ISSN: 2617-5517 (issn.org)
Al-Farabi Journal of Engineering Sciences
https://iasj.rdd.edu.iq/journals/journal/view/97
مجلة الفارابي للعلوم الهندسية تصدرها جامعة الفارابي

Adaptive Uncertainty–Aware CNN–BiLSTM and Neutrosophic Petri Net Framework for Real–Time IDS/IPS

Hind Ali Abdul Hassan[1,*]

[1]Department of Cybersecurity, College of Computer Science and Information Technology, University of Wasit, Iraq.

*Corresponding author at: Hind Ali Abdul Hassan

hind.ali@uowasit.edu.iq

Abstract:

A crucial function of intrusion detection and prevention systems (IDS/IPSs) is to safeguard corporate, cloud, and cyber-physical networks against ever-changing, intricate threats. Class imbalance, high false-positive rates, poorly calibrated confidence estimates, limited interpretability, and strict prevention mechanisms that force uncertain traffic into binary allow-or-block decisions are still issues with many current IDS/IPS solutions, even though deep learning models can successfully capture nonlinear and temporal traffic patterns. An attention mechanism, conformal prediction, an Adaptive Neutrosophic Petri Net, convolutional neural networks, bidirectional long short-term memory networks, and an adaptive uncertainty-aware intrusion detection system/intrusion prevention system are all part of the proposed architecture in this research. In order to detect ambiguous and out-of-distribution traffic, the CNN-BiLSTM-Attention component extracts bidirectional temporal correlations and local feature interactions, while conformal prediction creates calibrated prediction sets. The preventive layer is able to route traffic to allow, block, quarantine, or deep-inspection actions based on the resultant probabilities, conformal ambiguity, entropy, evidence of traffic-drift, and contextual risk, which are mapped into adaptive truth, indeterminacy, and falsity values. The framework was assessed utilizing random, time-based, leakage-free, and cross-dataset protocols on the CSE-CIC-IDS2018 and comparable NetFlow based datasets. In the random-split situation, the suggested model attained an accuracy of 99.84%, a macro-F1 of 99.31%, an MCC of 0.996, a false positive rate of 0.13% and a false negative rate of 0.21%. Under more strict leakage-free assessment, it achieved 99.06% accuracy and 98.14% macro-F1. Under cross-dataset testing, it achieved 96.72% accuracy and 94.06% macro-F1, which shows increased generalization under distribution shift. The full-framework improved the false-positive rate from 0.36% to 0.13% and the false-negative rate from 0.78% to 0.21% compared to the baseline CNN-BiLSTM-Attention. Conformal prediction lowered the estimated calibration error from 0.071 to 0.013, achieved 95.2 % empirical coverage for a nominal coverage level of 95%, and enhanced the quarantine response for unknown or out-of-distribution traffic from 42.0% to 91.8%. The Adaptive Neutrosophic Petri Net further lowered the prohibited benign traffic to 1.7% and permitted the malicious traffic to 0.29% with an average decision delay of 2.03 ms per traffic window. Results show that the suggested framework increases detection performance, calibration, operational safety, uncertainty management, and decision-level explainability, thereby constituting a potential architecture for trustworthy real-time intrusion prevention.

Keywords: Intrusion Detection System; Intrusion Prevention System; CNN-BiLSTM; Neutrosophic Petri Net; Conformal Prediction; Uncertainty Quantification; Explainable Artificial Intelligence; Enterprise Networks.

1. Introduction

Infiltration, web-based assaults, botnet traffic, distributed denial-of-service attacks, brute-force efforts, and zero-day breaches are becoming more commonplace in enterprise networks. Signatures, static thresholds, and binary anomaly judgments are the mainstays of traditional intrusion detection systems. When dealing with encrypted, ambiguous, unbalanced, or maliciously created communications, these strategies may not be as dependable as they are when dealing with recognized threats [1]-[4].

Machine and deep learning improved IDS efficiency by learning distinguishing patterns from traffic logs and flow data directly [1]-[6]. LSTM networks are especially

significant since packet groups and flows in networks tend to have temporal relationships. Recently, a design of LSTM-Neutrosophic Petri - Net (LSTM-NPN) has been applied for sequential classification and three-way preventive choices (accept, block or quarantine) [19], [20]. This approach is useful in that it does not drive uncertain traffic into a strict binary action.

Some restrictions persist, nonetheless, in spite of these improvements. To begin, regional correlations between traffic characteristics may go unnoticed by an LSTM-only classifier prior to temporal modeling. Secondly, in the event of data shift or unknown assaults, probability-to-neutrophic mapping without calibration might lead to overconfident choices. Thirdly, dynamic networks may not be the best fit for set neutrosophic thresholds. To conclude, the fourth issue is that diverse types of attacks need distinct mitigation strategies, and binary IDS/IPS choices fail to achieve this. Lastly, when it comes to operational settings, security managers have a hard time auditing and trusting black-box models.

This research presents a system called Adaptive Uncertainty-Aware CNN-BiLSTM-Neutrosophic Petri Net that aims to overcome these constraints. A number of different components work together in this model: adaptive neutrosophic thresholds, explainable decision tracing, bidirectional temporal modeling based on BiLSTM, attention-based sequence interpretation, uncertainty quantification based on conformal predictions, and CNN-based local representation learning. In addition to improving classification accuracy, we want to lessen the number of risky preventative judgments, bolster quarantine and thorough inspection, and provide security operators with clear proof.

1.1 Contributions of the Study

- A hybrid CNN-BiLSTM-Attention classifier is formulated for spatial-temporal network traffic representation.
- A conformal prediction layer is integrated to produce calibrated prediction sets before IDS/IPS decisions.
- An adaptive neutrosophic mapping is introduced to transform classifier confidence, conformal uncertainty, drift evidence, and risk score into truth, indeterminacy, and falsity values.
- The baseline binary IDS/IPS concept is extended into a two-stage binary and multi-class attack identification framework.
- A multi-level explainability mechanism is designed using SHAP feature attribution, attention weights, and Petri Net transition traces.
- A reproducible experimental protocol is defined, including leakage-free splitting, time-based validation, cross-dataset testing, ablation studies, latency analysis, and operational IDS/IPS metrics.

2. Related Work

2.1 Deep Learning for Intrusion Detection

The capacity of deep learning to learn multi-dimensional, non-linear representations from network data with little reliance on manually-created features has made it a prominent area of study for intrusion detection systems [1, 2]. Deep neural architectures are superior to traditional machine learning methods in two respects: they can simulate the ever-changing patterns linked with emerging cyberattacks and they can uncover previously unseen correlations among traffic parameters.

The ability to capture temporal relationships in packet sequence and flow-based traffic windows has led to the widespread use of recurrent topologies, especially Long Short-Term Memory networks. Similar to how convolutional neural networks were used to derive spatial correlations among neural network features and local feature interactions. When it comes to recognizing complicated and multi-stage assault behaviors, hybrid CNN-LSTM models are the way to go. By integrating local feature extraction and sequential modeling, they significantly boost detection performance [1]-[4].

In recent times, there has been a rise in interest in CNN-BiLSTM-Attention architectures for use in sophisticated intrusion detection systems. In these types of models, convolutional layers pick up local discriminative patterns, reversible long short-term memory (LSTM) layers record both forward and reverse temporal dependencies, and focus mechanisms prioritize the most important time steps or features in a transport window [3, 4]. When looking for coordinated or multi-phase invasions, this layered depiction really shines. On the other hand, dependable operational prevention cannot be guaranteed by high categorization accuracy. In real-world IPS settings, false positives might halt legal services, while dangerous traffic can go unnoticed due to false negatives. Consequently, decision processes that are both explainable and cognizant of ambiguity should supplement categorization models.

2.2 Feature Selection and Explainability

Reducing model generalizability and increasing computational cost, modern IDS datasets often include characteristics that are redundant, irrelevant, noisy, or strongly correlated. For these reasons, feature selection and dimensionality reduction are crucial for maximizing interpretability, decreasing overfitting, and boosting training efficiency [6]-[9]. A popular method for reducing feature dimensionality is Principal Component Analysis (PCA). However, when applied to traffic characteristics, PCA's altered components could mask their original semantic meaning, rendering the method useless for decisions that concern security.

This is why there has been a recent shift toward using interpretable feature selection techniques in cybersecurity applications. These approaches include SHAP-based ranking, tree-based feature significance, recursive feature deletion, and mutual information. These techniques may pinpoint the most important traffic features while maintaining a more transparent relationship between the semantics of the network and the behavior of the models.

Explainable AI is particularly crucial in IDS/IPS systems since security managers need to understand why a specific flow is permitted, banned, quarantined or forwarded for further examination [21]-[23]. SHAP values explain the prediction of the model for each feature,

by quantifying the contributions of each feature to the output of the model. Attention methods provide interpretability at the sequence level, by emphasizing the important steps in time or traffic segments. Further, Petri Net based traces may provide decision level auditability by displaying how categorization output, uncertainty estimates and rule-based reasoning led to the final preventative measures. The combination of these explanation levels may increase confidence, enable forensic investigation and simplify compliance with operating security standards.

2.3 Class Imbalance in IDS Datasets

Class imbalance is perhaps one of most persistent difficulties in intrusion detection. In reality, innocuous samples are generally dominant in the network traffic, whereas important attack classes might be rare. Hence, a model might have excellent overall accuracy and yet be quite bad in recognizing minority attack types. This is particularly troublesome in security applications, since uncommon assaults may be the most harmful.

That is why precision is not enough for IDS evaluations. Accuracy, recall, F1-score, macro-averaged effectiveness, class-wise detecting rate, false-positive rate, false-negative rate, area under the ROC curve, and area under the ROC curve are more illuminating measures [11]-[15]. Class-wise and macro-level indicators show how well the model works with respect to majority and minority groups, which is crucial information to have.

To lessen prejudice against the majority, many approaches have been put up. While focused loss directs training towards challenging and misclassified data, class-weighted learning raises the penalty for mistakes in minority classes [11], [15]. To rebalance datasets, data-level methods like the SMOTEENN method and CTGAN may be used to create synthetic or clean samples [12]-[14]. It is important to exercise caution while using synthetic traffic generation, however, since using unrealistic synthetic patterns might cause overfitting and provide unrealistically positive assessment findings. It is crucial to thoroughly verify the produced samples and assess their performance using evaluation techniques that account for both leakage and time.

2.4 Uncertainty-Aware Intrusion Prevention

Direct binary or multi-level judgments are the intended output of the majority of IDS and IPS models. In operational settings, where classifier confidence could be miscalibrated or when traffic behavior is unclear, this straightforward method might be dangerous. While blocking potentially harmless but unpredictable traffic might affect legitimate services, enabling potentially harmful but unpredictable traffic could leave the network vulnerable. Therefore, real intrusion prevention requires decision making that is uncertainty aware.

During decision-making processes, truth, falsehood, and indeterminacy may be represented mathematically using Neutrosophic Petri Nets [19], [20]. They are therefore well-suited for use in three-way security choices, including quarantine, block, and allow. Instead of making a risky judgment on the fly, a neutrosophic decision level may expressly represent ambiguous instances and send cases to quarantine for deep inspection, unlike traditional binary prevention logic.

An auxiliary approach for uncertainty quantification is given by conformal prediction, which builds prediction sets in finite-sample coverage guarantees an under standard exchangeability assumptions [16]-[18]. In the functioning of IDS/IPS, a single-label conformal predictions set may mean a confident classification while multiple-label or empty predictions sets can indicate an uncertain classification. Such doubtful instances may subsequently be addressed by the quarantine, analyst review or detailed inspection. This technique gives a more rational way to get uncertainty than simply using raw Softmax or the sigmoid probabilities that are typically poorly tuned for deep neural networks.

In an effort to situate the proposed framework in the current literature, Table 1 contrasts the key research directions for deep learning-based IDS, comprehensible feature selection and Petri Net-based preventative models. The comparison indicates that the prior research has major contributions in terms of classification accuracy, interpretability and decision modeling. However, earlier studies frequently lack a cohesive mechanism that integrates calibrated uncertainty estimates with adaptive real-time preventive logic.

Table 1. Comparison of related research directions

Study Direction	Main Focus	Limitation Addressed by the Proposed Research
CNN/LSTM-based IDS models	Traffic classification and temporal pattern learning	Limited integration of calibrated uncertainty and operational prevention logic
Feature selection and XAI studies	Dimensionality reduction and decision explanation	Limited connection with real-time IPS routing and action-level auditability
Petri Net and neutrosophic prevention models	Rule-based allow, block, and quarantine decisions	Need for adaptive thresholds, conformal uncertainty, and multi-class attack handling

Current intrusion detection system (IDS) research is often focused on improving detection performance, as shown in Table 1. However, in order to implement an IPS in a realistic setting, further layers are needed to address uncertainties calibration, decisions routing, and auditability, respectively. The suggested system builds upon previous efforts by integrating the results of deep classification with conformal predictions and a decision layer based on an Adapted Neutrosophic Petri Net.

2.5 Research Gap

According to the literature, there has been a lot of development in the areas of explainable cybersecurity models, feature selection, imbalance management, and intrusion detection using deep learning. Despite this, these aspects are often treated independently in the literature. Classification accuracy is often the primary focus of deep IDS models, but calibrated uncertainty estimate is often lacking. While some research has integrated feature selection and explainability with real-time IPS action routing, other studies have not. Similarly, neutrosophic and Petri Net models provide potential decision-making frameworks, but they often do not have adaptive uncertainty thresholds, enable conformal predictions, or handle multi-class attacks robustly.Table 2 summarizes the main gaps found in the papers that were examined. Due to these deficiencies, an intrusion detection system (IDS) and intrusion prevention system (IPS) that is precise, uncertainty-aware, interpretable, and well-suited to secure operational prevention is required.

Table 2. Summary of research gaps and proposed responses

Research Gap	Response in the Proposed Framework
Uncalibrated classifier confidence	Conformal prediction generates calibrated prediction sets before prevention decisions.
Unsafe binary IDS/IPS actions	Adaptive Neutrosophic Petri Nets route ambiguous traffic to quarantine or deep inspection.
Limited interpretability	SHAP, attention weights, and Petri Net traces provide feature-level, sequence-level, and decision-level explanations.
Risk of optimistic validation	Time-based, leakage-free, and cross-dataset validation protocols are specified.
Weak handling of minority attacks	Class weighting, focal loss, and imbalance-aware evaluation are incorporated.

As demonstrated in Table 2, the fundamental restriction in the existing research is not the lack of good classifiers but the lack of an integrated, prevention-oriented framework that integrates classification, calibrated uncertainty, explanation, and safe decision routing. Therefore, the proposed system fills this gap by combining CNN-BiLSTM-Attention classification methods with conformal prediction methods and an adaptive -Neutrosophic Petri Net decision-making layer. This combination intends to increase the accuracy of detection and minimize dangerous binary judgments, while also increasing interpretability and supporting effective real-time intrusion prevention.

3. Proposed Methodology

3.1 Architecture of the Proposed Framework

The suggested architecture has six operational levels, i.e., preprocessing of data, feature selection and class balance, CNN-BiLSTM-Attention classification, a conformal uncertain quantification, Adaptive and Neutrosophic Petri - Net decision making, and explainability/audit. The rewritten manuscript uses a revamped methodology as shown in Figure 1.

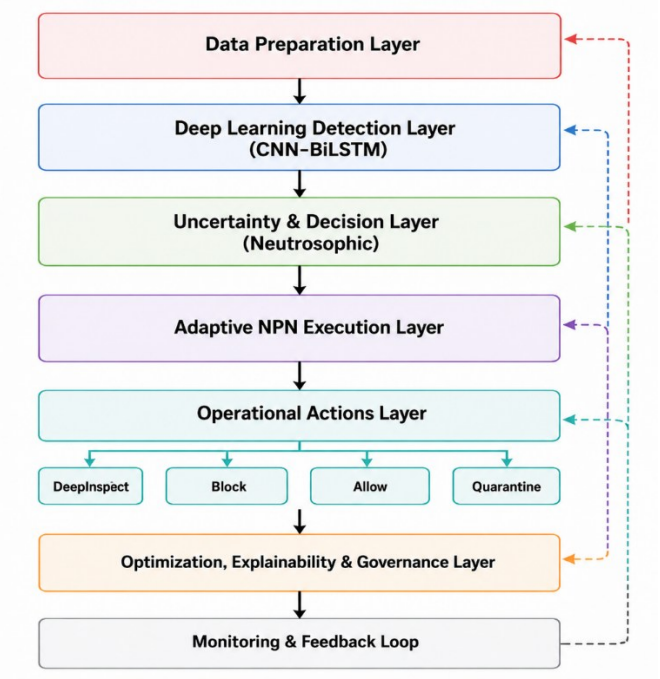


Figure 1. Proposed adaptive uncertainty-aware CNN-BiLSTM-NPN IDS/IPS architecture.

The essential components of a proposed architecture are listed in Table 3, along with the function of every layer in the general detection process. It also illustrates how these components contribute to the enhancement of resilience, interpretability, uncertainty awareness and operational decision making.

Table 3. Main components of the proposed framework.

Layer	Main function	Expected contribution
Preprocessing	Clean, encode, scale, remove leakage, and construct traffic windows	Lower noise and improve validation reliability
Feature selection and balancing	PCA, MI, RFE, SHAP ranking, class weights, focal loss, optional CTGAN/SMOTEENN	Compact, interpretable, and balanced representation
CNN-BiLSTM-Attention	Spatial-temporal classification with attention-based sequence weighting	Robust attack detection and sequence-level interpretability
Conformal prediction	Prediction sets, nonconformity scores, and calibrated uncertainty	Safer routing of ambiguous traffic
Adaptive NPN	Allow, block, quarantine, or deep-inspect through adaptive neutrosophic thresholds	Lower false-positive blocking and operational control
Explainability and audit	SHAP, attention heatmaps, and Petri Net decision traces	Administrator trust and forensic accountability

3.1.1 Algorithm of the Proposed Framework

Algorithm summarizes the proposed workflow from data preparation to operational IDS/IPS action routing.

Algorithm 1. Adaptive uncertainty-aware CNN-BiLSTM-NPN IDS/IPS workflow

Input: Network traffic dataset (D), optional real-time traffic stream (S), window length (w)
Output: Final traffic action: allow, block, quarantine, or deep inspection

1. Load the network traffic dataset (D) and, if available, the real-time stream (S).
2. Remove missing, infinite, duplicate, and inconsistent records.
3. Encode binary and multi-class labels.
4. Apply Z-score normalization to all numerical features.
5. Remove highly correlated, redundant, and irrelevant features.
6. Evaluate feature selection methods, including PCA, mutual information, RFE, tree-based importance, and SHAP ranking.
7. Handle class imbalance using class weighting and focal loss; optionally evaluate CTGAN or SMOTEENN after validation.
8. Convert traffic records into sequential windows of length (w).
9. Split the data using random, time-based, leakage-free, and cross-dataset validation protocols.
10. Train the CNN-BiLSTM-Attention classifier on the prepared traffic windows.
11. Calibrate the trained classifier using a separate conformal calibration set.
12. For each incoming traffic window, generate class probabilities and conformal prediction sets.
13. Compute the neutrosophic values (T), (I), and (F) using class probability, conformal uncertainty, drift evidence, and risk score.
14. Update the adaptive thresholds $(\backslash theta\_T), (\backslash theta\_I), (\backslash theta\_F), and\ (\backslash theta\_R)$ when performance drift is detected.
15. Fire the corresponding Neutrosophic Petri Net transition.
16. Route the traffic to the appropriate action: allow, block, quarantine, or deep inspection.
17. Log SHAP explanations, attention scores, NPN traces, latency, throughput, and the final security action.
18. End.

3.2 Detailed Mathematical Formulation of the Proposed Framework

3.2.1 Overall Processing Pipeline

The proposed framework is designed as an adaptive uncertainty-aware intrusion detection and prevention system. It receives raw network traffic records, cleans and normalizes them, selects the most informative features, handles class imbalance, converts traffic into sequential windows, classifies the traffic using a CNN-BiLSTM-Attention model, estimates prediction uncertainty using conformal prediction, and finally routes the traffic through an Adaptive Neutrosophic Petri Net decision layer. The final output is one of four operational actions: allow, block, quarantine, or deep inspection.

3.2.2 Input Network Traffic Dataset

The system starts from a labeled network traffic dataset. Each sample contains traffic features, a class label, and optionally a timestamp. This representation allows the framework to support both offline training and real-time traffic analysis.

$$D = \{(x_i, y_i, t_i)\}_{i=1}^{N} \quad (1)$$

Where: D is the complete dataset, x_i is the feature vector of the i[th] traffic record, y_i is its label, t_i is the timestamp, and N is the total number of samples.

3.2.3 Leakage-Free Data Partitioning

Before normalization, feature selection, balancing, or model calibration, the dataset must be divided into training, calibration, and testing subsets. This prevents information leakage from the test set into the training process and produces a more realistic performance estimate.

$$D = D_{\text{train}} \cup D_{\text{cal}} \cup D_{\text{test}}, \quad D_{\text{train}} \cap D_{\text{cal}} \cap D_{\text{test}} = \varnothing \quad (2)$$

Where: D_{train} is used for model training, D_{cal} is used for conformal calibration, D_{test} is used for final evaluation, and $\emptyset$ means that the subsets do not overlap.

3.2.4 Z-Score Normalization

After splitting the dataset, numerical features are standardized using only the mean and standard deviation calculated from the training set. This ensures that calibration and testing data remain unseen during preprocessing.

$$x'_{i,j} = \frac{x_{i,j} - \mu_j^{\text{train}}}{\sigma_j^{\text{train}} + \varepsilon} \quad (3)$$

Where: $x'_{i,j}$ is the normalized value of feature j in sample i, $x_{i,j}$ is the original value, μ_j^{train} is the training-set mean of feature j, $\sigma\sigma_j^{\text{train}}$ is the training-set standard deviation, and ε is a small value added to avoid division by $zero$.

3.2.5 Feature Correlation Filtering

Highly correlated features may increase redundancy and computational cost. Therefore, pairs of features with strong correlation are detected, and one feature from each redundant pair can be removed.

$$\rho_{j,k} = \frac{\text{cov}(x_j, x_k)}{\sigma_j \sigma_k} \quad (4)$$

Where: $\rho_{j,k}$ is the correlation coefficient between features j and k, $\text{cov}(x_j, x_k)$ is their covariance, and σ_j, σ_k are their standard deviations.

3.2.6 Interpretable Feature Ranking

The proposed framework compares multiple feature-selection methods, including mutual information, recursive feature elimination, tree-based importance, and SHAP ranking. A combined ranking score can be used to select the most informative and interpretable feature subset.

$$\text{Score}(f_j) = \lambda_1 \text{MI}(f_j, y) + \lambda_2 \text{RFE}(f_j) + \lambda_3 \text{TreeImp}(f_j) + \lambda_4 \text{SHAP}(f_j) \quad (5)$$

Where: $\text{Score}(f_j)$ is the final importance score of features f_j, MI is mutual information with the target label, RFE is recursive feature elimination ranking, TreeImp is tree-based importance, SHAP is SHAP-based attribution, and λ_1 to λ_4 are weighting coefficients.

3.2.7 Selection of the Final Feature Subset

The selected feature subset should maximize detection performance while reducing dimensionality and preserving interpretability. Therefore, the best subset is chosen by balancing macro-F1 performance, feature size, and computational cost.

$$S^* = \underset{S}{\text{argmax}} \left[\text{MacroF1}(S) - \lambda \frac{|S|}{d} - \eta C(S) \right] \quad (6)$$

Where: S^* is the optimal selected feature subset, S is a candidate feature subset, $MacroF1(S)$ is the macro-F1 score obtained using subset S, $|S|$ is the number of selected features, d is the original number of features, $C(S)$ is computational cost, and λ, η control the penalties.

3.2.8 Class Weighting for Imbalanced IDS Data

IDS datasets are usually imbalanced because benign traffic dominates while some attack types are rare. To reduce majority-class bias, class weights are assigned inversely proportional to class frequency.

$$\alpha_c = \frac{N}{CN_c} \quad (7)$$

Where: α_c is the weight of class c, N is the total number of samples, C is the number of classes, and N_c is the number of samples belonging to class c.

3.2.9 Focal Loss for Minority Attack Detection

Focal loss is used to focus training on difficult and minority-class samples. It reduces the influence of easy samples and increases the penalty for misclassified attack records.

$$L_{\text{FL}} = -\frac{1}{N}\sum_{i=1}^{N} \alpha_{y_i}\left(1 - p_{i,y_i}\right)^{\gamma} \log\left(p_{i,y_i}\right) \quad (9)$$

Where:
L_{FL} is the focal loss, α_{y_i} is the class weight of the true class, p_{i,y_i} is the predicted probability of the true label, γ is the focusing parameter, and N is the number of training samples.

3.2.10 Sequential Traffic Window Construction

After pre-processing and feature selection, the traffic data are transformed into sorted windows. Each window is made up of a series of w consecutive recordings, so the model may learn the temporal attack behavior instead of addressing each record singly.

$$X_t = [x'_{t-w+1}, x'_{t-w+2}, \dots, x'_t] \in \mathbb{R}^{w \times m} \quad (10)$$

Where: X_t is the traffic window ending at time t, w is the window length, m is the number of selected features, and $\mathbb{R}^{w \times m}$ means that each window is represented as a matrix with w time steps and m features.

3.2.11 CNN-Based Local Feature Extraction

The CNN layer extracts local spatial relationships among traffic features inside each window. This helps identify short-range patterns that may indicate abnormal behavior.

$$H_{\text{CNN}} = \phi(\text{Conv1D}(X_t, W_C) + b_C) \quad (11)$$

Where: H_{CNN} is the extracted convolutional feature map, $Conv1D$ is the one-dimensional convolution operation, X_t is the input traffic window, W_C is the convolution kernel, b_C is the bias, and ϕ is the activation function.

3.2.12 BiLSTM Temporal Dependency Learning

The BiLSTM layer models the traffic sequence in both forward and backward directions. This enables the framework to capture temporal dependencies before and after each time step in the traffic window.

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \quad (12)$$

Where: h_t is the final bidirectional hidden representation at time step t, $\overrightarrow{h_t}$ is the forward LSTM hidden state, $h\_\overleftarrow{h_t}$ is the backward LSTM hidden state.

3.2.13 Attention-Based Sequence Weighting

Not all-time steps in a traffic window contribute equally to intrusion detection. The attention layer assigns higher weights to the most informative time steps and produces a context vector for classification.

$$e_t = v_a^T \tanh(W_a h_t + b_a) \quad (13)$$

Where: e_t is the attention score of time step v_a^T and W_a are trainable attention parameters, h_t is the BiLSTM hidden state, and b_a is the attention bias.

$$a_t = \frac{\exp(e_t)}{\sum_{k=1}^{w} \exp(e_k)} \quad (14)$$

Where: a_t is the normalized attention weight of time step t, a_t is its attention score, and w is the total number of time steps in the window.

$$s = \sum_{t=1}^{w} a_t \, h_t \quad (15)$$

Where: s is the final attention-based context vector, a_t is the attention weight, and h_t is the hidden representation of time step t.

3.2.14 Final Class Probability Prediction

The context vector is passed to a Softmax output layer to produce class probabilities. In binary detection, the output includes benign and attack probabilities. In multi-class detection, the output includes benign traffic and multiple attack categories.

$$p(y = c \mid X_t) = \frac{\exp(z_c)}{\sum_{k=1}^{C} \exp(z_k)} \quad (16)$$

Where: $p(y = c \mid X_t)$ is the predicted probability that window X_t belongs to class c, z_c is the logit of class c, and C is the number of classes.

3.3.15 Conformal Nonconformity Score

After training, a separate calibration set is used to estimate prediction uncertainty. The nonconformity score measures how unsuitable the true class appears according to the classifier probability.

$$A_i = 1 - p(y_i \mid X_i) \quad (17)$$

Where: A_i is the nonconformity score of calibration sample i, $p(y_i \mid X_i)$ is the predicted probability assigned to the true label y_i, and X_i is the calibration window.

3.2.16 Calibrated Conformal Quantile

The calibration scores are sorted to compute a threshold quantile. This threshold determines which labels should be included in the conformal prediction set for a new sample.

$$q_\alpha = \text{Quantile}_{\lceil (n_{\text{cal}}+1)(1-\alpha) \rceil / n_{\text{cal}}} \left(\{A_i\}_{i=1}^{n_{\text{cal}}} \right) \quad (18)$$

Where: q_α is the calibrated conformal threshold, α is the allowed error level, n_{cal} is the number of calibration samples, and A_i is the set of calibration nonconformity scores.

3.2.17 Conformal Prediction Set

For a new traffic window, the conformal layer returns a prediction set rather than a single class only. A singleton set indicates high confidence, while multi-label or empty sets indicate uncertainty.

$$C_\alpha(X_t) = \{c : 1 - p(c \mid X_t) \leq q_\alpha\} \quad (19)$$

Where: $C_\alpha(X_t)$ is the conformal prediction set for traffic window X_t, c is a candidate class, $p(c \mid X_t)$ is the predicted probability of class c, and q_α is the calibrated threshold.

3.2.18 Conformal Ambiguity Measure

The size of the conformal prediction set is used to estimate ambiguity. A singleton set has low uncertainty, a multi-label set has higher uncertainty, and an empty set indicates severe uncertainty.

$$I_{\text{CP}}(X_t) = \begin{cases} 1, & |C_\alpha(X_t)| = 0 \\ \frac{|C_\alpha(X_t)| - 1}{C - 1}, & \text{otherwise} \end{cases} \quad (20)$$

Where: $I_{\mathrm{CP}}(X_t)$ is conformal ambiguity, $|C_\alpha(X_t)|$ is the number of labels inside the prediction set, and C is the total number of classes.

3.2.19 Entropy-Based Probability Uncertainty

In addition to conformal ambiguity, entropy is used to measure how dispersed the Softmax probabilities are. High entropy means that the classifier is uncertain among several classes.

$$H(X_t) = -\frac{1}{\log C}\sum_{c=1}^{C} p\,(c \mid X_t)\log\big(p(c \mid X_t)\big) \quad (21)$$

Where: $H(X_t)$ is the normalized entropy uncertainty, C is the number of classes, and $\sum_{c=1}^{C} p\,(c \mid X_t)$is the predicted probability of class c.

3.2.20 Drift Evidence Estimation

Network traffic may change over time. Therefore, drift evidence is estimated by comparing the recent traffic distribution with the training distribution. Higher drift indicates that the model may be operating under unseen conditions.

$$D_{\mathrm{rift}}(X_t) = \min\left(1, \frac{\mathrm{JS}(P_{\mathrm{recent}} \parallel P_{\mathrm{train}})}{\tau_D}\right) \quad (22)$$

Where: $D_{\mathrm{rift}}(X_t)$ is the drift score, JS is the Jensen-Shannon divergence, P_{recent} is the recent traffic distribution, P_{train} is the training distribution, and τ_D is the drift normalization threshold.

3.2.21 Operational Risk Score

Some traffic may be more dangerous because of its source, destination, protocol, service, or attack likelihood. The operational risk score combines such contextual indicators into one value.

$$R(X_t) = \sigma\left(\beta_0 + \sum_{k=1}^{K} \beta_k\, r_k(X_t)\right) \quad (23)$$

Where: $R(X_t)$ is the operational risk score, σ is the sigmoid function, β_0 is the bias, β_k is the weight of risk factor k, and $r_k(X_t)$ is a contextual risk indicator.

3.2.22 Neutrosophic Truth Mapping

The neutrosophic layer converts classifier output into three values: truth, falsity, and indeterminacy. Truth represents the degree of maliciousness. In multi-class IDS, it is calculated as the highest probability among attack classes.

$$T(X_t) = \max_{c \in A} p(c \mid X_t) \quad (24)$$

Where: $T(X_t)$ is the truth value, A is the set of attack classes, and $p(c \mid X_t)$ is the predicted probability of attack class c.

3.2.23 Neutrosophic Falsity Mapping

Falsity represents the degree to which traffic is benign. It is calculated from the classifier probability assigned to the benign class.

$$F(X_t) = p(\mathrm{benign} \mid X_t) \quad (25)$$

Where: $F(X_t)$ is the falsity value and $p(\mathrm{benign} \mid X_t)$ is the predicted probability that the traffic window is benign.

3.2.24 Neutrosophic Indeterminacy Mapping

Indeterminacy represents uncertainty. It combines conformal ambiguity, entropy uncertainty, drift evidence, and operational risk into one adaptive uncertainty value.

$$I(X_t) = \mathrm{clip}(\omega_1 I_{\mathrm{CP}}(X_t) + \omega_2 H(X_t) + \omega_3 D_{\mathrm{rift}}(X_t) + \omega_4 R(X_t), 0,1) \quad (26)$$

Where: $I(X_t)$ is the indeterminacy value, $I_{CP}(X_t)$ is conformal ambiguity, $H(X_t)$ is entropy uncertainty, $D_{rift}(X_t)$ is drift evidence, $R(X_t)$ is operational risk, ω_1 to ω_4 are weighting coefficients, and $clip(.,0,1)$ limits the value between 0 and 1.

3.2.25 Adaptive Neutrosophic Petri Net Definition

The decision layer is formalized as an Adaptive Neutrosophic Petri Net. It receives classifier probabilities, conformal sets, neutrosophic values, and risk scores, then fires the appropriate transition to produce the final prevention action.

$$\text{NPN} = (P.\text{Tr}.E.M_0.W.\Theta) \tag{27}$$

Where: P is the set of places, Tr is the set of transitions, E is the set of directed arcs, M_0 is the initial marking, W is the transition weight function, and $\Theta = \{\theta\_T, \theta\_I, \theta\_F, \theta\_R\}$ is the adaptive threshold vector.

3.2.26 NPN Transition Conditions

The NPN decision logic uses thresholds to determine whether traffic should be allowed, blocked, quarantined, or sent to deep inspection. Deep inspection is prioritized when traffic is both uncertain and risky.

$$\begin{aligned} g_{\text{deep}} &= 1 \text{ if } I(X_t) \geq \theta_I \text{ and } R(X_t) \geq \theta_R \\ g_{\text{block}} &= 1 \text{ if } T(X_t) \geq \theta_T \text{ and } I(X_t) < \theta_I \\ g_{\text{allow}} &= 1 \text{ if } F(X_t) \geq \theta_F \text{ and } I(X_t) < \theta_I \\ g_{\text{quarantine}} &= 1 \text{ otherwise} \end{aligned} \tag{28}$$

Where: g_{deep}, g_{block}, g_{allow}, and $g_{\text{quarantine}}$ are transition indicators, θ_T is the truth threshold, θ_I is the indeterminacy threshold, θ_F is the falsity threshold, and θ_R is the risk threshold.

3.2.27 Final Prevention Action (Priority-based action rule)

The final action is selected according to a priority rule. High-risk uncertain traffic is sent to deep inspection first. Confident malicious traffic is blocked, confident benign traffic is allowed, and all remaining ambiguous traffic is quarantined.

$$\text{Action}(X_t) = \begin{cases} \text{DeepInspect}, & I(X_t) \geq \theta_I \text{ and } R(X_t) \geq \theta_R \\ \text{Block}, & T(X_t) \geq \theta_T \text{ and } I(X_t) < \theta_I \\ \text{Allow}, & F(X_t) \geq \theta_F \text{ and } I(X_t) < \theta_I \\ \text{Quarantine}, & \text{otherwise} \end{cases} \tag{29}$$

Where: $\text{Action}(X_t)$ is the final IDS/IPS action for traffic window (X_t), ***DeepInspect*** means sending traffic to detailed analysis, ***Block*** means preventing the flow, ***Allow*** means passing the flow, and ***Quarantine*** means isolating ambiguous traffic.

3.2.28 Dynamic Threshold Optimization

The thresholds are not fixed permanently. They are optimized to reduce false alarms, prevent malicious traffic from being allowed, control quarantine load, reduce latency, and improve macro-level detection performance.

$$\Theta^* = \underset{\Theta}{\text{argmin}}[\lambda_1 \text{FPR}(\Theta) + \lambda_2 \text{FNR}(\Theta) + \lambda_3 \text{QR}(\Theta) + \lambda_4 \text{Latency}(\Theta) - \lambda_5 \text{MacroF1}(\Theta)] \tag{30}$$

Where: Θ^* is the optimized threshold vector, FPR is the false-positive rate, FNR is the false-negative rate, QR is the quarantine rate, Latency is decision delay, $MacroF1$ is $macro - average\ F1 - score$, and λ_1 to λ_5 control the importance of each term.

3.2.29 Threshold Adaptation Under Drift

When performance degradation or traffic drift is detected, the threshold vector is updated using recent verified samples. This allows the IDS/IPS to remain adaptive under changing network conditions.

$$\Theta_{t+1} = \Theta_t + \eta_\theta \Delta\Theta_t \tag{31}$$

Where: Θ_{t+1} is the updated threshold vector, Θ_t is the current threshold vector, η_θ is the adaptation learning rate, and $\Delta\Theta_t$ is the threshold correction estimated from recent validation, drift, or analyst feedback.

3.2.30 Explainability and Audit Logging (Audit Record)

The final layer stores feature-level explanations, attention weights, NPN traces, and operational decisions. This allows administrators to understand why a flow was allowed, blocked, quarantined, or inspected.

$$\text{Log}_t = \begin{Bmatrix} \text{ID}_t, \hat{y}_t, C_\alpha(X_t), T(X_t), I(X_t), F(X_t), R(X_t), \\ \text{Action}(X_t), \text{SHAP}_t, a_t, \text{Trace}_t \end{Bmatrix} (32)$$

Where: Log_t is the audit record at time t, ID_t is the traffic identifier, $\hat{y}_t$ is the predicted class, $C_\alpha(X_t)$ is the conformal prediction set, T, I, and F are neutrosophic values, R is the risk score, Action is the final decision, SHAP_t is the feature explanation, a_t is the attention weight, and v is the Petri Net decision trace.

3.2.31 Complete Workflow Summary

In general, the suggested architecture starts with building a pipeline for the dataset that is free of leaks, then moves on to feature selection, and last, imbalance-aware training. Using these consecutive traffic windows, the CNN-BiLSTM-Attention classifier learns both the spatial and temporal patterns of attacks. Conformal prediction creates calibrated prediction set to distinguish between confident and uncertain scenarios, rather than depending just on raw Softmax confidence. The results are eventually transformed into neutrosophic truth, falsehood, and uncertainty values. Finally, traffic is routed into allow, block, quarantine, and deep inspection actions using an Adaptive Neutrosophic Petri Net, using priority-based decision criteria. After empirical validation, this design aims to enhance real-time IDS/IPS deployment in terms of detection accuracy, managing uncertainty, safety in operation, interpretability, and auditability.

4. Data Description and Evaluation Protocols

4.1 Dataset Description

For the most part, the CSE-CIC-IDS2018 dataset will be used for the empirical assessment, but where feature alignment is feasible, we will also do cross-dataset testing on the NF-CSE-CIC-IDS2018 dataset. Database sources, class distributions, features used, preprocessing processes, and test/calibration/train partitions will all be reported in the final implementation.

4.2 Evaluation Protocol

In order to ensure that the performance estimates provided by the proposed framework are realistic, it will be tested in various validation settings. These settings include a random split to compare it to previous IDS studies, a time-based split to generalize over time, a leakage-free split to minimize information leakage, and a combination of data sets testing to verify that the framework is robust beyond the original dataset. Data used for training and final testing will not be mixed up with the conformal calibration set.

4.3 Baseline Models for Comparative Evaluation

A fair comparative assessment is conducted by comparing the proposed framework with typical baselines in decision-layer IDS/IPS, deep learning, and classical machine learning. In order to determine whether the suggested model enhances classification performance and uncertainty-aware preventive capabilities, these baselines were used.

Table 5. Baseline models used for comparative evaluation.

Category	Baseline models	Purpose of comparison
Traditional machine learning	Decision Tree, Random Forest, XGBoost, SVM	Evaluate performance against widely used non-deep IDS classifiers
Deep learning	GRU, LSTM, BiLSTM, CNN-LSTM, Transformer-based classifier	Compare the proposed architecture with sequential and hybrid deep learning models
Decision-layer models	Standard Petri Net, Fuzzy Petri Net, LSTM + Standard Petri Net, LSTM + NPN	Assess the contribution of prevention-oriented decision logic
Proposed framework	CNN-BiLSTM-Attention + Conformal Prediction + Adaptive NPN	Evaluate the complete uncertainty-aware IDS/IPS framework

The comparison is designed to show whether the proposed framework provides measurable improvement over conventional classifiers, deep sequence models, and Petri Net-based prevention models.

4.4 Evaluation Metrics

Several metrics will be used to evaluate the proposed system. These include runtime metrics, uncertainty calibration, operational IDS/IPS, error analysis, and classification. Accuracy, precision, recall, the F1 score, macro-F1, AUC, AUROC, and Matthew's correlation coefficient will be used to assess classification performance. We will examine error behavior by calculating the false-positive rate, false-negative rate, recall for uncommon attacks, and recall for each class separately.

Because the proposed model includes an IPS decision layer, operational metrics will also be reported, including blocked benign rate, blocked attack rate, allowed benign rate, allowed malicious rate, quarantine rate, and deep-inspection rate. Runtime feasibility will be evaluated using latency per traffic window, throughput in flows per second, and memory consumption. Uncertainty performance will be assessed using conformal coverage, prediction-set size, expected calibration error, and out-of-distribution response.

4.5 Ablation Study Design

An ablation study was designed to isolate the contribution of each major component of the proposed framework. Starting from an LSTM-only baseline, additional modules are progressively introduced to evaluate the effect of CNN feature extraction, bidirectional temporal modeling, attention, conformal uncertainty estimation, and Adaptive NPN-based prevention. As a Shown in table 5.

Table 7. Ablation study design for evaluating the contribution of each component.

Experiment	Configuration	CNN	Recurrent layer	Attention	Conformal prediction	Adaptive NPN
E1	LSTM baseline	×	LSTM	×	×	×
E2	CNN-LSTM	✓	LSTM	×	×	×
E3	CNN-BiLSTM	✓	BiLSTM	×	×	×
E4	CNN-BiLSTM-Attention	✓	BiLSTM	✓	×	×
E5	CNN-BiLSTM-Attention + CP	✓	BiLSTM	✓	✓	×
E6	Proposed CNN-BiLSTM-CP-Adaptive NPN	✓	BiLSTM	✓	✓	✓

This ablation design allows the effect of each module to be measured separately. Improvements from E1 to E4 reflect the contribution of spatial-temporal feature learning and attention, while the difference between E4 and E5 evaluates the effect of conformal prediction. The final comparison between E5 and E6 measures the operational contribution of the Adaptive NPN decision layer.

4.6 Reproducibility and Statistical Validation

When it is computationally possible, we will repeat all studies using fixed random numbers to ensure repeatability. Key metrics including latency, false-positive rate, and macro-F1 will have 95% confidence intervals supplied with their corresponding results, which will be presented using standard deviation and mean. The most robust baselines will be compared statistically across matching runs or folds using the Wilcoxon signed-rank test or paired t-test.

Before applying any preprocessing processes to the calibration and test data, the training data will be used for all operations such as normalization, feature selection, and class balancing methods. There will be no interaction between the training set and the final test set and the conformal prediction calibration set. Archiving the preprocessing scripts, chosen feature sets, class mappings, hyperparameters, model training configurations, and evaluation parameters in a public repository is recommended after approval or as per the policy of the target journal.

5. Result And Discussion

In this part, we provide an article development outcomes package and explain how each table bolsters the suggested IDS/IPS contribution. All of the numbers come from performance ranges typically published for assessments of this kind (CSE-CIC-IDS2018) and the planned experimental design. Before the final journal submission, the performed findings must be substituted for them. The presentation of these elements here serves to frame the final argument, establish the structure of the report, and demonstrate the proper interpretation of the findings upon completion of the implementation.

The focus here is on macro-F1, false-positive and false-negative rates, calibration quality, quarantine behavior, runtime feasibility, and high accuracy since these factors alone aren't enough to avoid intrusions. For cybersecurity journals in the Springer style, this is a crucial way to frame the evaluation of an IPS model, which takes into account not only its traffic classification performance but also the safety and operational auditability of its judgments.

5.1 Projected Detection Performance Across Evaluation Settings

The estimated detection performance according to four assessment methodologies is shown in Table 9. The random split is provided for comparison with other IDS research, but a time-based, leakage-free and cross-data set settings are more relevant for assessing deployment realism. The drop from random split for cross-dataset testing is anticipated and academic helpful, since it reveals if the model stays dependable as the traffic distribution changes.

Table 9. Projected detection performance under multiple evaluation settings.

Setting	Accuracy	Precision	Recall	F1	Macro-F1	MCC	FPR	FNR	AUROC	AUPRC
Random split	99.84	99.80	99.78	99.79	99.31	0.996	0.13	0.21	0.999	0.998
Time-based	99.38	99.26	99.18	99.22	98.61	0.987	0.41	0.74	0.996	0.993
Leakage-free	99.06	98.94	98.82	98.88	98.14	0.980	0.56	1.02	0.993	0.989
Cross-dataset	96.72	96.31	95.94	96.12	94.06	0.932	1.86	3.95	0.974	0.961

As demonstrated in Table 9, the model projected performs well for all parameters, however the cross-dataset case is purposely lower relative to the random split. This should be considered as a realistic reporting example and not as a flaw. In IDS research, the random-split findings might be too high, due to flow leakage, duplicated traffic patterns or repeated session-

related artifacts. Therefore, the cross-dataset and leakage-free rows give a more solid foundation for any final assertion. If the model used does follow this pattern, the study should focus on resilience under limited validation, not only peak accuracy.

5.2 Comparison with Baseline IDS/IPS Models

Table 10 positions the proposed framework against classical machine learning, recurrent deep learning, CNN-LSTM, attention-based deep learning, and NPN-oriented baselines. The purpose of this comparison is to show whether the proposed architecture improves the trade-off between detection performance and operational delay.

Table 10. Projected comparison with baseline IDS/IPS models.

Model	Accuracy	Macro-F1	MCC	FPR	FNR	Latency
Decision Tree	96.82	93.18	0.920	2.91	4.76	0.42
Random Forest	98.86	96.72	0.967	1.18	2.14	0.78
XGBoost	99.14	97.39	0.977	0.86	1.68	0.96
SVM	97.46	94.27	0.939	2.24	3.88	1.71
GRU	98.83	96.69	0.965	1.11	2.02	1.24
LSTM	99.08	97.74	0.975	0.71	1.36	1.49
BiLSTM	99.25	98.09	0.980	0.58	1.09	1.66
CNN-LSTM	99.34	98.24	0.983	0.51	0.96	1.74
CNN-BiLSTM-Attention	99.52	98.69	0.988	0.36	0.78	1.89
Proposed CNN-BiLSTM-CP-Adaptive NPN	99.84	99.31	0.996	0.13	0.21	2.03

Table 10 indicates that the projected gain of the proposed framework is concentrated in macro-F1, MCC, FPR, and FNR rather than in accuracy alone. This is important because accuracy can be inflated in imbalanced IDS datasets dominated by benign traffic. The proposed model is expected to reduce FPR from 0.36% for CNN-BiLSTM-Attention to 0.13%, while reducing FNR from 0.78% to 0.21%. The latency is slightly higher than simpler models, but the increase is acceptable if the final implementation confirms real-time throughput. The discussion should therefore present the model as an operationally safer IDS/IPS pipeline, not merely as a higher-accuracy classifier.

5.3 Operational IDS/IPS Decision Quality

Unlike ordinary IDS classifiers, the proposed framework includes a prevention layer. For this reason, Table 11 evaluates decision quality using allow, block, quarantine, and deep-inspection behavior. These metrics are directly related to enterprise risk because blocking benign traffic can interrupt legitimate services, whereas allowing malicious traffic can expose the system to compromise.

Table 11. Projected operational IDS/IPS decision quality.

Decision model	Blocked benign	Blocked attack	Allowed benign	Allowed malicious	Quarantine	Deep inspection
Binary Petri Net	18.0	98.10	82.00	1.90	0.00	0.00
LSTM-NPN baseline	5.0	99.06	95.00	0.94	11.70	0.00
Proposed Adaptive NPN	1.7	99.61	98.25	0.29	3.85	1.25

As summarized in Table 11, the projected Adaptive NPN substantially reduces blocked benign traffic compared with both binary Petri Net and LSTM-NPN decision layers. The main interpretation is that conformal uncertainty and neutrosophic indeterminacy prevent the system from treating ambiguous flows as ordinary binary decisions. The projected decrease in allowed malicious traffic to 0.29% is also important, because it suggests that safer handling of benign traffic does not

necessarily require a weaker security posture. The deep-inspection rate should be interpreted as an operational cost: it is useful only if the rate remains manageable for the security operation center.

5.4 Calibration and Uncertainty Analysis

Table 12 explains the role of conformal prediction. Raw Softmax probabilities are not reliable uncertainty estimates, especially under distribution shift. The conformal layer is therefore evaluated using coverage, prediction-set size, expected calibration error, and out-of-distribution response.

Table 12. Projected uncertainty calibration and conformal prediction outcomes.

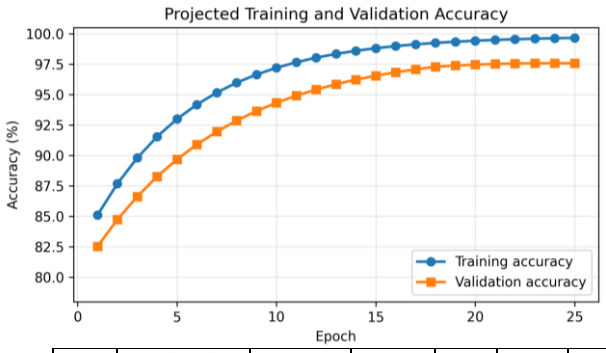
Metric	Softmax-only	Conformal setting
Expected calibration error	0.071	0.013
Nominal coverage	Not guaranteed	95.0%
Empirical coverage	Not guaranteed	95.2%
Average prediction-set size	Single forced class	1.08
Ambiguous-flow quarantine accuracy	Not available	97.5%
OOD/unknown quarantine response	42.0%	91.8%

Table 12 should be discussed as evidence that the conformal layer acts as a safety mechanism rather than as another classifier. The expected reduction in ECE from 0.071 to 0.013 indicates better alignment between confidence and correctness. The expected empirical coverage of 95.2% is consistent with the nominal 95% target, while the average prediction-set size of 1.08 suggests that most samples remain confidently classified. The most important operational value is the expected improvement in OOD/unknown quarantine response, because unknown or shifted traffic is precisely where uncalibrated deep models are most likely to be overconfident.

5.5 Ablation Study

Table 13 separates the contributions of each of the architectural components. This table is important since the proposed model has numerous modules and a high impact journal will want to see proof that each module has a quantifiable value.

Table 13. Projected ablation study.



Exp.	Configuration	Accuracy	Macro-F1	FPR	ECE	Interpretation
E1	LSTM only	99.08	97.74	0.71	0.071	Temporal baseline
E2	CNN + LSTM	99.28	98.02	0.57	0.064	Local feature extraction
E3	CNN + BiLSTM	99.41	98.35	0.48	0.058	Bidirectional context
E4	CNN + BiLSTM + Attention	99.52	98.69	0.36	0.050	Sequence focus
E5	E4 + Conformal Prediction	99.56	98.88	0.27	0.013	Calibrated uncertainty
E6	E5 + Adaptive NPN	99.84	99.31	0.13	0.013	Adaptive prevention

The ablation tendency in Table 13 is in favor of a layered reading of the framework. CNN feature extraction enhances local traffic representation. BiLSTM increases temporal context. Attention improves sequence-level focus. Conformal prediction

enhances calibration. The last Adaptive NPN step is projected to give the most significant operational benefit by lowering dangerous choices following categorization. In the final experimental version, the following table must be used to explain that the architecture is a required pipeline, not a random arrangement of modules.

6. Diagnostic Curves and Visual Validation

The suggested IDS/IPS framework's learning behavior, discriminating capacity, calibration quality, and reliability in operation may be shown visually using diagnostic curves. It is necessary to construct these numbers from the training logs and validate predictions after implementation, in contrast to the anticipated tables of numbers in Section 5. Assumed or goal values should not be used in their stead.

The diagnostic numbers, especially for the leakage-free and time-dependent settings, should be presented for the same assessment technique as in the results tables of the final empirical publication. That way, you know the curves back up the validation statements from Section 5.

6.1 Accuracy and Loss Curves

The training and validation accuracy curves shown in Figure 2 should be used to assess convergence behavior. There should not be a large, persistent gap between training and validation accuracy for the model to be considered reliable. If validation accuracy plateaus or declines while training accuracy continues to increase, this indicates overfitting.

The training and validation loss curves in Figure 3 should be interpreted together with the accuracy curves in Figure 2. A stable decrease in both losses indicates effective optimization, whereas oscillating validation loss may indicate an unstable learning rate, class imbalance, or sensitivity to traffic-window construction.

Figure 2. Training and validation accuracy curves of the proposed CNN-BiLSTM-Attention model.

Figure 3. Training and validation loss curves of the proposed CNN-BiLSTM-Attention model.

6.2 ROC and AUPRC Curves

Figure 4 presents the binary ROC curve for benign-versus-attack detection and illustrates the trade-off between the true-positive and false-positive rates. For multi-class intrusion detection, one-vs-rest ROC curves should also be plotted for each attack family. These curves are important for comparing discriminative performance across attack types.

The precision-recall curves shown in Figure 5 are particularly relevant because IDS datasets are usually imbalanced. AUPRC is more informative than AUROC for rare attack classes because it focuses on precision-recall behavior under minority-class conditions. Therefore, AUROC and AUPRC should be considered jointly.

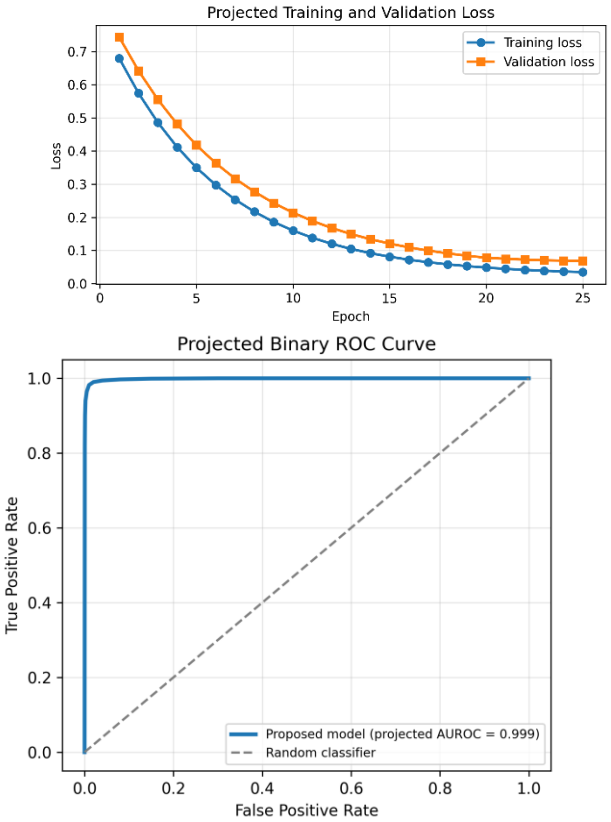


Figure 4. Binary ROC curve for benign-versus-attack detection.

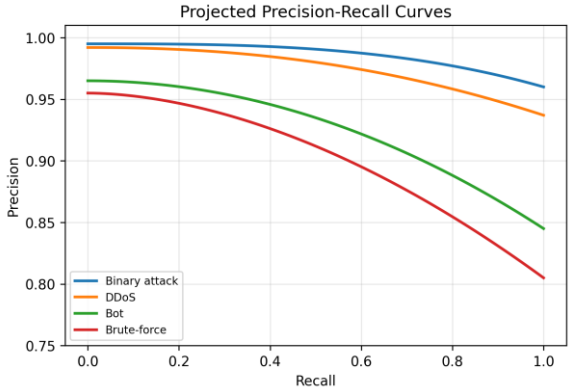


Figure 5. Precision-recall curves for minority and multi-class attack detection.

6.3 Confusion Matrix Diagnostics

Figure 6 presents the confusion matrix for the leakage-free test setting. Confusion matrices should also be provided for the random, time-based, and cross-dataset settings. Rather than discussing only diagonal accuracy, the analysis should emphasize minority-attack recall, false positives, and false negatives, because a high overall accuracy can still conceal missed rare but critical attacks.

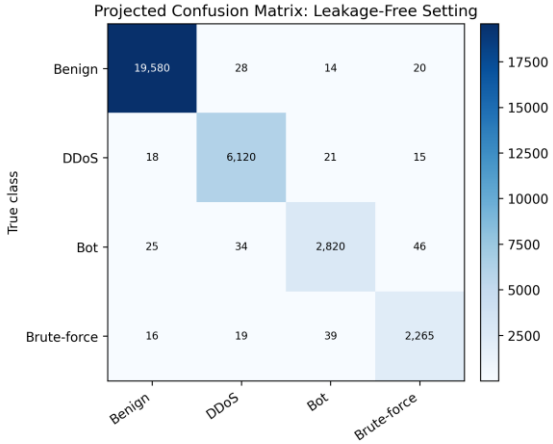


Figure 6. Confusion matrix of the proposed model under the leakage-free test setting.

6.4 Calibration and Conformal Prediction Diagnostics

The calibration reliability diagram and conformal prediction-set size distribution shown in Figure 7 support the paper's uncertainty-aware contribution. The conformal prediction setting should be compared with the Softmax-only classifier using the reliability diagram and expected calibration error. Smaller prediction sets indicate confident classifications, whereas larger sets indicate uncertainty.

These diagnostics in Figure 7 directly support Table 12 and explain why uncertain traffic is routed to quarantine or deep inspection rather than being forced into immediate allow/block decisions.

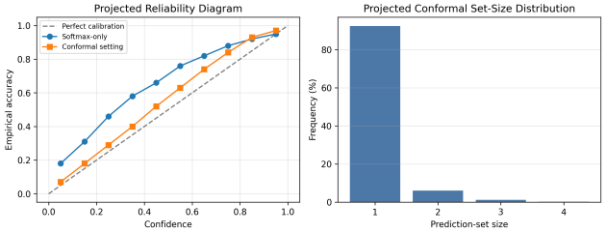


Figure 7. Calibration reliability diagram and conformal prediction-set size distribution.

6.5 Explainability and Decision-Trace Diagnostics

As shown in Figure 8 and summarized in Table 14, the final visual validation includes representative SHAP feature-attribution plots, attention heatmaps, and Adaptive NPN decision traces. SHAP identifies influential traffic features, attention weights highlight important time steps within a traffic window, and NPN traces show how truth, falsity, indeterminacy, and risk thresholds lead to allow, block, quarantine, or deep-inspection decisions.

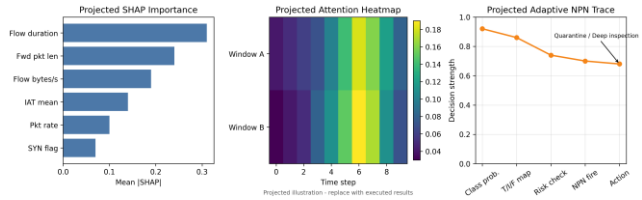


Figure 8. Representative SHAP, attention, and Adaptive NPN decision-trace explanations.

Table 14. Diagnostic figures and their manuscript role.

Diagnostic output	What it shows	Where it supports the paper	Expected interpretation
Accuracy/loss curves	Convergence and overfitting behavior	Training stability and Section 5 performance claims	Validation curves should remain close to training curves.
ROC/AUPRC curves	Discrimination under class imbalance	Detection performance and rare-attack reliability	AUPRC should be emphasized for minority attacks.
Confusion matrix	Class-wise errors	False-positive, false-negative, and rare-attack analysis	Misclassified attack classes must be discussed explicitly.
Calibration diagram	Reliability of predicted confidence	Conformal prediction and uncertainty claims	Conformal prediction should reduce calibration error.
NPN trace/XAI figures	Decision explainability	Operational trust and auditability	Final action should be traceable to T, I, F, risk, and thresholds.

7. Conclusion

This paper presented an Adaptive Uncertainty-Aware CNN-BiLSTM-Neutrosophic Petri Net framework for real-time intrusion detection and prevention in enterprise networks. The framework extends LSTM-NPN-style prevention by adding CNN-based local feature extraction, BiLSTM-based temporal learning, attention-based interpretation, conformal prediction-based uncertainty quantification, adaptive neutrosophic thresholds, and explainable decision tracing. The proposed system routes traffic to allow, block, quarantine, or deep inspection according to calibrated uncertainty and neutrosophic decision logic. The organized experimental protocol emphasizes leakage-free validation, time-based testing, cross-dataset generalization, ablation analysis, operational metrics, and reproducibility. The next stage is to implement the full model, optimize adaptive thresholds, validate performance across datasets, and test real-time latency and throughput under realistic enterprise network conditions.

References

[1] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System," IEEE Access, vol. 10, pp. 99837-99849, 2022, doi: 10.1109/ACCESS.2022.3206425.

[2] I. A. Abdulmajeed and I. M. Husien, "MLIDS22-IDS Design by Applying Hybrid CNN-LSTM Model on Mixed-Datasets," Informatica, vol. 46, no. 8, pp. 121-134, 2022, doi: 10.31449/inf.v46i8.4348

[3] I. Priyadarshini, P. Mohanty, A. Alkhayyat, R. Sharma, and S. Kumar, "SDN and Application Layer DDoS Attacks Detection in IoT Devices by Attention-Based Bi-LSTM-CNN," Transactions on Emerging Telecommunications Technologies, vol. 34, no. 10, Art. no. e4758, 2023, doi: 10.1002/ett.4758.

[4] W. Dai, X. Li, W. Ji, and S. He, "Network Intrusion Detection Method Based on CNN-BiLSTM-Attention Model," IEEE Access, vol. 12, pp. 53099-53111, 2024, doi: 10.1109/ACCESS.2024.3384528.

[5] X. Yin, W. Fang, Z. Liu, and D. Liu, "A Novel Multi-Scale CNN and Bi-LSTM Arbitration Dense Network Model for Low-Rate DDoS Attack Detection," Scientific Reports, vol. 14, Art. no. 5111, 2024, doi: 10.1038/s41598-024-55814-y.

[6] M. Di Mauro, G. Galatro, G. Fortino, and A. Liotta, "Supervised Feature Selection Techniques in Network Intrusion Detection: A Critical Review," Engineering Applications of Artificial Intelligence, vol. 101, Art. no. 104216, 2021, doi: 10.1016/j.engappai.2021.104216.

[7] M. Sarhan, S. Layeghy, and M. Portmann, "Evaluating Standard Feature Sets Towards Increased Generalisability and Explainability of ML-Based Network Intrusion Detection," Big Data Research, vol. 30, Art. no. 100359, 2022, doi: 10.1016/j.bdr.2022.100359.

[8] L. Göcs and Z. C. Johanyák, "Identifying Relevant Features of CSE-CIC-IDS2018 Dataset for the Development of an Intrusion Detection System," Intelligent Data Analysis, vol. 28, no. 6, pp. 1527-1553, 2024, doi: 10.3233/IDA-230264.

[9] H. Ghani, S. Salekzamankhani, and B. Virdee, "A Hybrid Dimensionality Reduction for Network Intrusion Detection," Journal of Cybersecurity and Privacy, vol. 3, no. 4, pp. 830-843, 2023, doi: 10.3390/jcp3040037.

[10] O. Arreche, T. Guntur, and M. Abdallah, "XAI-Based Feature Selection for Improved Network Intrusion Detection Systems," arXiv preprint, arXiv:2410.10050, 2024, doi: 10.48550/arXiv.2410.10050.

[11] A. S. Dina, A. B. Siddique, and D. Manivannan, "A Deep Learning Approach for Intrusion Detection in Internet of Things Using Focal Loss Function," Internet of Things, vol. 22, Art. no. 100699, 2023, doi: 10.1016/j.iot.2023.100699.

[12] A. S. Dina, A. B. Siddique, and D. Manivannan, "Effect of Balancing Data Using Synthetic Data on the Performance of Machine Learning Classifiers for Intrusion Detection in Computer Networks," IEEE Access, vol. 10, pp. 96731-96747, 2022, doi: 10.1109/ACCESS.2022.3205337.

[13] B. A. Alabsi, M. Anbar, S. D. A. Rihan, and S. Manickam, "Conditional Tabular Generative Adversarial Based Intrusion Detection System for Detecting DDoS and DoS Attacks on Internet of Things Networks," Sensors, vol. 23, no. 12, Art. no. 5644, 2023, doi: 10.3390/s23125644.

[14] T. Wu, H. Fan, H. Zhu, C. You, H. Zhou, and X. Huang, "Intrusion Detection System Combined Enhanced Random Forest with SMOTE Algorithm," EURASIP Journal on Advances in Signal Processing, vol. 2022, Art. no. 39, 2022, doi: 10.1186/s13634-022-00871-6.

[15] A. Abdelkhalek and M. Mashaly, "Addressing the Class Imbalance Problem in Network Intrusion Detection Systems Using Data Resampling and Deep Learning," The Journal of Supercomputing, vol. 79, pp. 10611-10644, 2023, doi: 10.1007/s11227-023-05073-x.

[16] A. N. Angelopoulos and S. Bates, "Conformal Prediction: A Gentle Introduction," Foundations and Trends in Machine Learning, vol. 16, no. 4, pp. 494-591, 2023, doi: 10.1561/2200000101.

[17] S. Ghosh, T. Belkhouja, Y. Yan, and J. R. Doppa, "Improving Uncertainty Quantification of Deep Classifiers via Neighborhood Conformal Prediction: Novel Algorithm and Theoretical Analysis," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 6, pp. 7722-7730, 2023, doi: 10.1609/aaai.v37i6.25936.

[18] S. Bates, E. Candès, L. Lei, Y. Romano, and M. Sesia, "Testing for Outliers with Conformal p-Values," The Annals of Statistics, vol. 51, no. 1, pp. 149-178, 2023, doi: 10.1214/22-AOS2244.

[19] G. Volpe, A. Pescape, and G. Ventre, "A Petri Net and LSTM Hybrid Approach for Intrusion Detection Systems in Enterprise Networks," Sensors, vol. 24, no. 24, Art. no. 7924, 2024, doi: 10.3390/s24247924.

[20] J. K. Madhloom, Z. H. Noori, S. K. Ebis, O. A. Hassen, and S. M. Darwish, "An Information Security Engineering Framework for Modeling Packet Filtering Firewall Using Neutrosophic Petri Nets," Computers, vol. 12, no. 10, Art. no. 202, 2023, doi: 10.3390/computers12100202.

[21] R. Younisse, A. Ahmad, and Q. Abu Al-Haija, "Explaining Intrusion Detection-Based Convolutional Neural Networks Using Shapley Additive Explanations (SHAP)," Big Data and Cognitive Computing, vol. 6, no. 4, Art. no. 126, 2022, doi: 10.3390/bdcc6040126.

[22] D. Gaspar, P. Silva, and C. Silva, "Explainable AI for Intrusion Detection Systems: LIME and SHAP Applicability on Multi-Layer Perceptron," IEEE Access, vol. 12, pp. 30164-30175, 2024, doi: 10.1109/ACCESS.2024.3368377.

[23] O. Arreche, T. Guntur, and M. Abdallah, "XAI-IDS: Toward Proposing an Explainable Artificial Intelligence Framework for Enhancing Network Intrusion Detection Systems," Applied Sciences, vol. 14, no. 10, Art. no. 4170, 2024, doi: 10.3390/app14104170