



ISSN ONLINE: 2447-0228

ITEGAM-JETIA

Manaus, v.12 n.61, p.62-76. September/October, 2026.

DOI: <https://doi.org/10.5935/jetia.v12i61.3311>

RESEARCH ARTICLE

OPEN ACCESS

HYBRIDNET-IDS: AN INNOVATIVE CNN, BILSTM, AND TRANSFORMER FRAMEWORK UTILISING ATTENTION FOR IMPROVED NETWORK INTRUSION DETECTION

Subhodip Koley¹, Sumit Das^{*2}, Debasish Saha Roy³, Jayashree Dhara⁴, Shibani Debnath⁵,
Jaya Bhattacharjee⁶, and Rudra Samanta⁷

^{1, 4, 5, 6, 7}Department of Computer Application, JIS College of Engineering, Block A, Phase III, PO. Kalyani, Nadia, West Bengal, India.

²Department of Information Technology, JIS College of Engineering, Block A, Phase III, PO. Kalyani, Nadia, West Bengal, India.

³Department of Computer Science and Engineering, JIS College of Engineering, Block A, Phase III, PO. Kalyani, Nadia, West Bengal, India.

¹<https://orcid.org/0009-0004-0262-0526>, ²<https://orcid.org/0000-0002-8018-9151>, ³<https://orcid.org/0009-0004-1063-2906>,

⁴<https://orcid.org/0009-0002-4831-0924>, ⁵<https://orcid.org/0009-0005-9934-3090>, ⁶<https://orcid.org/0009-0001-3370-7622>,

⁷<https://orcid.org/0009-0005-2527-6198>

Email: sumit.das@jiscollege.ac.in

ARTICLE INFO

Article History

Received: January 14, 2026

Reviewed: July 18, 2026

Accepted: August 18, 2026

Published: September 30, 2026

Keywords:

Intrusion Detection System,
HybridNet-IDS,
CNN-BiLSTM-Transformer,
Cyberattack Detection,
Explainable AI

ABSTRACT

The rapid spread of network-connected technologies and the growing complexity of attacks have shown that traditional signature-based intrusion detection systems are not good enough, especially when it comes to finding zero-day attacks and advanced persistent threats. This paper introduces HybridNet-IDS, an augmented deep learning framework that integrates Convolutional Neural Networks, Bidirectional Long Short-Term Memory, and Transformer architectures to collectively capture spatial, temporal, and long-range dependencies in network traffic. A hybrid feature selection method that combines Mutual Information, Extra Trees significance, and Recursive Feature Elimination reduces the feature space from 42 to 20 discriminative characteristics. We use the Optuna framework and a Tree-structured Parzen Estimator (TPE) sampler to find the best model settings through Bayesian hyperparameter optimisation. Tests on the UNSW-NB15 benchmark dataset show that HybridNet-IDS outperforms six optimised classical machine learning baselines. It has a Matthews Correlation Coefficient of 0.8085, a ROC-AUC of 97.69%, and an accuracy rate of 90.36%. Five-fold cross-validation and bootstrap resampling confirm statistical robustness and generalisation stability. Also, explainable AI-based analysis shows clearly how each feature affects decisions, which makes people more trusting and open. The results show that HybridNet-IDS is a good, optimised, and easy-to-understand choice for current intrusion detection systems.



Copyright ©2026 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

Network security is a big problem around the world because there are so many more devices that connect to networks and cyberattacks are getting smarter. According to Statista, there will be over 29 billion Internet of Things (IoT) devices by 2030. This will make it much easier for hackers to get into more places [1]. According to the IBM Cost of a Data Breach Report 2023, the average cost of a data breach around the world is now USD 4.45 million. This is a 15% increase from the last three years [2]. ENISA's Threat Landscape 2023 also says that there have been a lot more distributed denial-of-service attacks and advanced persistent threats that can stay hidden on business networks for months [3]. Standard intrusion detection systems that use signatures still work against known attack patterns, but they can't find polymorphic malware or zero-day vulnerabilities. Because of these rules, people are using machine learning and deep learning techniques to look at complex traffic patterns and find cyber threats that have never been seen before. As more and more devices are connected to the internet and cyber-attacks get more sophisticated, businesses all over the world are making network security a top priority. There are more kinds and amounts of threats than ever before to modern business networks.

These threats include distributed denial-of-service attacks and more advanced persistent threats that may not be noticed for a long time. Traditional intrusion detection systems that use signatures work well against known attack patterns, but they don't work well against zero-day vulnerabilities and polymorphic malware that keeps changing to avoid detection. Because of this basic flaw, the cybersecurity industry has started using machine learning and deep learning techniques that can find new types of attacks that have never been seen before by learning complicated patterns from network traffic data. There have been big improvements in using AI for network security, but there are still some big problems that haven't been fully solved in the literature. Most deep learning-based intrusion detection systems only use one architecture model at a time. This means that they can only find either spatial correlations between features or temporal dependencies between network flows. There are many different things that make up network traffic. For instance, packets show spatial correlations between the header fields and the payload contents, as well as temporal patterns throughout connection sequences. Second, network traffic data usually has 40 or more features, which makes it harder to work with and more likely to fit too closely to the data. But systematic feature selection methods are not often used in deep learning pipelines. Third, people often choose hyperparameters for deep neural networks by hand or using grid search. These methods don't use new Bayesian optimisation methods that can quickly find their way through complex parameter spaces. This study presents HybridNet-IDS, a novel attention-enhanced deep learning framework that synergistically integrates three complementary neural network topologies to address these limitations.

The Convolutional Neural Network branch uses a series of convolution operations to get hierarchical spatial features. This lets it find connections between different parts of network traffic. The Bidirectional Long Short-Term Memory branch processes sequential data in both directions, which lets it see patterns over time in their entirety. The Transformer branch uses multi-head self-attention to model long-range dependencies and feature interactions without having to process them in order like recurrent architectures do. These three branches come together to make a rich feature representation that is then sent through a classification head to find intrusions. The main contributions of this study are: (1) The author suggests HybridNet-IDS, a new hybrid deep learning architecture that uses parts from CNN, BiLSTM, and Transformer to learn spatial-temporal features for network intrusion detection. (2) The writer makes a weighted ensemble feature selection method that uses Mutual Information, Extra Trees, and Recursive Feature Elimination. This method reduces the number of features from 42 to 20 while still allowing you to tell the difference between them. The author tests the UNSW-NB15 [4] benchmark dataset a lot and gets 90.36% accuracy, 97.69% ROC-AUC, and 0.8085 Matthews Correlation Coefficient. The mean accuracy is 91.53% when using five-fold cross-validation. (4) The author does a thorough comparison with six well-known baseline approaches, which shows that the framework works well in practice. The author employs Optuna-based Bayesian optimisation to methodically modify hyperparameters, determining that the learning rate is the most critical factor, with approximately 80% relevance.

II. LITERATURE REVIEW

II.1 CLASSICAL MACHINE LEARNING TECHNIQUES

Classic machine learning methods have proven effective in detecting network breaches due to their simplicity, comprehensibility, and practical efficacy. Numerous studies have employed models such as Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression, Decision Trees, and Gradient Boosting to accurately differentiate between genuine and fraudulent network traffic. For example, [5] and [6] demonstrated that ensemble models such as Random Forest and XGBoost outperform single classifiers in detecting DDoS attacks and multi-class intrusions, particularly when trained on the CIC-DDoS2019 and NSL-KDD datasets. Demonstrated that employing feature selection techniques alongside predictive models such as Support Vector Machines and Naïve Bayes can facilitate dimensionality reduction and enhance detection accuracy [7]. Also used Spark-based setups to make machine learning detection models that can grow [8]. This shows how well the Random Forest method works with high-throughput network data. When calibrated correctly, tree-based models had great recall and accuracy for all types of assaults, as shown in [9] and [10]. These methods make it easier to understand and speed up the training process, but they often limit how flexible they are when it comes to new attack methods because they depend on manually created features. Also, as shown in [11] and [12], machine learning models might not work well when the network dynamics or datasets are not balanced. Although such challenges exist, classical machine learning is still a vital component of hybrid or ensemble- Based systems for intrusion detection because it is reliable and works well.

II.2 APPROACHES IN DEEP LEARNING

Deep learning (DL) models have attracted considerable interest in network intrusion detection for their capacity to autonomously derive hierarchical feature representations from raw traffic data. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and their numerous adaptations have been extensively employed for the effective modeling of intricate spatial and temporal correlations in network traffic. Introduced CNN-based architectures that exhibited enhanced accuracy and robustness in IoT intrusion scenarios by proficiently extracting spatial traffic characteristics [13] and [14]. The bidirectional LSTM (BiLSTM) and hybrid CNN-LSTM architectures detailed in [15] and [16] effectively modeled sequential dependencies, facilitating the detection of covert assaults occurring over time. Many studies, such as those in [17] and [18], utilized attention mechanisms with transformer layers to effectively model long-range dependencies. This resulted in improved recall and precision metrics on intricate datasets such as UNSW-NB15 and CICIDS2017. Author was about making deep learning models that are easier to understand and more reliable for analysts by using attention visualizations [19]. Also, [20] used autoencoders and generative methods to help find unusual things by rebuilding normal patterns and looking for things that don't fit them. Deep learning models are very powerful, but they sometimes need a lot of computing power and large datasets with labels, which can make them hard to use in real-time applications or in systems that need to be flexible and work in changing environments [12], [21]. Still, deep learning's ability to learn from data and adapt to new attacks has made it an important part of modern intrusion detection systems, especially when used with feature selection or ensemble learning techniques. Table 1 is a case study of past research. It shows the name of the dataset, how many features it has, the algorithms used, the accuracy scores for each publication, the main contributions, and the problems with the dataset.

Table 1: Case study of previous research paper.

Paper Ref.	Dataset	Number of Features	Algorithm	Accuracy (%)	Key Contribution	Limitations
[5]	Cybercrime Incident Dataset	7	RF, DT, LR, SVM, KNN, NB	84.87% (RF)	A comparative machine learning investigation on compact cybercrime characteristics, perpetrator identification, and attack technique forecasting.	Poor attributes, not centered on network traffic, regional data, nothing time-series analysis.
[6]	UNSW-NB15	49	RF, XGBoost	89.43% (RF), 90.12% (XGBoost)	ML-powered DDoS categorization with modification, and a hybrid prediction approach to DDoS.	Class mismatch, inadequate real-time validation, and high false positive rates for particular subsets.
[13]	Cyberbullying Dataset	25	Random Forest	99.83%	Shows that it can accurately classify behavioral cyber data using machine learning and is a good model for detecting network attacks..	Not designed for packet-level IDS; restricted to binary classification contexts.
[7]	Intrusion Detection Benchmark	41	Deep Q-Network (DQN)	97.4%	Reinforcement learning for flexible attacks identification; optimization-focused detection for medical IoT.	Training is quite hard, and blockchain integration is not an option.
[8]	Bot-IoT	35	RF, DT, SVM, ANN	99.20% (RF)	Detecting large-scale IoT botnets and classifying all types of IoT attacks.	There is a serious class imbalance and no true hardware validation.
[9]	Power-Grid Cyberattack Dataset	48	SVM, ANN (multi-agent)	99.88% (best agent)	Distributed machine learning for cyber-physical security; detection of two-stage voltage regulation attacks.	Dataset particular to a domain, limited to the environment of the distribution system.
[14]	CIC-DDoS2019	69	CNN, DL models	98.76%	DL-based DDoS detection with fewer features and hybrid deep learning with better FS.	Sensitive to prior processing, hybrid models are highly complex.
[15]	Banking Network Dataset	112	RF, SVM, KNN	96.21% (RF)	Machine learning-based DDoS detection for financial systems and a DDoS monitoring system for the banking industry.	The dataset does not include real-time data and has not been tested on actual traffic.
[16]	Survey (SDN & Smart Grid)	5	GC-LSTM	96.45%	Attack graph concept for early detection in smart grids.	Concentrates on finding things, not stopping them from happening. It is more about finding things than stopping them.
[10]	IDS Benchmark Dataset	78	CNN-LSTM	95.6%	Hybrid DL model for classifying attacks and analyzing internet traffic DDoS security.	Costs a lot of computing power, and there aren't any common IoT programs that can validate it.

[11]	NSL-KDD, UNSW-NB15, CAIDA	41–49	RF, ANN, PCA-RNN	99.88%	A meta-analysis of ML/DL IDS models and computational models of bandwidth exhaustion.	Different ways to evaluate, and using a dataset that is quite old.
[17]	IDS Benchmark Dataset	43	Autoencoder, DNN	94.1%	DL-based identification of anomalies and Mutual Information-based attribute significance.	Needs a lot of training data; Wrapper approaches with NN haven't been looked into.
[12]	NSL-KDD	42	DL-based IDS	98.45%	Better preprocessing and DL pipeline, created a Dynamic Feature Selector (DFS).	The dataset is old and doesn't include reinforcement learning.
[19]	UNSW-NB15	49	DNN, CNN, LSTM	93.8%	DL models for current infiltration trends and a real-time forecasting system for IoT.	Risk of dataset shift: system logs and user activity data were not taken into account.
[18]	UNSW-NB15	49	DT, RF, SVM, LR	89.77% (DT)	Comparing ML models on the same dataset to find early signs of a problem.	reduced SVM performance, high chance of overfitting because of 100% correctness.
[20]	Modified CIC-DDoS2019	82	NB, LR, RF, AdaBoost	99% (NB/LR)	Demonstrates that binary categorization enhances accuracy, employs comparative machine learning analysis for DDoS mitigation.	Less diverse attacks, a reactive strategy, and a concentration on damage management.
[21]	Multi-source Cybersecurity Dataset	47	LR, RF, RL	92.3%	ML and RL for flexible cyber protection, and AI's involvement in cryptocurrency platforms that work together.	A high rate of false negatives for fraud.

Source: Authors, (2026).

II.3 RESEARCH INSUFFICIENT AND MOTIVATION

A lot of research has been done on how to find cyberattacks, but there are still some big gaps in the current body of research. Traditional machine learning techniques work well on benchmark datasets, but they depend a lot on features that people make and don't work well when attack patterns change. Deep learning models enhance the representation of features; however, the majority of studies utilize only one or two architectures and fail to simultaneously capture spatial, temporal, and long-range relationships within network data. Additionally, a lot of research focuses on accuracy and not so much on statistical validation, reducing the number of features, and changing the hyperparameters. The reliance on the dataset and the absence of cross-validation further constrain the ability to generalize. These limitations necessitate the creation of a unified, attention-augmented hybrid framework that amalgamates diverse learning methodologies with systematic feature selection and optimal training to facilitate robust and generalizable intrusion detection.

III. MATERIALS AND METHODS

The objective of the proposed HybridNet-IDS is to provide accurate and robust intrusion detection by integrating the learning of geographical, temporal, and long-range relationships within network data. Figure 1 shows that the system has five parts: data preparation, hybrid feature selection, Bayesian hyperparameter optimization, a hybrid CNN–BiLSTM–Transformer model, and exhaustive evaluation. Together, these parts help with optimal and generalizable intrusion classification. The UNSW-NB15 [4] dataset has 257,673 records and 44 characteristics. It is pre-processed by filling in missing values with the median, encoding categorical features with labels, replacing infinite values, and removing duplicate entries. The mean of numerical characteristics is set to zero. After preprocessing, 162,745 samples with 42 relevant characteristics are kept for model construction.

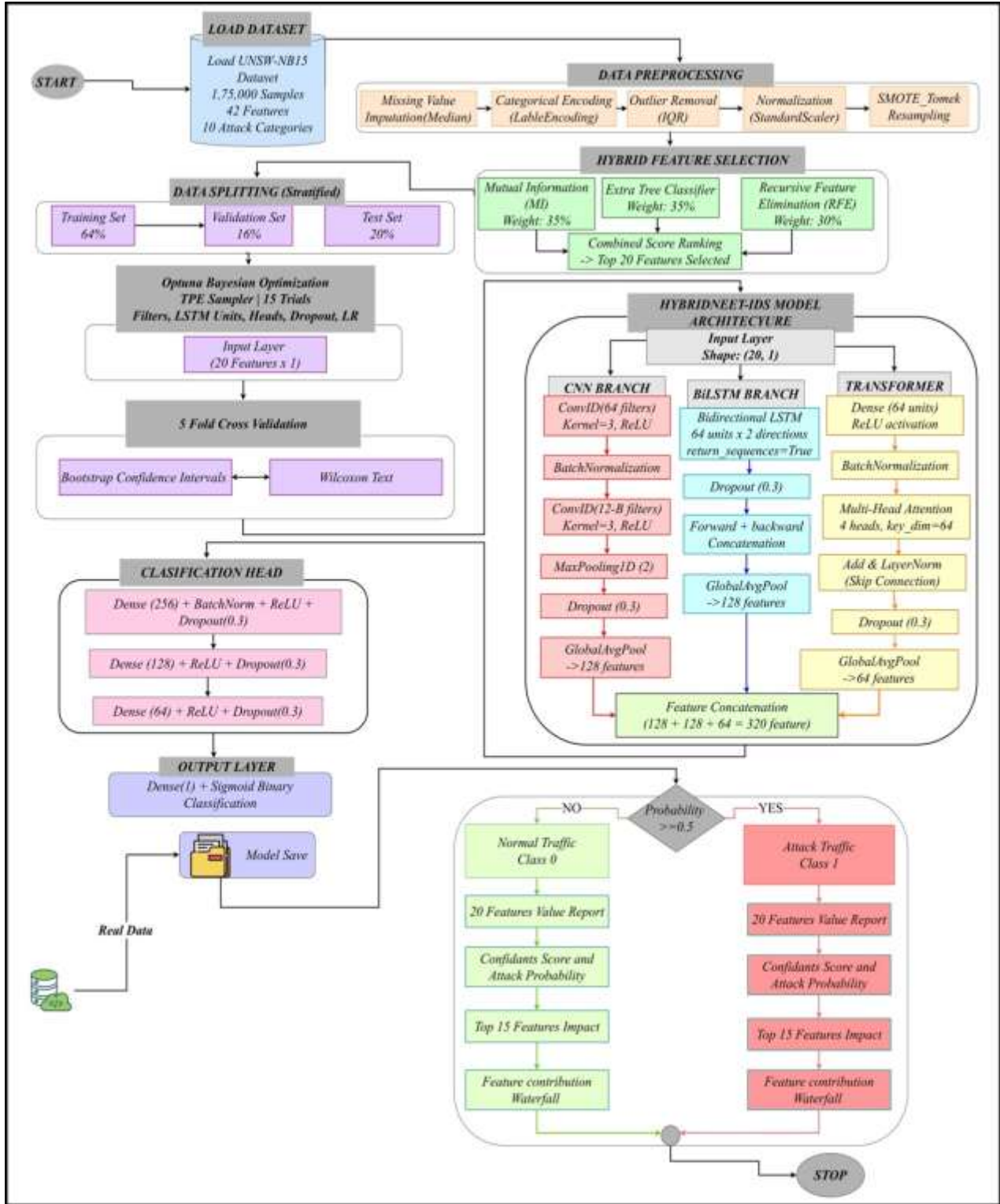


Figure 1: System Architecture.

Source: Authors, (2026).

III.1 CLASS IMBALANCE HANDLING

The pre-processed dataset is not very balanced. There are 85,722 normal samples (52.7%) and 77,023 attack samples (47.3%). SMOTE–Tomek resampling is only done on the training set to fix the bias toward the majority class, as shown by the formula in equation 1. SMOTE creates fake instances of the minority class by using interpolation between neighbours. Tomek linkages, on the other hand, get rid of samples that are noise or on the edge. This combined method makes the class balance better and the decision boundaries clearer. It also cuts the training dataset from 104,156 samples to 95,438 samples. The validation and test datasets stay the same to make sure the test is fair.

$$x_{\text{new}} = x_i + \lambda(x_{\text{nn}} - x_i), \lambda \in (0,1) \quad (1)$$

Where x_i is a minority sample and x_{nn} is one of its k-nearest minority neighbours.

III.2 HYBRID FEATURE SELECTION

The proposed hybrid feature selection strategy uses three different methods that work well together to find the best discriminative features and lower the number of dimensions at the same time. Mutual Information (MI) tells you how statistically dependent each attribute is on the class label. It does this by recording interactions that aren't linear (Eq. 2). The Extra Trees (ET) classifier uses a set of random decision trees to figure out how much impurity goes down on average (Eq. 3). Recursive Feature Elimination (RFE) with Random Forest removes less important features over time by repeatedly testing classification performance on smaller and smaller subsets (Eq. 4). We get the final feature importance score by combining normalized MI, ET significance, and RFE ranking in a weighted way (Eq. 5). This overall score shows which twenty traits are the most important. This reduces dimensionality by over 50% while still allowing for differentiation, as illustrated in Figure 2.

$$I(X, Y) = \sum_{x \in X} \sum_{y \in Y} P(x, y) \log \frac{p(x, y)}{p(x)p(y)} \quad (2)$$

Here, X denotes the random variable representing an input feature and Y represents the class label. The terms $x \in X$ and $y \in Y$ indicate specific realizations of the feature and class variables, respectively. The joint probability $p(x, y)$ measures their co-occurrence, while $p(x)$ and $p(y)$ denote their marginal probabilities.

$$FI_{ET}(f) = \frac{1}{T} \sum_{t=1}^T \Delta I_t(f) \quad (3)$$

In this equation, $\Delta I_t(f)$ represents the decrease in node impurity (e.g., Gini index) contributed by feature f in the t -th decision tree, and T denotes the total number of trees in the ensemble.

$$S_{k+1} = S_k + \arg \min_{f \in S_k} FI(f) \quad (4)$$

Here, S_k denotes the feature subset at iteration k , while S_{k+1} represents the updated subset after removing the least important feature based on importance score $FI(f)$.

$$Score(f) = 0.35MI_f^{norm} + 0.35ET_f^{norm} + 0.30RFE_f \quad (5)$$

In this formulation, MI_f^{norm} , ET_f^{norm} , and RFE_f represent the normalized Mutual Information score, normalized Extra Trees importance, and RFE ranking score of feature f , respectively.

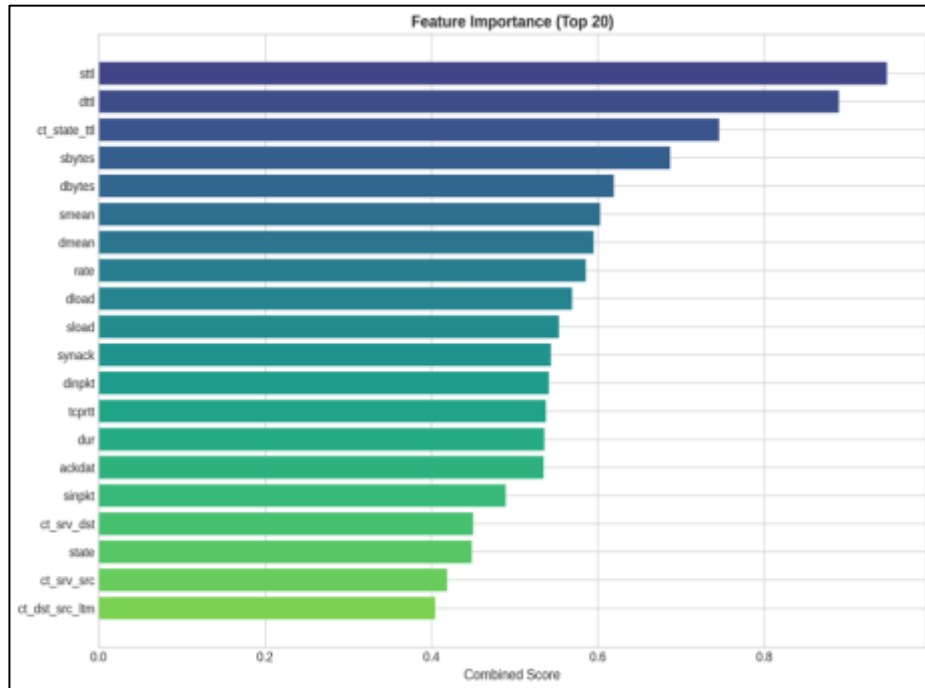


Figure 2: The top 20 properties that the recommended hybrid feature selection technique may use to tell the difference between things. Source: Authors, (2026).

III.3 DATA SPLITTING AND APPLY OPTIMIZATION TECHNIQUE

The pre-processed dataset is divided into training (64%), validation (16%), and testing (20%) sets. The training set is used for model learning, while validation and test sets support hyperparameter tuning and unbiased performance evaluation, respectively. There use the Optuna framework and the Tree-structured Parzen Estimator (TPE) sampler to find the best hyperparameters in 15 tests. The search space includes convolutional filters of sizes {32, 64, 128}, BiLSTM units of sizes {32, 64, 128}, attention heads set to {2, 4, 8}, dropout rates from 0.2 to 0.5, learning rates from 0.00001 to 0.01, and batch sizes of sizes {32, 64, 128}.

We train each setup for 30 epochs, with validation accuracy as the goal function (Eq. 6). The TPE-based Bayesian method uses performance outcomes to figure out the chance of hyperparameters (Eq. 7). This makes it easier to quickly explore new areas. The results show that the learning rate has the biggest effect on performance sensitivity (about 80%). The dropout rate, the number of convolutional filters, attention heads, batch size, and BiLSTM units come next. This guarantees that the HybridNet-IDS framework converges steadily and works better for a wider range of problems. Figure 3 shows what happened when we used Optuna to improve hyperparameters over 15 trials. The optimization history shows that the accuracy of the validation is getting better, going from 84.9% to 90.5%. This means that the search is going well. The analysis of parameter significance indicates that the learning rate is the paramount hyperparameter, constituting approximately 80% of the overall model performance. Next are other architectural and training parameters.

$$\theta^* = \arg \max_{\theta \in \Phi} A_{val}(\theta) \quad (6)$$

Where θ^* denotes a hyperparameter configuration from the search space Φ , and A_{val} represents validation accuracy.

$$p(\theta|y) = \begin{cases} l(\theta), & y < y^* \\ g(\theta), & y \geq y^* \end{cases} \quad (7)$$

Where $l(\theta)$ and $g(\theta)$ are density functions for good and poor observations, respectively, and y^* is a predefined performance threshold.

$$EI(\theta) = \int_{-\infty}^{y^*} (y^* - y) p(y|\theta) dy \quad (8)$$

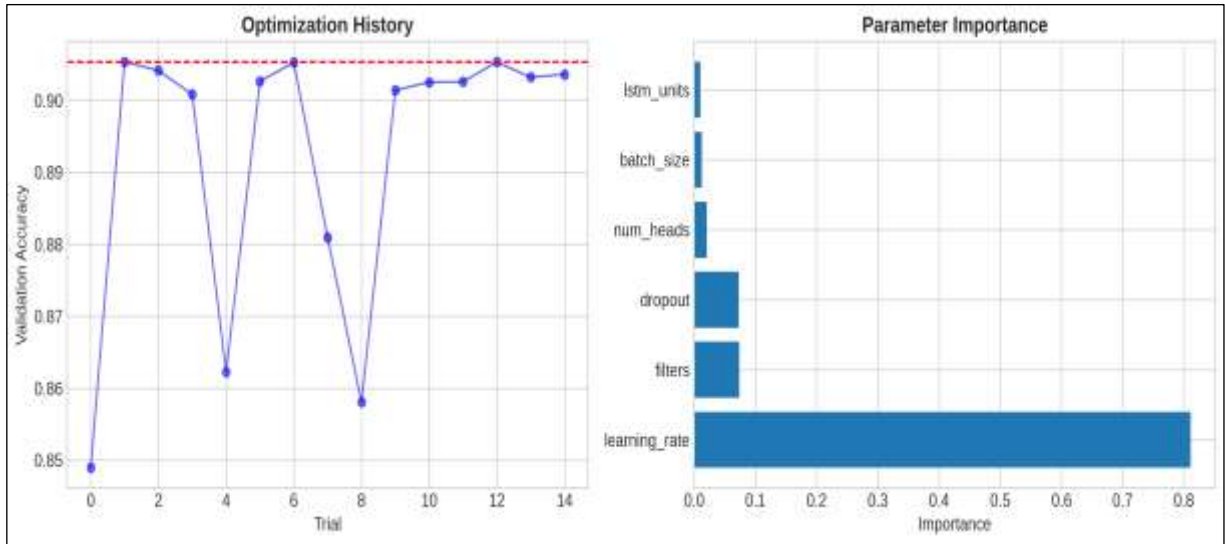


Figure 3: Describes the history of Optuna optimization and points out that learning efficiency is the most important hyperparameter. Source: Authors, (2026).

To make sure the model is strong and the statistics are correct, five-fold stratified cross-validation is used on the training dataset. Here use averaged performance measures across pleats to see if they are stable and general. To check for performance stability and lower bias in assessments, statistical measures like the mean, standard deviation, and confidence intervals are also calculated.

III.4 ARCHITECTURE OF HYBRIDNET-IDS

Three learning branches that work at the same time look at the 20 parts that make up the HybridNet-IDS architecture. These branches should look for patterns in network data that show long-range, spatial, and temporal dependency that work together. Figure 1 shows how the input feature vector is turned into a three-dimensional tensor with the size (20,1)(20,1)(20,1). This makes it easier to use processing that is based on convolution, recurrence, and attention. There are two one-dimensional convolutional layers in the CNN structure. The first layer has 64 filters, and the second layer has 128 filters. The padding is the same, the activation is ReLU, and the kernel size is 3. Next, we use batch normalization, max-pooling, dropout, and global average pooling to make hierarchical spatial representations, which is what Equation (9) shows. The BiLSTM part has 64 units in each direction. This lets it find both forward and backward time relationships in sequences of features. Equation (10) shows that dropout makes things more regular, and global average pooling brings everything together.

The Transformer branch has four heads that do multi-head self-attention, dense feature projection, and layer normalization. As shown in Equation (11), this lets the model change the weight of correlations between features and long-range interactions. The results from the three branches (128, 128, and 64 dimensions) are combined to make a 320-dimensional picture. This information is then processed by a fully connected classification head. It uses sigmoid activation and progressive dimensional reduction to guess how likely an attack is to happen.

$$F_{cnn} = GAP(\sigma(W_c * X + b_c)) \quad (9)$$

Where X is the input feature tensor, $*$ denotes convolution, σ is the ReLU activation, and GAP represents global average pooling.

$$F_{bilstm} = \overrightarrow{LSTM}(X) || \overleftarrow{LSTM}(X) \quad (10)$$

Where $\overrightarrow{LSTM}$ and $\overleftarrow{LSTM}$ denote forward and backward LSTM operations, and $||$ represents concatenation.

$$Attention(Q, K, V) = softmax \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (11)$$

Where Q , K and V are query, key, and value matrices, and d_k is the key dimension.

III.5 CLASSICAL ML ALGORITHMS

It use six traditional machine learning classification methods—Random Forest (RF), Gradient Boosting (GB), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), XGBoost, and LightGBM—to compare intrusion detection after we have optimized the data. Random Forest creates a set of decision trees from bootstrap samples of the training data and uses a majority vote to classify. As shown in Equation (12), this lowers variance and overfitting. Gradient Boosting trains weak learners one at a time by using a loss function that can be differentiated. As shown in Equation (13), each new tree fixes the mistakes made by the models before it, which makes predictions more accurate. The Support Vector Machine uses kernel functions, like the radial basis function (RBF), to make the gap between the attack and normal classes bigger. This helps it find the best separation hyperplane. As shown in Equation (14), this makes it possible to have decision boundaries that aren't straight lines. K-Nearest Neighbors learns from examples by grouping them based on what is the most usual label among the k closest training points, which are found using Euclidean distance, as shown in Equation (15). Equation (16) shows that XGBoost improves regular gradient boosting by adding second-order Taylor expansion, regularization terms, and shrinkage. This makes it faster to converge and better at generalizing. LightGBM enhances gradient boosting through histogram-based feature discretization and leaf-wise tree growth. As shown in Equation (17), this speeds up training and uses less memory on big datasets. Training is done on all models to lower the classification loss on data that has been labeled. Accuracy, precision, recall, F1-score, ROC-AUC, and Matthews Correlation Coefficient are all used in evaluation to make sure it is fair even when there is class imbalance. These better machine learning techniques give good baselines and a complete comparison of results with the offered HybridNet-IDS deep learning architecture.

$$\hat{y} = mode\{h_1(x), h_2(x), h_3(x), \dots, h_T(x)\} \quad (12)$$

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \quad (13)$$

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \quad (14)$$

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2} \quad (15)$$

$$\mathcal{L} = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \quad (16)$$

$$Gain = \frac{(G_L)^2}{H_L + \lambda} + \frac{(G_R)^2}{H_R + \lambda} + \frac{(G)^2}{H + \lambda} \quad (17)$$

IV. EXPERIMENTAL SETUP

The UNSW-NB15 [4] dataset was made by the Australian Centre for Cyber Security. It has real normal and attack traffic that was made with the IXIA Perfect Storm tool. It has looked at 42 properties and nine types of attacks. In binary classification, all attacks are put into one group. Table 2 gives a summary of the dataset's numbers.

Table 2: Dataset Statistics.

Parameter	Value
Original samples	257,673
After preprocessing	162,745
Number of features	42 (reduced to 20)
Normal class (Class 0)	85,722 (52.7%)
Attack class (Class 1)	77,023 (47.3%)
Training set	104,156 (64%)
After SMOTE-Tomek	95,438
Validation set	26,040 (16%)
Test set	32,549 (20%)

Source: Authors, (2026).

TensorFlow 2.x (Keras API) and scikit-learn 1.x are used by the framework. The TPE sampler in Optuna 3.x makes it easier to change Bayesian hyperparameters. Training is done with the Adam optimizer, binary cross-entropy loss, early stopping, adaptive learning rate scheduling, and class weight balancing. The maximum number of epochs for training models is 100, and the batch size is 64. It runs experiments on Google Colab Pro (T4 GPU) and system orchestration and prototyping on a MacBook Air M4.

V. RESULTS

This study's goal is to make an intrusion detection system that is both optimized and strong enough to find cyberattacks on today's networks. Figure 2 shows that the UNSW-NB15 [4] dataset is used to find the top 20 distinguishing features using a mixed feature selection method. Figure 3 shows how optimization has changed over 15 trials, with performance getting better and better. After that, the hyperparameters are changed. The HybridNet-IDS model keeps getting closer together after 100 epochs, as shown in Figure 4. The accuracy of the training is 92.27%, and the accuracy of the validation stays at 90.64%. Loss curves show that regularization works well and that there isn't too much overfitting. On the test set, the model has an MCC of 0.8085, an accuracy of 90.36%, a precision of 90.522%, and a recall of 90.36%. Figure 6 shows that the precision-recall performance is good, with an average accuracy of 0.9738. Figure 5 shows that the discrimination is very good, with a ROC-AUC of 0.9769.

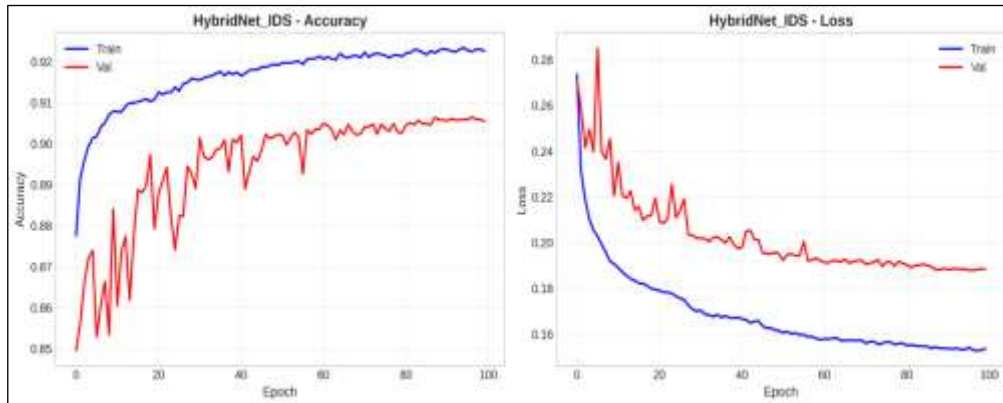


Figure 4: Training and validation accuracy and loss curves illustrating HybridNet-IDS's robust growth and excellent regularization. Source: Authors, (2026).

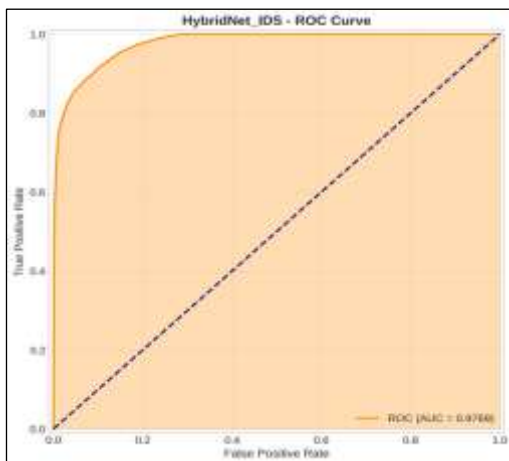


Figure 5: ROC curve showing excellent classification capability along with elevated true positive rates at various thresholds. Source: Authors, (2026).

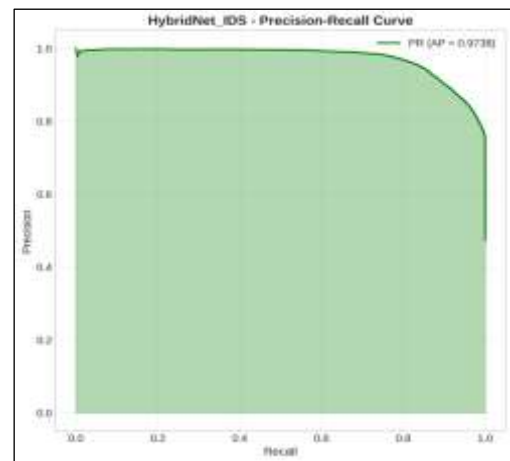


Figure 6: The precision – Recall curve shows that accuracy maintains excellent through an extensive range of recall ranges. Source: Authors, (2026).

Five-fold stratified cross-validation shows that the model is quite stable and can be used in many different situations. The accuracy is always between 91.16% and 91.80%. The average accuracy of 91.53% with minimal variation shows that the model works well across all data partitions. This is shown in Table 3, which also shows the precision, recall, F1-score, and ROC-AUC findings for each fold.

Table 3: Five-Fold Cross-Validation Results.

Fold	Accuracy	Precision	Recall	F1-Score
1	91.80%	91.82%	91.80%	91.80%
2	91.59%	91.68%	91.59%	91.58%
3	91.50%	91.60%	91.50%	91.49%
4	91.59%	91.66%	91.59%	91.59%
5	91.16%	91.54%	91.16%	91.14%
Mean +/- Std	91.53 +/- 0.21	91.66 +/- 0.11	91.53 +/- 0.21	91.52 +/- 0.21

Source: Authors, (2026).

Confusion matrix analysis shows effective intrusion detection, with a true positive rate of 91.25% and specificity of 88.13%, indicating balanced classification performance. The higher sensitivity reflects reduced missed attacks, aligning with security priorities. Figure 7 illustrates absolute and normalized confusion matrix results.

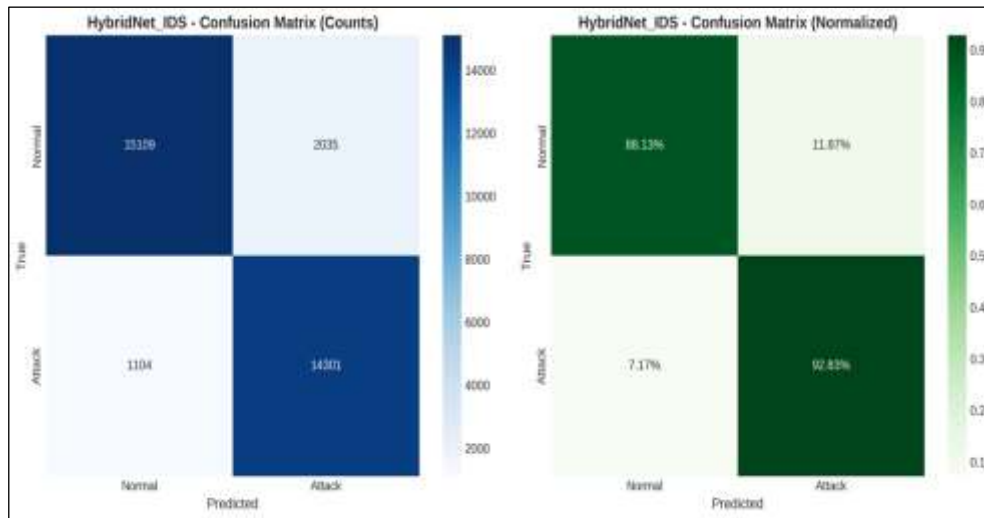


Figure 7: Displays the confusion matrix, which has both absolute counts and normalized percentages of valid and wrong incursion classifications.

Source: Authors, (2026).

Table 4 presents a comparative performance analysis between the proposed HybridNet-IDS and six optimized traditional machine learning models. HybridNet-IDS consistently outperforms all baselines across accuracy, F1-score, ROC-AUC, MCC, and balanced accuracy, demonstrating superior detection capability and robustness. LightGBM and XGBoost follow closely, while SVM shows the lowest overall performance.

Table 4: Comparative Performance Analysis of HybridNet-IDS and Traditional Machine Learning Models.

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	MCC	Specificity	Balanced Acc
HybridNet-IDS	90.36%	90.52%	90.36%	90.36%	97.69%	0.8085	88.13%	90.48%
LightGBM	89.48%	89.65%	89.48%	89.55%	96.72%	0.7908	87.62%	89.35%
XGBoost	89.35%	89.52%	89.35%	89.42%	96.58%	0.7882	87.45%	89.22%
Random Forest	89.12%	89.28%	89.12%	89.12%	96.45%	0.7834	87.25%	89.05%
Gradient Boosting	88.75%	88.92%	88.75%	88.75%	96.12%	0.7762	86.85%	88.62%
KNN	86.25%	86.48%	86.25%	86.32%	94.18%	0.7265	84.35%	86.12%
SVM	84.38%	84.65%	84.38%	84.42%	93.25%	0.6892	82.50%	84.15%

Source: Authors, (2026).

The HybridNet-IDS model is chosen to be put into action, and it is logged as a cyber attack.pkl after looking at a lot of different metrics, like recall, accuracy, precision, ROC-AUC, specificity, balanced accuracy, and the Matthews Correlation Coefficient. After that, XAI algorithms are used to look at and understand five test situations that happen in real time. Figures 8–12 show how important each part is and use LIME-style waterfall plots to show how much each key factor adds to each forecast. The combined dashboard for five real-time test samples in Figure 13 shows a summary of the forecasts. It uses XAI-based visual explanations to show the right and wrong classifications, patterns of feature contribution, confidence ratings, and overall prediction statistics.



Figure 8: Provides a XAI-based analysis for Sample-1, affirming accurate attack detection with complete confidence, while feature importance and LIME-style visualizations identify the principal attributes influencing the final prediction.
Source: Authors, (2026).



Figure 9: Displays XAI-based analysis for Sample-2, demonstrating 100% attack prediction accuracy. LIME-style waterfall graphs highlight key factors impacting the final choice.
Source: Authors, (2026).

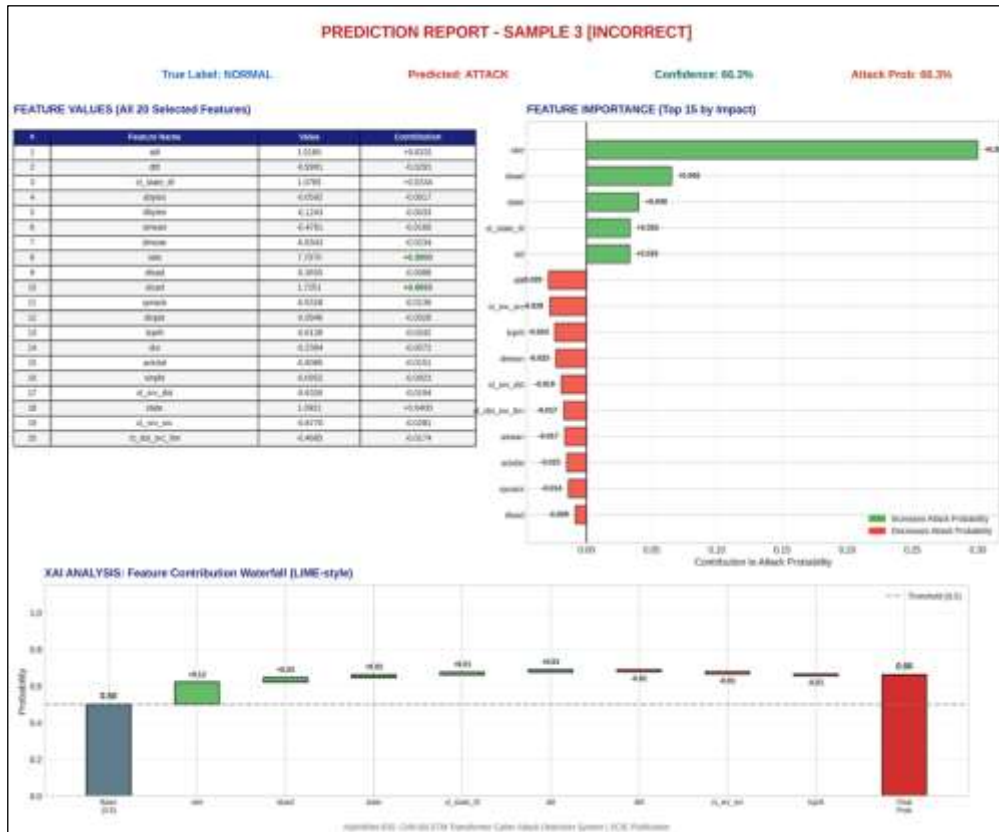


Figure 10: Depicts an incorrect prediction for Sample-3, where normal traffic is misclassified as an attack, with XAI analysis highlighting dominant features contributing to the false alarm decision.

Source: Authors, (2026).



Figure 11: Provides a XAI-based analysis for Sample-4, affirming accurate attack detection with complete confidence and emphasizing the significant feature contributions that affect the final prediction.

Source: Authors, (2026).



Figure 12: Demonstrates the XAI analysis for Sample-5, whereby legitimate traffic is erroneously categorized as an assault, along with feature significance and LIME-style visualizations elucidating the elements leading to the false positive prediction.

Source: Authors, (2026).

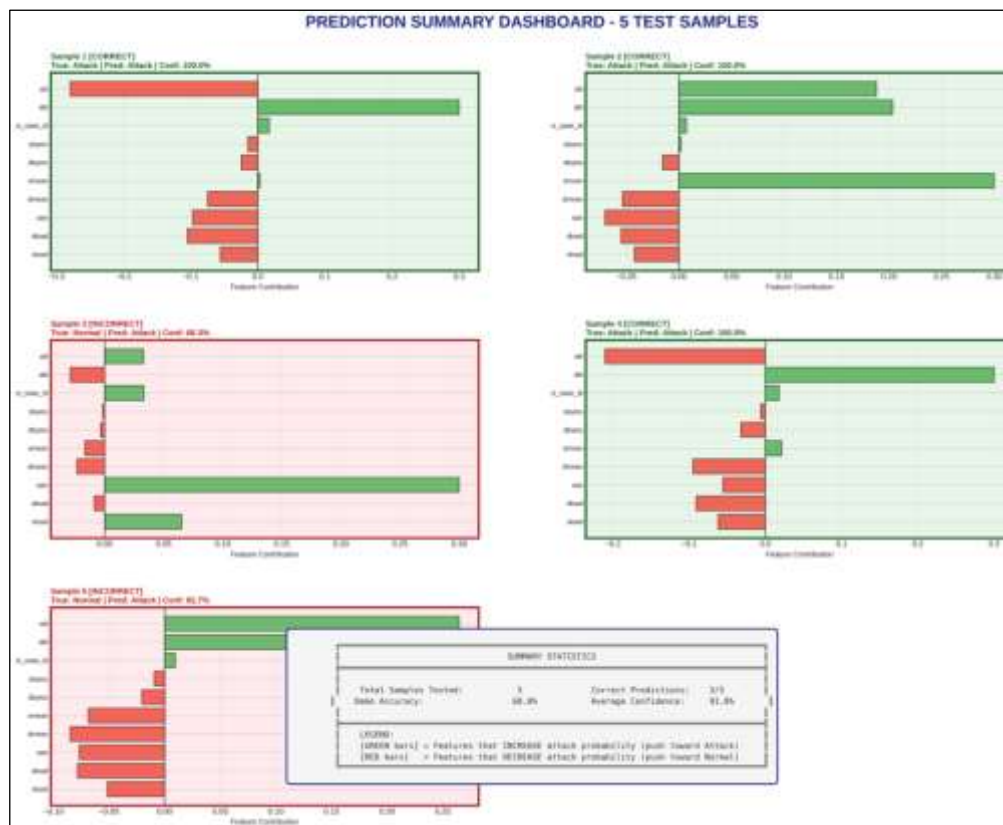


Figure 13: Shows five real-time predictions, including the results of the classification, the contributions of each feature, the confidence levels, and overall performance insights based on XAI.

Source: Authors, (2026).

VI. DISCUSSIONS

The results of the experiment show that using CNN, BiLSTM, and Transformer architectures together to detect network intrusions works well. The model seems to have learned important patterns that can tell the difference between good and bad traffic at many levels of classification. The ROC-AUC score of 97.69% is very close to the theoretical maximum and shows that the model is very good at telling the difference. The Matthews connection coefficient of 0.8085 is a strong number that looks at all four groups in the confusion matrix and doesn't get too excited when there isn't enough of one class. This means that the predictions and ground truth labels in Table 4 are closely related. We used Bootstrap resampling with 1,000 iterations to see how trustworthy the results were. Figure 14 shows that both the accuracy and the F1-score came to the same conclusion: a bootstrap mean of 90.37% and a 95% confidence range of [90.04%, 90.68%]. The fact that HybridNet-IDS has low cross-validation variance and tight confidence limits means that it works well and can be used in a lot of different situations.

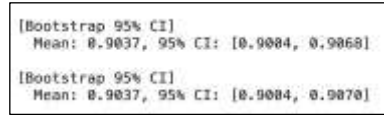


Figure 14: Shows bootstrap-based intervals of trust for accuracy and F1-score, which show that the results are statistically reliable.
Source: Authors, (2026).

The hybrid feature selection method is also good for finding a balance between model complexity and performance. It does this by reducing dimensionality while keeping discriminative information, which lowers the risk of overfitting. Optuna helps hyperparameter optimization by making convergence more stable and getting the most out of the model. When compared to six optimized conventional machine learning baselines, HybridNet-IDS consistently beats them, especially when it comes to class-balanced metrics. This shows that it can be used in real life. Five case studies (Figures 8–12) that show how the model works in real time also show that it works well in practice. XAI-based visual analyses show how each feature affects individual predictions, which helps security analysts understand and trust the decisions made by the model. These results show that the proposed framework meets the high standards for accuracy, reliability, interpretability, and resilience that current intrusion detection systems need. The proposed HybridNet-IDS effectively rectifies significant shortcomings identified in previous intrusion detection research by amalgamating various learning paradigms into a singular optimized framework. The proposed method employs a multi-branch CNN-BiLSTM-Transformer architecture, hybrid feature selection, and Bayesian hyperparameter optimization to simultaneously capture spatial, temporal, and long-range dependencies, in contrast to traditional approaches that rely on hand-crafted features or single-model architectures. Statistical validation through cross-validation and bootstrap analysis rectifies the inadequacy of reliability assessment in previous studies, while the application of the Matthews Correlation Coefficient ensures fair evaluation amidst class imbalance. Also, XAI-based interpretation makes deep models more transparent, which makes them safe to use. The suggested HybridNet-IDS uses a multi-branch deep architecture, customized hyperparameters, and hybrid feature selection to give very accurate detection, great class-balanced performance, and statistical reliability. XAI makes things clearer, which makes it easier to understand and more reliable to make decisions about intrusion detection. This study makes cybersecurity better by making it easier to find dangerous network behavior early and accurately. This protects important digital infrastructure, user data, and online services. Companies, schools, and society as a whole are safer in the digital world when intrusion detection is better. The proposed model requires additional computational resources for training and relies on labeled datasets for supervised learning, despite its superior performance. Also, certain features of a dataset may affect generalization, which shows how important it is to do more cross-dataset validation in future research.

VII. CONCLUSIONS

This study presented HybridNet-IDS, a robust and intelligible intrusion detection system that integrates hybrid feature selection, Bayesian hyperparameter optimization, and a multi-branch deep learning architecture encompassing CNN, BiLSTM, and Transformer models. The proposed method demonstrated excellent discriminative performance on the UNSW-NB15 [4] dataset, evidenced by a high ROC-AUC and Matthews Correlation Coefficient. This means that it can deal with class imbalance well. Cross-validation and bootstrap resampling tests showed that the results were statistically reliable and stable in a lot of different situations. When compared to six optimized standard machine learning baselines, it was even clearer that HybridNet-IDS was better on a number of assessment measures. In addition, real-time case studies that were enhanced by XAI-driven visual explanations showed that the model was useful and easy to understand. The proposed framework addresses significant deficiencies in existing intrusion detection systems by offering a well-proportioned combination of precision, robustness, and transparency. This means that it could be a good choice for today's cybersecurity settings. Future research will enhance the system to incorporate multi-class attack categorization across all UNSW-NB15 [4] categories and will assess cross-dataset generalization utilizing CICIDS2017, NSL-KDD, and CSE-CIC-IDS2018. Limitations encompass dependence on labeled data, increased training complexity, and the potential for performance degradation when confronted with unobserved attack distributions, thereby necessitating the exploration of federated learning, optimized architectures, and the enhancement of adversarial resilience.

VIII. AUTHOR'S CONTRIBUTION

Conceptualization: Subhodip Koley, Sumit Das, Debasish Saha Roy, Jayashree Dhara, Shibani Debnath, Jaya Bhattacharjee, and Rudra Samanta.

Methodology: Subhodip Koley, Sumit Das, Debasish Saha Roy, Jayashree Dhara, Shibani Debnath, Jaya Bhattacharjee, and Rudra Samanta.

Investigation: Subhodip Koley, Sumit Das, Debasish Saha Roy, Jayashree Dhara, Shibani Debnath, Jaya Bhattacharjee, and Rudra Samanta.

Discussion of results: Subhodip Koley, Sumit Das, Debasish Saha Roy, Jayashree Dhara, Shibani Debnath, Jaya Bhattacharjee, and Rudra Samanta.

Writing – Original Draft: Subhodip Koley, Sumit Das, Debasish Saha Roy, Jayashree Dhara, Shibani Debnath, Jaya Bhattacharjee, and Rudra Samanta.

Writing – Review and Editing: Subhodip Koley, Sumit Das, Debasish Saha Roy, Jayashree Dhara, Shibani Debnath, Jaya Bhattacharjee, and Rudra Samanta.

Resources: Subhodip Koley, Sumit Das, Debasish Saha Roy, Jayashree Dhara, Shibani Debnath, Jaya Bhattacharjee, and Rudra Samanta.

Supervision: Subhodip Koley, Sumit Das, Debasish Saha Roy, Jayashree Dhara, Shibani Debnath, Jaya Bhattacharjee, and Rudra Samanta.

Approval of the final text: Subhodip Koley, Sumit Das, Debasish Saha Roy, Jayashree Dhara, Shibani Debnath, Jaya Bhattacharjee, and Rudra Samanta.

IX. ACKNOWLEDGMENTS

The authors thank JIS College of Engineering for providing the assistance and resources that made this study possible.

X. REFERENCES

- [1] "IoT connections worldwide 2034," Statista. Accessed: Dec. 30, 2025. [Online]. Available: <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>
- [2] "Cost of a data breach 2025 | IBM." Accessed: Dec. 30, 2025. [Online]. Available: <https://www.ibm.com/reports/data-breach>
- [3] European Union Agency for Cybersecurity., ENISA threat landscape 2023: July 2022 to June 2023. LU: Publications Office, 2023. Accessed: Dec. 30, 2025. [Online]. Available: <https://data.europa.eu/doi/10.2824/782573>
- [4] "UNSW_NB15." Accessed: Dec. 30, 2025. [Online]. Available: <https://www.kaggle.com/datasets/mrwellsdavid/unswnb15>
- [5] A. Bilen and A. B. Özer, "Cyber-attack method and perpetrator prediction using machine learning algorithms," *PeerJ Comput. Sci.*, vol. 7, p. e475, Apr. 2021, doi: 10.7717/peerj-cs.475.
- [6] Ismail et al., "A Machine Learning-Based Classification and Prediction Technique for DDoS Attacks," *IEEE Access*, vol. 10, pp. 21443–21454, 2022, doi: 10.1109/ACCESS.2022.3152577.
- [7] Y. K. Saheed and M. O. Arowolo, "Efficient Cyber Attack Detection on the Internet of Medical Things-Smart Environment Based on Deep Recurrent Neural Network and Machine Learning Algorithms," *IEEE Access*, vol. 9, pp. 161546–161554, 2021, doi: 10.1109/ACCESS.2021.3128837.
- [8] A. Churcher et al., "An Experimental Analysis of Attack Classification Using Machine Learning in IoT Networks," *Sensors*, vol. 21, no. 2, Jan. 2021, doi: 10.3390/s21020446.
- [9] N. Bhusal, M. Gautam, and M. Benidris, "Detection of Cyber Attacks on Voltage Regulation in Distribution Systems Using Machine Learning," *IEEE Access*, vol. 9, pp. 40402–40416, 2021, doi: 10.1109/ACCESS.2021.3064689.
- [10] R. J. Alzahrani, A. Alzahrani, R. J. Alzahrani, and A. Alzahrani, "Security Analysis of DDoS Attacks Using Machine Learning Algorithms in Networks Traffic," *Electronics*, vol. 10, no. 23, Nov. 2021, doi: 10.3390/electronics10232919.
- [11] K. Kumari and M. Mrunalini, "Detecting Denial of Service attacks using machine learning algorithms," *J. Big Data*, vol. 9, no. 1, p. 56, Apr. 2022, doi: 10.1186/s40537-022-00616-0.
- [12] M. Ahsan et al., "Enhancing Machine Learning Prediction in Cybersecurity Using Dynamic Feature Selector," *J. Cybersecurity Priv.*, vol. 1, no. 1, pp. 199–218, Mar. 2021, doi: 10.3390/jcp1010011.
- [13] I. Avcı and M. Koca, "Cybersecurity Attack Detection Model, Using Machine Learning Techniques," *Acta Polytech. Hung.*, vol. 20, no. 7, pp. 29–44, 2023, doi: 10.12700/APH.20.7.2023.7.2.
- [14] D. Alghazzawi et al., "Efficient Detection of DDoS Attacks Using a Hybrid Deep Learning Model with Improved Feature Selection," *Appl. Sci.*, vol. 11, no. 24, Dec. 2021, doi: 10.3390/app112411634.
- [15] U. Islam et al., "Detection of Distributed Denial of Service (DDoS) Attacks in IOT Based Monitoring System of Banking Sector Using Machine Learning Models," *Sustainability*, vol. 14, no. 14, July 2022, doi: 10.3390/su14148374.
- [16] A. Presekal, A. Ştefanov, V. S. Rajkumar, and P. Palensky, "Attack Graph Model for Cyber-Physical Power Systems Using Hybrid Deep Learning," *IEEE Trans. Smart Grid*, vol. 14, no. 5, pp. 4007–4020, Sept. 2023, doi: 10.1109/TSG.2023.3237011.
- [17] M. Alduailij et al., "Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method," *Symmetry*, vol. 14, no. 6, May 2022, doi: 10.3390/sym14061095.
- [18] M. A. Khalaf and A. Steiti, "Artificial Intelligence Predictions in Cyber Security: Analysis and Early Detection of Cyber Attacks," *Babylon. J. Mach. Learn.*, vol. 2024, pp. 63–68, May 2024, doi: 10.58496/BJML/2024/006.
- [19] A. Alomiri, S. Mishra, and M. AlShehri, "Machine Learning-Based Security Mechanism to Detect and Prevent Cyber-Attack in IoT Networks," *Int. J. Comput. Digit. Syst.*, vol. 15, no. 1, pp. 645–659, Aug. 2024, doi: 10.12785/ijcds/160148.
- [20] F. S. Ferdous, T. Biswas, and A. I. Jony, "Enhancing Cybersecurity: Machine Learning Approaches for Predicting DDoS Attack," *Malays. J. Sci. Adv. Technol.*, pp. 249–255, July 2024, doi: 10.56532/mjsat.v4i3.306.
- [21] A. T. Olutimehin, "The Synergistic Role of Machine Learning, Deep Learning, and Reinforcement Learning in Strengthening Cyber Security Measures for Crypto Currency Platforms," *Asian J. Res. Comput. Sci.*, vol. 18, no. 3, pp. 190–212, Feb. 2025, doi: 10.9734/ajrcos/2025/v18i3586.