded the lowest MAE, MSE, and RMSE on both datasets, confirming its high prediction precision and minimal error rates. Figure 6 provides a clear visual comparison of the prediction errors, where SIDHELNet exhibits the lowest values, validating its efficiency and stability. The ensemble's performance boost stems from model complementarity, where diverse classifiers capture varying decision boundaries, enabling the HEL model to generalize better across heterogeneous IoMT attack patterns.

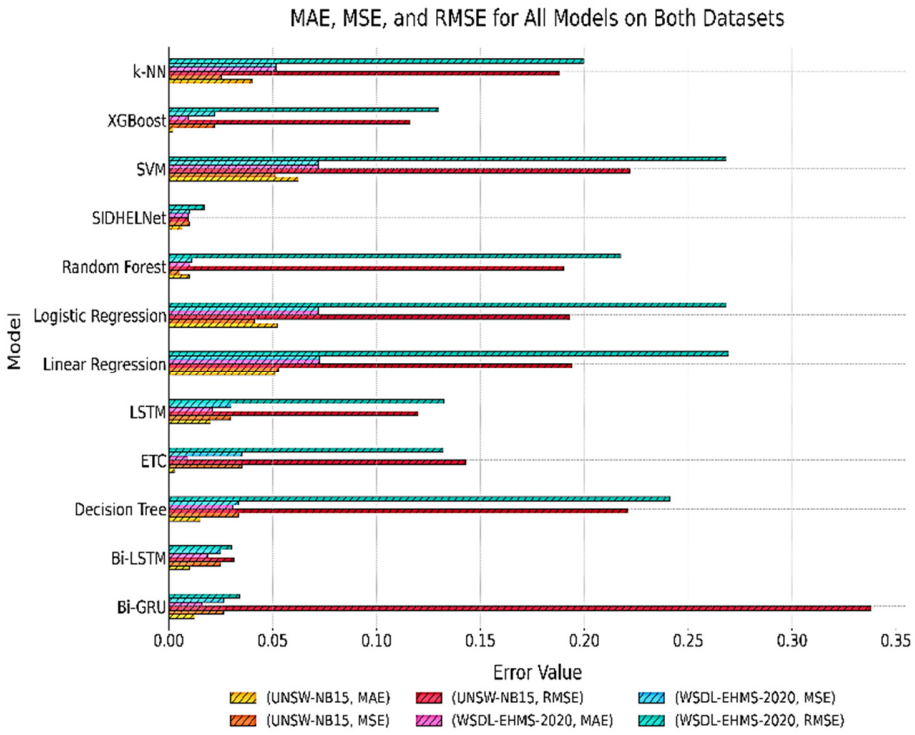

Fig. 6. Error values plot.

TABLE II. ASSESSMENT ON UNSW-NB15

Technique	Model	Accuracy (%)	Precision (%)	Recall (%)	F-score
ML methods	Linear Regression	97.78	97.00	99.44	0.99
	Logistic Regression	98.00	97.00	99.21	0.99
	SVM	97.72	97.00	99.07	0.99
	k-NN	98.21	98.00	99.00	0.99
	Random Forest	98.41	99.00	99.00	0.99
	Decision Tree	98.00	99.00	99.00	0.99
	ETC	98.27	98.00	99.14	0.99
	XGBoost	98.32	98.00	99.77	0.99
DL methods	LSTM	98.32	99.00	99.00	0.99
	Bi-LSTM	98.76	96.21	99.69	0.98
	Bi-GRU	98.25	99.38	98.36	0.99
HEL Ensemble	SIDHELNet	99.77	99.39	99.94	0.99

Table VII compares SIDHELNet with 21 existing methods. Although some models showed strong individual metrics, SIDHELNet consistently achieved the best overall results with 99.77% accuracy, 99.39% precision, and 99.60% F-score, proving its superiority in comprehensive intrusion detection.

As shown in Figure 7, SIDHELNet achieved high classification precision, with only 15 false negatives and 20 false positives out of 10,000 samples. This low error distribution confirms the reliability and suitability of the model for critical IoMT environments where misclassification costs are high. During experimentation, handling class imbalance in WUSDL-EHMS-2020 required active period segmentation and threshold filtering. Additionally, while the HEL model improves accuracy, it increases training time, which was mitigated by using feature fusion and early stopping. The average inference latency of SIDHELNet was 13.2 ms per sample, indicating its suitability for real-time intrusion detection in IoMT systems with minimal delay. SIDHELNet required approximately 2.4 GB of memory during training on an NVIDIA GTX 1660 GPU and used ~38% CPU during inference, supporting its deployment in resource-constrained medical networks.

TABLE III. MODEL PERFORMANCE ASSESSMENT (WITHOUT DUPLICATES)

Reference	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
[5]	95.40	96.20	93.50	95.75
[6]	78.50	81.00	78.50	76.50
[7]	99.84	99.94	98.81	99.37
[8]	99.59	-	-	-
[9]	82.14	87.28	67.89	76.37
[10]	95.60	-	-	-
[11]	84.98	-	95.96	-
[12]	98.43	-	-	-
[13]	85.24	-	-	-
[14]	74.19	-	-	90.26
[15]	99.01	96.71	98.58	97.64
[16]	99.56	99.52	99.55	-
[17]	99.10	99.07	99.10	99.02
[18]	84.25	-	-	-
[19]	97.01	100.00	98.48	97.01
[20]	98.48	100.00	96.10	98.20
[21]	95.14	-	-	-
[22]	98.07	97.06	99.22	98.13
[23]	99.98	-	-	-
[24]	93.53	57.45	31.87	41.00
[25]	99.83	99.00	99.00	99.00
SIDHELNet	99.77	99.39	99.94	99.60

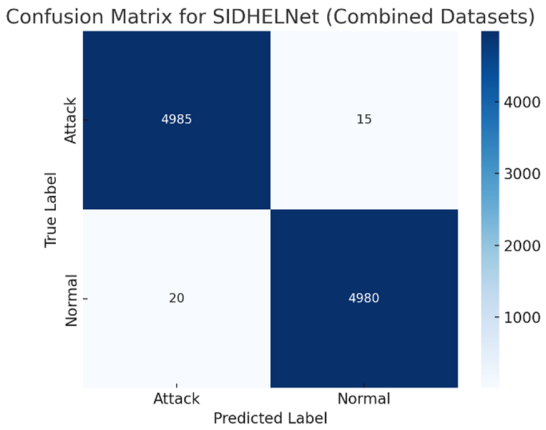

Fig. 7. Confusion matrix.

IV. CONCLUSION

Rapid advancement of IoT technologies has led to transformative applications in domains such as e-healthcare, surveillance, and automation. Among these, IoMT plays a crucial role by enabling continuous patient monitoring, remote diagnosis, and personalized care delivery. However, many existing intrusion detection models focus on isolated intrusion types, limiting their scalability and adaptability in heterogeneous IoMT environments. To overcome these challenges, this research proposes a robust, semantically enriched deep feature ensemble model using heterogeneous ensemble learning, called SIDHELNet, for multi-type intrusion detection in IoMT. Bi-LSTM and Bi-GRU models extract complementary temporal features, which are fused and normalized to prevent overfitting. A HEL classifier combines SVM, Logistic Regression, k-NN, Decision Tree, AdaBoost, Random Forest, and XGBoost to perform final classification through majority voting. SIDHELNet significantly outperforms conventional models, achieving 99.77% accuracy, 99.39% precision, 99.94% recall, and a 99.60% F-score. The novelty of SIDHELNet lies in its synergistic integration of semantic feature embedding (via Word2Vec) with a dual deep architecture (Bi-LSTM and Bi-GRU), followed by a HEL classifier. Unlike existing approaches that rely solely on single deep models or traditional ensemble techniques, SIDHELNet combines semantic understanding with bidirectional temporal learning and diversified classifier consensus. Comparative analysis with baseline models, including Linear and Logistic Regression, SVM, Random Forest, XGBoost, LSTM, Bi-LSTM, and Bi-GRU, confirms the proposed model's superior accuracy, reliability, and applicability in real-world IoMT security systems. SIDHELNet's use of semantic embedding and ensemble diversity allows it to adapt to varying device behaviors and novel attack patterns, supporting generalization across evolving IoMT threat landscapes. As a future extension, we aim to simulate adversarial noise and perform robustness analysis under data perturbation to evaluate the resilience of SIDHELNet in real-world deployment conditions. We also plan to enhance SIDHELNet by integrating continuous learning and domain-adaptive transfer learning to ensure sustained performance in dynamically evolving IoMT ecosystems.

REFERENCES

[1] H. Wang, Y. Wang, X. Xie, and M. Li, "A Scheduling Scheme for Minimizing Age Under Delay Tolerance in IoT Systems With Heterogeneous Traffic," *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 16902–16914, Feb. 2024, https://doi.org/10.1109/JIOT.2024.3366766.

[2] N. K. Al-Shammari, T. H. Syed, and M. B. Syed, "An Edge – IoT Framework and Prototype based on Blockchain for Smart Healthcare Applications," *Engineering, Technology & Applied Science Research*, vol. 11, no. 4, pp. 7326–7331, Aug. 2021, https://doi.org/10.48084/etasr.4245.

[3] CH. Nagaraju, M. Khan, S. Fahimuddin, and S. Karimullah, "Time-Correlated Fading Channels for Minimization of Energy Consumption in Wireless Networked Control Systems," in *Proceedings of the Third International Conference on Cognitive and Intelligent Computing, Volume 2*, 2025, pp. 617–625, https://doi.org/10.1007/978-981-97-9266-5_60.

[4] "WUSTL EHMS 2020 Dataset for Internet of Medical Things (IoMT) Cybersecurity Research." [Online]. Available: https://www.cse.wustl.edu/~jain/ehms/index.html.

[5] Z. Liu *et al.*, "Deep Learning Approach for IDS," in *Fourth International Congress on Information and Communication Technology*, 2020, pp. 471–479, https://doi.org/10.1007/978-981-15-0637-6_40.

[6] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, https://doi.org/10.1109/ACCESS.2019.2895334.

[7] A. Thakkar and R. Lohiya, "Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System," *Information Fusion*, vol. 90, pp. 353–363, Feb. 2023, https://doi.org/10.1016/j.inffus.2022.09.026.

[8] A. M. Aleesa, M. Younis, A. A. Mohammed, and N. M. Sahar, "Deep-Intrusion Detection System with Enhanced UNSW-NB15 Dataset Based on Deep Learning Techniques," *Journal of Engineering Science and Technology*, vol. 16, no. 1, pp. 711–727, 2021.

[9] C. Tang, N. Luktarhan, and Y. Zhao, "SAAE-DNN: Deep Learning Method on Intrusion Detection," *Symmetry*, vol. 12, no. 10, Oct. 2020, Art. no. 1695, https://doi.org/10.3390/sym12101695.

[10] L. Ashiku and C. Dagli, "Network Intrusion Detection System using Deep Learning," *Procedia Computer Science*, vol. 185, pp. 239–247, Jan. 2021, https://doi.org/10.1016/j.procs.2021.05.025.

[11] P. Wu and H. Guo, "LuNet: A Deep Neural Network for Network Intrusion Detection," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, Xiamen, China, Dec. 2019, pp. 617–624, https://doi.org/10.1109/ssci44817.2019.9003126.

[12] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Information Sciences*, vol. 513, pp. 386–396, Mar. 2020, https://doi.org/10.1016/j.ins.2019.10.069.

[13] C. Liu, Z. Gu, and J. Wang, "A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning," *IEEE Access*, vol. 9, pp. 75729–75740, 2021, https://doi.org/10.1109/ACCESS.2021.3082147.

[14] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," *Computer Communications*, vol. 199, pp. 113–125, Feb. 2023, https://doi.org/10.1016/j.comcom.2022.12.010.

[15] J. Figueiredo, C. Serrão, and A. M. de Almeida, "Deep Learning Model Transposition for Network Intrusion Detection Systems," *Electronics*, vol. 12, no. 2, Jan. 2023, Art. no. 293, https://doi.org/10.3390/electronics12020293.

[16] V. Rajasekar and S. Sarika, "An efficient intrusion detection model based on recurrent neural network," in *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, 2022.

[17] V. Hnamte, H. Nhung-Nguyen, J. Hussain, and Y. Hwa-Kim, "A Novel Two-Stage Deep Learning Model for Network Intrusion Detection:

LSTM-AE," *IEEE Access*, vol. 11, pp. 37131–37148, 2023, https://doi.org/10.1109/ACCESS.2023.3266979.

[18] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, "BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset," *IEEE Access*, vol. 8, pp. 29575–29585, 2020, https://doi.org/10.1109/ACCESS.2020.2972627.

[19] S. Sivamohan, S. S. Sridhar, and S. Krishnaveni, "An Effective Recurrent Neural Network (RNN) based Intrusion Detection via Bi-directional Long Short-Term Memory," in *2021 International Conference on Intelligent Technologies (CONIT)*, Hubli, India, Jun. 2021, pp. 1–5, https://doi.org/10.1109/conit51480.2021.9498552.

[20] B. Roy and H. Cheung, "A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, Sydney, Nov. 2018, pp. 1–6, https://doi.org/10.1109/atnac.2018.8615294.

[21] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Systems with Applications*, vol. 185, Dec. 2021, Art. no. 115524, https://doi.org/10.1016/j.eswa.2021.115524.

[22] H. Ma, J. Cao, B. Mi, D. Huang, Y. Liu, and S. Li, "A GRU-Based Lightweight System for CAN Intrusion Detection in Real Time," *Security and Communication Networks*, vol. 2022, no. 1, 2022, Art. no. 5827056, https://doi.org/10.1155/2022/5827056.

[23] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, Montreal, Canada, Jun. 2018, pp. 202–206, https://doi.org/10.1109/netsoft.2018.8460090.

[24] A. S. Qureshi, A. Khan, N. Shamim, and M. H. Durad, "Intrusion detection using deep sparse auto-encoder and self-taught learning," *Neural Computing and Applications*, vol. 32, no. 8, pp. 3135–3147, Apr. 2020, https://doi.org/10.1007/s00521-019-04152-6.

[25] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, Oct. 2018, https://doi.org/10.1109/TETCI.2017.2772792.

[26] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, Nov. 2015, pp. 1–6, https://doi.org/10.1109/milcis.2015.7348942.

[27] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1–3, pp. 18–31, Apr. 2016, https://doi.org/10.1080/19393555.2015.1125974.

[28] N. Moustafa, J. Slay, and G. Creech, "Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks," *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 481–494, Sep. 2019, https://doi.org/10.1109/TBDATA.2017.2715166.

[29] N. Moustafa, G. Creech, and J. Slay, "Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models," in *Data Analytics and Decision Support for Cybersecurity: Trends, Methodologies and Applications*, I. Palomares Carrascosa, H. K. Kalutarage, and Y. Huang, Eds. Springer International Publishing, 2017, pp. 127–156.

[30] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems," in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer International Publishing, 2021, pp. 117–135.