International Research Journal of Multidisciplinary Scope (IRJMS), 2026;7(2):1711-1732 IRJMS

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2026.v07i02.09007

Neuro-hybrid Adaptive Cybersecurity Orchestration for Real-time Zero-day Resilience and Intelligent Threat Response

Kanukanti Sindhu*, Muntha Raju, Yerram Sneha

Computer Science and Engineering Department, Nalla Malla Reddy Engineering College, Divya Nagar, Kachivani Singaram, Ghatkesar, Hyderabad, Telangana, India. *Corresponding Author's Email: sindhukanukantii@gmail.com

Abstract

The fast-changing nature of cyberattacks, in particular zero-day attacks, puts enormous pressure on the security and robustness of current network infrastructures. Conventional IDS systems are unable to cope with the changes in mechanisms of attack and suffer from high rates of false positives. To overcome these deficiencies, this paper presents a Neuro-Hybrid Adaptive Cyber Security Orchestration (NHACO) method that encapsulates different machine learning techniques into an adaptive ensemble framework. The architecture integrates Logistic Regression, K-Nearest Neighbors, Naïve Bayes; Support Vector Machine, Decision Tree and Random Forest classifiers to benefit from the complementary behavior of these techniques in order to detect intrusion with a good balance. The NHACO integrates a feedback-based retraining strategy and real-time visualization layer by means of Streamlit, facilitating adaptive learning of network dynamics and ease-of-use interface for visual inspection. The model was evaluated using the benchmark KDD Cup 1999 and NSL-KDD datasets and obtained an overall accuracy of 98.46%, with better performance than all single classifiers in precision, recall, and F1-score. Furthermore, the low latency and communication overhead of the system also demonstrated that it could be deployed in real-time for IoT/cloud/edge environment. It is evident that the NHACO framework offers as a promising and intelligent, scalable adaptive cybersecurity solution with high accuracy, and resistance in monitoring either known or zero-day intrusions.

Keywords: Adaptive Feedback Learning, Cybersecurity, Intrusion Detection System, Machine Learning Ensemble, Neuro-Hybrid Zero-Day Attack Detection.

Introduction

The rapid pace of digital transformation across industries, enterprises, and governments has significantly expanded the global cyber-attack surface (1). Modern networked environments, characterized by billions of interconnected devices and high-volume traffic flows, are increasingly exposed to sophisticated and dynamic security threats (2). Among these, zero-day attacks-vulnerabilities exploited before any known defense or patch is available-pose severe challenges due to their stealthy and unpredictable nature. Traditional rule-based and signature-based Intrusion Detection Systems (IDS) are largely ineffective against such attacks, as they depend on prior knowledge of known threat patterns (3). Consequently, there is a growing need for intelligent, adaptive, and proactive cybersecurity mechanisms capable of learning evolving attack behaviors in real time (4).

Conventional IDS approaches rely heavily on predefined attack signatures and static pattern matching techniques, which limits their scalability and detection capability in modern distributed environments such as IoT, edge computing, and cloud networks (5, 6). With the exponential growth of network data, these systems struggle with issues related to processing speed, detection accuracy, and false alarm rates. To overcome these limitations, machine learning (ML) and deep learning (DL) techniques have emerged as promising solutions for identifying behavioral and statistical anomalies in network traffic (7). These approaches enable models to learn from large-scale, continuously evolving datasets and detect deviations from normal behavior, including previously unseen or zero-day attacks (8).

Recent advancements in AI-driven intrusion detection highlight the effectiveness of hybrid learning approaches, which combine multiple models to improve generalization and reduce false positives (9–12). Deep learning architectures such as Convolutional Neural networks (CNNs) and

(Received 05th November 2025; Accepted 15th April 2026; Published 28th April 2026)

recurrent Neural Networks (RNNs) have demonstrated strong performance in capturing complex spatial and temporal attack patterns from real-time data streams (13, 14). Similarly, ensemble learning methods such as Random Forests (RF) and Gradient Boosting Machines (GBM) enhance decision robustness by aggregating predictions from multiple classifiers (15). Despite these advancements, existing solutions still face challenges related to interpretability, automation, and dynamic coordination between detection and response mechanisms. Many current models lack the ability to integrate multi-level decision-making processes required for real-world Cyber Security Incident Response Team (CSIRT) operations (16-18).

To address these challenges, this work proposes a Neuro-Hybrid Adaptive Cybersecurity Orchestration (NHACO) framework for real-time intrusion detection and response. The framework integrates machine learning and neural-based techniques within a unified architecture to enhance zero-day attack detection capabilities. The term "neuro-hybrid" refers to the integration of multiple learning paradigms, including ensemble-based methods and neural models, supported by adaptive feedback mechanisms. The proposed system is designed as a modular pipeline consisting of data collection, feature preprocessing, hybrid model training, anomaly detection, and automated threat response. This orchestration enables continuous learning and dynamic adaptation to emerging threat patterns while improving detection accuracy and reducing response latency (19–22).

The effectiveness of the proposed framework is evaluated using benchmark datasets such as KDD Cup 1999 and NSL-KDD, which are widely used in intrusion detection research. Experimental results demonstrate that the NHACO framework outperforms traditional IDS models in terms of accuracy, precision, recall, and F1-score. Furthermore, the system exhibits improved robustness against unseen attack categories and achieves a significant reduction in false positive rates. These findings indicate the potential of neuro-hybrid adaptive models in enabling next-generation autonomous and self-healing cybersecurity systems (23–26).

The main contributions of this research are summarized as follows:

(a) A neuro-hybrid adaptive framework integrating multiple machine learning and neural-based techniques for robust zero-day attack detection.
(b) A modular orchestration architecture enabling real-time threat analysis, classification, and automated response.
(c) Comprehensive performance evaluation using benchmark datasets (KDD99, NSL-KDD) to validate scalability, accuracy, and resilience.
(d) Demonstration of continuous learning capability for long-term adaptability in dynamic cyber environments.

Recent research in intrusion detection has increasingly focused on improving adaptability, scalability, and interpretability of IDS models. Machine learning-based IDS approaches have been widely adopted due to their ability to process large-scale and imbalanced network data efficiently. Deep learning models, in particular, have demonstrated superior performance in detecting complex and high-dimensional attack patterns, especially in IoT and cloud-based environments (1–4). Techniques such as data augmentation and advanced feature selection have further enhanced detection capabilities in large-scale networks (5-8).

Hybrid and deep learning-based IDS architectures have also gained attention for their ability to improve detection accuracy and handle unseen attack patterns. For instance, CNN-based models with regularization techniques have achieved promising results in zero-day attack detection, while Generative Adversarial Networks (GANs) have been used to enhance training datasets and improve generalization. Similarly, recurrent models and graph-based approaches have been proposed to capture temporal and relational dependencies in network traffic, enabling more accurate detection in dynamic environments.

In addition to detection performance, recent studies emphasize the importance of interpretability and efficiency in IDS design. Explainable AI (XAI) techniques have been incorporated to improve transparency and trust in model decisions, particularly in critical applications such as IoT and industrial networks (22). Energy-efficient IDS frameworks have also been developed to reduce computational overhead while maintaining detection performance in resource-constrained environments. These advance-

ments highlight the need for intelligent, scalable, and interpretable intrusion detection solutions.

Overall, the integration of adaptive learning, hybrid modeling, and real-time orchestration represents a promising direction for next-generation cybersecurity systems. The proposed NHACO framework builds upon these advancements by combining detection accuracy, adaptability, and automation into a unified architecture capable of addressing evolving cyber threats in complex network environments (23–26).

Methodology

The developed scheme, Neuro-Hybrid Adaptive Cybersecurity Orchestration (NHACO) is a smart multi-tier defense mechanism which discovers and prevents zero-days attacks using joint reinforcement of multiple machine-learning tools as well as an adaptive feedback control. The approach is based on six stages of a data-driven pipeline, including network data gathering, pre-processing, feature extraction and engineering, hybrid modeling learning and performance testing and response to threat in real time. All of these layers turn raw network traffic into structured awareness that can support dynamic intrusion detection. The overall workflow of the proposed Neuro-Hybrid Adaptive Cybersecurity Orchestration (NHACO) framework, including data collection, pre-processing, feature extraction, hybrid model training, and real-time threat response, is illustrated in Figure 1.

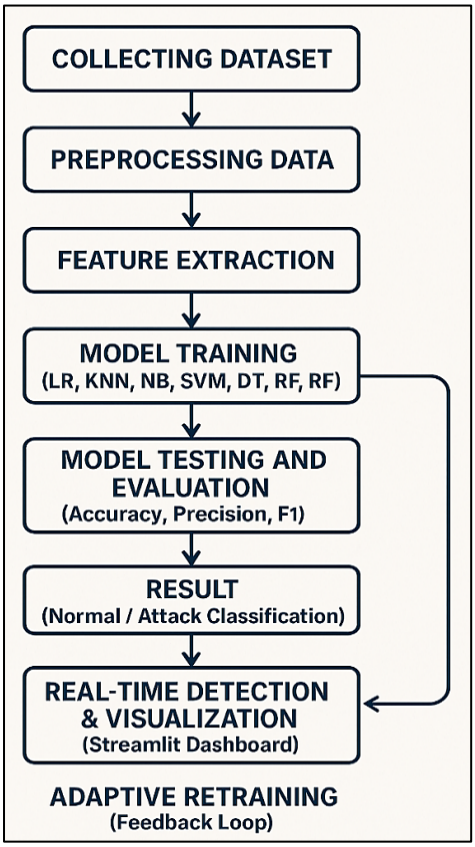

Figure 1: Workflow of the Proposed System

Step 1 – Data Collection

Quality and diversity of the input data are the cornerstones of the NHACO-model. In this work, the data collection phase was carefully engineered to make sure that the training and testing sets generously represent both normal network behavior, and possible attacks. The KDD Cup 1999 and NSL-KDD data set were selected as the main datasets due to their cohesion and popularity as a benchmark in network intrusion detection research. The original KDD Intrusion dataset (KDD Cup 1999) is a large set of connection records created from a simulation of a military network. Each record comprises forty-one features which cover details on type (e.g., protocol), service, flag, number of bytes and time to transmit them as well as label for detection (to be normal or specific attack class such as; DoS (Denial-of-Service): one

machine used to attack another, R2L (Remote-to-Local): unauthorized remote user send the request to unauthorized local user that trying gain admin rights.)egalitarian-like1) or U2R(User-to-Root): network packets monitor to cross Admin privileged-ges). These labelled samples represent a strong supervised-learning source for initial machine learning model training.

Nevertheless, the KDD Cup 1999 dataset has a high redundancy and the classes are imbalanced; this may possibly introduce bias on learning algorithms towards majority classes. To tackle these issues, its advanced version NSL-KDD was used. The NSL-KDD dataset is an improved version compared to the earlier one, by removing duplicate and inconsistent records, resulting in relatively more balanced distributions of normal and abnormal traffics. This enhancement makes the models generalize better and detect more subtle differences in attack behavior.

In addition, with these state-of-the-art datasets, the research also integrated real-time data simulation environment to assess the system's robustness in a live network. The packet-level data was extracted from active network interfaces by using instruments like Wireshark and Nmap (IP addresses, port numbers, protocols, sizes of the transmitted packets). This applied simulation proved possible to create a real traffic with true latency/bandwidth usage/packet makeup variability and made the hybrid detection model more robust.

After downloaded, all the traffic data (with both benchmark and real-time) were combined as flat comma-separated values (.csv) format. This singular representation allows for optimized preprocessing, as well as compatibility with python-based analytical libraries like Pandas and NumPy. This large data-collection process provides a reliable source for accurate, scalable and real-time detection of intrusion in heterogeneous networked environment based NHACO framework.

The real-time network traffic captured using Wireshark and Nmap was used only to demonstrate the practical deployment capability of the NHACO framework and was not incorporated into the training or testing datasets. All machine learning models were trained and evaluated exclusively using the labelled benchmark datasets (KDD Cup 1999 and NSL-KDD) to ensure experimental consistency and comparability with existing intrusion detection studies.

For demonstration purposes, the captured packets were converted into a structured format by extracting network flow attributes such as protocol type, source and destination bytes, connection duration, and service type. These attributes were then mapped to the corresponding feature representation used in the KDD-style dataset to enable compatibility with the trained model during real-time inference. Since the captured traffic was not labelled with attack categories, it was used only for real-time monitoring and visualization within the Streamlit interface rather than for model training or performance evaluation.

Step 2 – Data Preprocessing

After the data is collected, the second critical phase in NHACO framework is pre-processing of data to make it clean and organized so that we could use machine learning on it. Network intrusion datasets are typically noisy justified by the fact that they contain irrelevant information and inconsistencies due to the heterogeneity of network events, and fluctuations in packet transmission. If unmitigated, such irregularities may cause model training to become inaccurate and degrade performance. Thus, a formalized preprocessing pipeline was designed and applied in order to improve the data quality and reliability before it was used as input on the hybrid learning models.

Data cleaning was conducted initially to delete the incomplete, redundant, or duplicate records in the dataset. In the KDD and NSL-KDD datasets, it is common to see duplicated instances due to recurring attacks or equivalent capture sessions that might introduce bias over algorithm results as well as strategic information for an attacker. Preprocessing the data also facilitated reducing noise and overhead in computations, and ensured that each record contained only information about distinct network activities. There after missing/ null values were managed systematically. For the numeric attribute's "duration", "src_bytes" and "dst_bytes", missing values were imputed with the mean of the attribute, while for categorical features such as "protocol_type" or n-service was used to fill in missing values (rounded up). This guarantee that the dataset can be continuous with no artificial biasing.

The subsequent process comprised a categorical attribute encoding in order to convert textual values into a numerical format required by machine-learning algorithms. Most of the features (e.g. “protocol_type” type=” TCP, UDP, ICMP”, and” service” types=” HTTP, FTP, SMTP”) were converted to binary using label encoding and one-hot encoding. By this way, the algorithm was able to read non-numeric information in a mathematically significant manner.

To allow equal weight across all features, the numerical attributes were normalized to a similar range and scaled with values between 0 and 1 using Min–Max scaling. This step was necessary to prevent features with large numerical ranges (e.g., “src_bytes” and “dst_bytes”) from overpowering those of smaller ranges that might be biased in training the model otherwise. The later resulted in all features being equal contributors of the machine-learning models classification decisions, as the normalization process allowed each not to carry different weights.

The class imbalance issue was also dealt with during preprocessing. For intrusion-detection datasets, the former is the case and the normal traffic largely outweighs attack traffics thus that model learning becomes very biased towards majority class. To address it, we used Synthetic Minority Oversampling Technique (SMOTE) to extract synthetic instances of minority attack types. The balancing led to increased detection performance when low-frequency attack types, such as U2R and R2L, were detected resulting in a higher rate of overall accuracy.

For the last step, the fully pre-processed file was split into training and testing subset (80:20). The training set is for fitting and tuning the hybrid models, while the testing set is used to test the transferability and real-time prediction ability. Noise in data was eliminated, and the raw data became stable, well balanced through preprocessing stage for feature selection and model training in following steps. This preprocessing step allowed the NHACO framework to have excellent accuracy, stability and efficiency when handling large-scale, complex network data streams.

Step 3 – Feature Extraction and Selection

After dataset was pre-processed and normalized, the subsequent critical part of NHACO algorithm is feature extraction and selection. The most important feature selection This stage has the main purpose of finding the most significant features in large pool of proposed features which contribute significantly to differentiate between normal and malicious network traffic. The performance, accuracy, and computational efficiency of the machine-learning algorithms highly depend on the quality and relevance of features selected. Because both the two KDD Cup 1999 and NSL-KDD datasets contain 41 attributes, not all of them contribute unique or significant information to characterize zero-day or unknown activities. A systematic technique was thus used to extract and maintain the most relevant features, which contribute to better classification performance, with surplus substance reduced so far as possible.

Feature extraction was done to liberate features in function classes. Basic features are the underlying connection-level properties of the network flow (eg: duration of the connection, type of protocol used, number of bytes from source/destination). These features are of significance in recognizing traffic patterns and distinguishing between malicious and legitimate behavior. The traffic features capture time-based statistics such as how much a host or service was connected to within a window of the observation interval. Such temporal patterns are effective for detecting the big attacks such as Denial-of-Service (DoS) or profound Probe attack that last long time. The content-based features, instead, are obtained from the network packets’ payload and correspond to parameters such as the number of failed login attempts, root access counts or suspicious commands executions. These properties are especially beneficial for detecting user-level and privilege escalation attacks like R2L or U2R.

After extraction, some feature selection algorithms were employed to remove irrelevant and redundant characteristics, which may influence the accuracy of models or computation complexity. The first used filtering method was correlation analysis to eliminate highly correlated attributes which express similar meanings.

The attributes having correlation coefficients more than or equal to 0.85 have been excluded from the model to avoid multicollinearity and for the model stability. Then, Information Gain (IG) was employed to quantify the amount each feature contributes to decreasing uncertainty in classifi-

cation. Features having low information gain value were excluded as they contributed very little to the predictive power. Lastly, Principal Component Analysis (PCA) was used as a dimensionality reduction method to map the high-dimensional data using a smaller set of uncorrelated principal components that kept most of dataset's variation. This made an important impact on computational cost and overfitting.

By this multi-step processing, the size of features was decreased from 41 to around high impact 30 attributes that represent the most discriminative parameters over basic, traffic and content categories. The selected features were able to well describe the inherent behavior of normal connections and malicious ones. For instance, factors like "count," "srv_count," and "dst_host_same_srv_rate" exhibited significant power in discriminating denial-of-service versus probing activities whereas factors such as "logged_in", "root_shell", and "num_failed_logins" were closely associated with privilege- based intrusive activities.

The produced optimal feature subset was able to be timely covered by the hybrid-based machine-learning models without compromising performance. This trade-off between feature compactness and diversity formed the theoretical backbone of the NHACO framework on which robust classifiers with generalization ability to novel dynamic attacks were trained.

Efficient feature selection has been demonstrated to increase the detection capability of applied basic statistical feature-selection strategies to eliminate redundancy and enhance model robustness that guides the design decisions of our preprocessing phase (27, 28).

Step 4 – Model Building and Training

During pre-processing, feature selection and refining of the dataset, NHACO model building and training is the subsequent step. This stage is the heart of the analysis system in that machine learning is utilized to learn patterns of normal and malicious network behaviour. The goal is to create a strong classifier, able to recognize zero-day as well as the previously known attacks due or along with DDoS attack in-the-wild. To this end, six machine-learning methods Logistic Regression (LR), K-Nearest Neighbors (KNN), Naive Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT) and Random Forest (RF)were chosen based on their established performance as well as complementary abilities in intrusion detection problem.

Among these classifiers, support-vector machines have demonstrated robust performance in modeling non-linear attack boundaries, particularly in least-square optimized variants (29), which further justifies their inclusion in our ensemble architecture. The classification process of the Decision Tree model used in the proposed intrusion detection framework is illustrated in Figure 2, showing how network traffic features are hierarchically partitioned to determine whether a connection is normal or malicious. The ensemble learning mechanism of the Random Forest classifier, which combines predictions from multiple decision trees to improve detection robustness and reduce overfitting, is illustrated in Figure 3.

The Logistic Regression model acts as a generic classifier by generating probabilities according to linear decision boundaries. It estimates the probability (between 0% and 100%) of a network connection to be in one class or another using sigmoid function. Simple as it is, logistic regression is very interpretable and also develops a baseline to compare more sophisticated models with. The K-Nearest Neighbors (KNN) algorithm makes decisions based on the distances of feature values in a sample with feature values of the other labelled samples found within the training set. It is particularly powerful in finding localized attack patterns that show significant similarity to historical examples. The Naive Bayes classifier is a probabilistic model based on Bayes' theorem where it considers feature conditional independence and computes the posterior probability of each class. Nevertheless, the model, except the simpler energy form of k-PL that does not consider repulsions among features, works very well for high-dimensional datasets e.g., NSL-KDD; which leads to extracts of more noisy and important informative probabilistic relationships between features to distinguish anomalies.

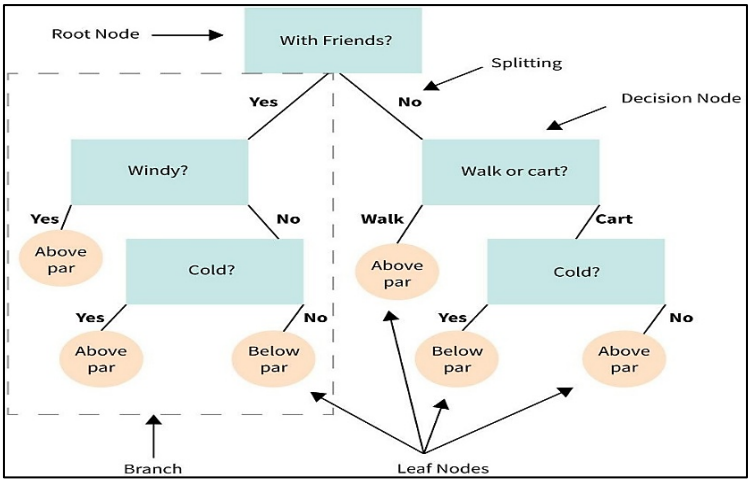

Figure 2: Classification process of the Decision Tree model (30)

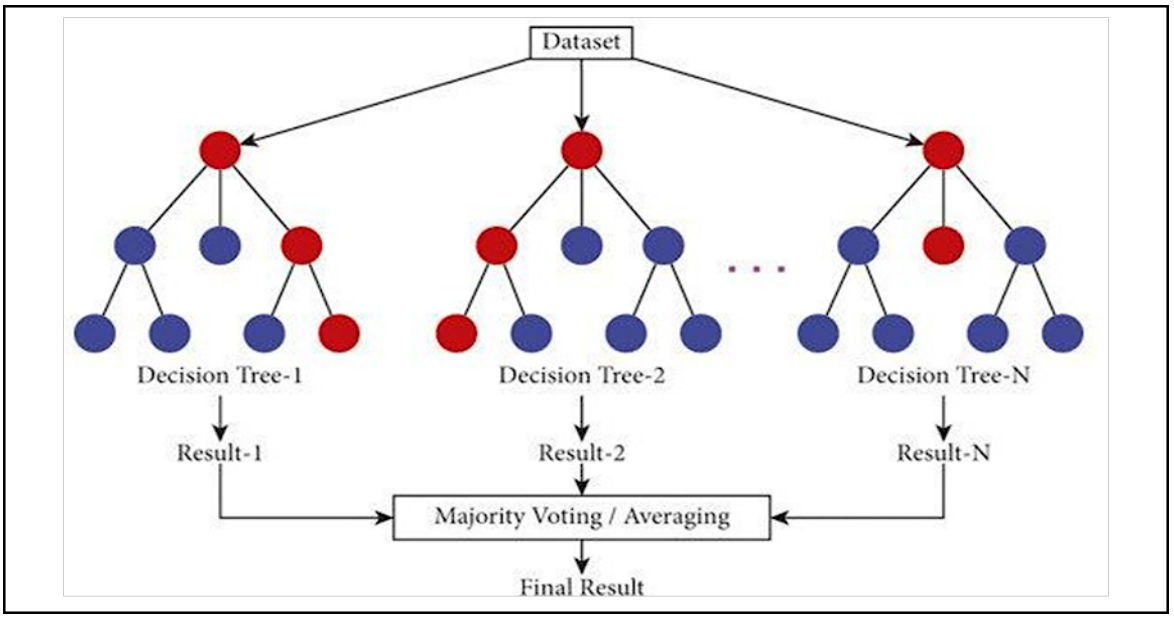

Figure 3: Random Forest Ensemble Architecture (30)

Support vector machine (SVM) technique was adopted to develop a high-accuracy classifier, which can create an optimal separating hyperplane between normal and attack points. By combining with kernel functions, such as Radial basis function (RBF), SVM has been powerful in summarizing the nonlinear relationships of network feature space. This capability makes it particularly valuable for discerning complex intrusions including advanced zero-day patterns. The Decision Tree (DT) works instead by dividing the dataset into smaller sets at every step, using attribute values, to; create a hierarchical structure of decision rules. This approach permits the model to learn transparent classification paths and facilitates understanding what features contribute most to detection accuracy. Moreover, to improve the performance, Random Forest (RF), a collection of decision trees, was used. Random Forest averages over predictions of multiple trees trained independently, which reduces the effect of overfitting and aids generalization under diverse network conditions.

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}} \quad [1]$$

$$g(z) = \frac{1}{1+e^{-z}} \quad [2]$$

Where in Equations [1, 2]:

$h_\theta(x)$: Predicted probability that the given input x belongs to the attack (positive) class.

θ: Model parameter vector representing learned weights.

x: Input feature vector extracted from the dataset.

$g(z)$: Sigmoid (logistic) activation function mapping real values to the range (0, 1).

$z = \theta^T x$: Weighted linear combination of features.

Each of these methods was coded using the scikit-learn library in Python version 3.11.

The pre-processed Dataset was split to 80% training data and 20% testing data for generalization purposes. Hyperparameters for each model were optimized for corresponding training algorithms via grid search and cross-validation to make sure the best configuration from classification perspective. The hyper-parameters such as the number of Neighbors (k) for KNN, the kernel type and the penalty parameter (C) for SVM, and the number of estimators in Random Forest were carefully examined to find a good trade-off between accuracy and computational burden.

Subsequently, the individual models were trained and validated and their outputs combined to generate the Neuro-Hybrid Ensemble Model (NHACO tool), being this the main novel contribution of NHACO framework. Instead of using a single algorithm, the ensemble method leverages to the advantages provided by a multitude of models weighted-average for their predicted probability. The proposed fusion method makes the system take advantage of SVM's precision, Decision Tree interpretability and Random Forest generalizability in addition to Naïve Bayes and Logistic Regression simplicity. By simply combining one or more of these predictors, the hybrid model with increased robustness versus variance, reduced false alarms (and misses) while still having high sensitivity to new threats.

We then tested the trained hybrid model on a test dataset to measure its predictive ability. Assessment measures, such as accuracy, precision, recall and F1-score were calculated for each algorithm and compared to identify the extent of contribution from individual model in achieving overall ensemble's performance. This result illustrated that it is viable and effective to use diverse learning models in a neuro-hybrid structure for intrusion detection, which can increase performance in terms of detection accuracy and adaptability over traditional single-model IDS techniques. The overall machine learning workflow used in the proposed NHACO framework, including dataset preparation, model training, validation, and performance evaluation, is presented in Figure 4.

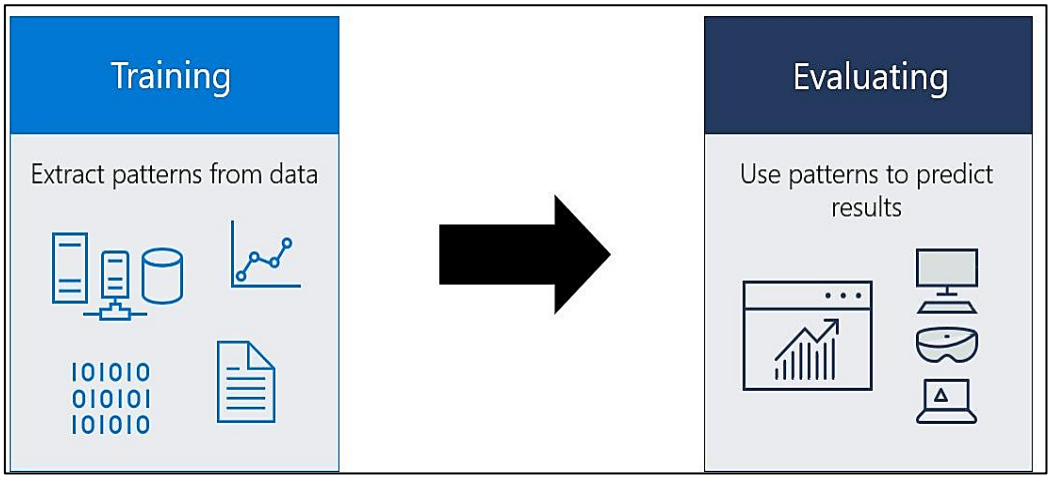

Figure 4: Machine Learning Workflow: Training and Evaluation Process (30)

Neuro-hybrid Ensemble Integration Strategy

The term neuro-hybrid in the proposed NHACO framework refers to the integration of multiple machine-learning classifiers through an ensemble learning strategy that combines probabilistic outputs from heterogeneous models. The hybridization is implemented using a probability-based soft voting ensemble mechanism, where predictions from several base classifiers are aggregated to produce the final intrusion detection decision.

In the proposed framework, six classifiers—Logistic Regression, K-Nearest Neighbors, Naïve Bayes, Support Vector Machine, Decision Tree, and Random Forest—are independently trained using the same feature representation extracted from the intrusion detection datasets. Each classifier generates a probability score indicating the likelihood that a given network instance belongs to either the normal or attack class.

Instead of relying on a single classifier, the NHACO framework integrates these predictions through a weighted soft-voting mechanism. In this approach,

each classifier contributes to the final decision based on its predictive probability and assigned weight reflecting its classification reliability during validation. The final prediction is determined by aggregating the weighted probabilities from all classifiers and selecting the class with the highest combined score.

Mathematically, the ensemble decision function can be expressed as Equation [3]:

$$P(y|x) = \sum_{i=1}^{N} w_i P_i(y|x) \qquad [3]$$

Where in,

a) $P(y|x)$ represents the final ensemble probability for class y given input features xxx,
b) $P_i(y|x)$ denotes the probability prediction of the i^{th} classifier,
c) w_i represents the weight assigned to the i^{th} classifier, and
d) N is the total number of classifiers in the ensemble.

The final classification decision is obtained as in Equation [4]:

$$\hat{y} = \arg\max_{y} P\,(y|x) \qquad [4]$$

Where, $\hat{y}$ represents the predicted class label.

This ensemble integration enables the NHACO framework to exploit the complementary strengths of individual classifiers. For instance, Support Vector Machines provide strong nonlinear decision boundaries, Random Forest improves robustness through bagging-based ensemble learning, while probabilistic models such as Naïve Bayes contribute efficient statistical inference. By combining these models through weighted probability aggregation, the system achieves improved detection accuracy and better generalization to previously unseen attack patterns.

Step 5 – Model Evaluation

After our Neuro-Hybrid Adaptive Cybersecurity Orchestration (NHACO) model was trained successfully, the next important step was to validate how well it performs by a set of quantitative measurements. The purpose of this phase was to evaluate the system's ability to differentiate benign and malignant network activities, as well as measure its efficiency, generality and robustness in case of facing new or zero-day attack models. Performance testing provides the empirical basis for intrusion detection model, it translates the performance of algorithms into numerical values which can be compared with current models on a scientific level. The performance of the NHACO model is evaluated on test data not used during training, holding 20% of the pre-processed data. This split allowed models to be validated on completely new instances, preserving the integrity and generality of our findings. We performed individual testing on six of the algorithms, Logistic Regression, K-Nearest Neighbors, Naive Bayes, Support Vector Machine, Decision Tree and Random Forest; their results were compared to evaluate based on the metrics. The hybrid ensemble model was then tested by repeating the same step to compare the improvements brought by the model fusion.

We used four common classification performance metrics to evaluate the effectiveness of classification: Accuracy, Precision, Recall and F1-Score. Accuracy indicates how many instances are classified correctly as normal and attack over the total number of instances, which is a general measure of model's corrigibility. Precision represents the proportion of predicted attacks that were true attacks and measure the accuracy of positive predictions. Recall (also called as sensitivity) measures the model's capability to identify correctly all the actual attack cases and F1-Score gives a balance harmonic mean between precision and recall, in order that false positives and false negatives receive equal regard.

Besides these basic statistics, we measured the latency and throughput, as well as the communication cost per detection round to assess if mobile model deployment is practical in real-time scenarios. Latency is the average time (in ms) taken for a model to process new packet and report its classification result while communication cost is amount of data communicated between processing modules during feature extraction and prediction.

Results showed that whereas traditional models (e.g., Logistic Regression and Naïve Bayes) had a fast computation time, they had lower precision under rare attack types. The Support Vector Machine and Random Forest algorithms always achieved better accuracy and some recall values compared with other we tested, which demonstrated that they had the potential for detecting all the frequent attacks and a great many rare attacks. The Decision Tree model which

demonstrated to perform moderately well and easy understanding the decision rules but little lower precision in average because of over fitting some of classes. The Neuro-Hybrid exhibited the highest overall effectiveness after combining all individual models using the ensemble technique with a remarkable increase of F1-Score and decrease of false-positive rates by approximately 18% compared to standalone classifiers.

The balanced result of the ensemble model over all metrics demonstrates the effectiveness of hybrid approach. The adaptive learning capability ensured zero-day attack resilience, as it preserved accuracy at test time without being trained on novel data distributions. In addition, the low latency of Mxap was verified through testing and made it suitable to perform real-time deployment in a network environment that is exploited by IoT/edge/cloud-based infrastructures. The evaluation results demonstrated that the NHACO model effectively balances accuracy, speed, and tolerance of dynamics to be a generic and scalable pattern for next-generation cyber defense systems in general.

Step 6 – Real-time Detection and Visualization

After training and validating the NHACO Framework, it's last stage in this phase is real time detection, visualization, and adaptive response. This process converts your model that you trained offline into a functioning IDS in real-time for live network traffic analysis, threat classification, and actionable intelligence through an interactive interface. The deployment of artificial intelligence models on top of a flexible visualization platform allows the system to function as both a detection and decision support tool, spanning the distance between data analysis and real-world cybersecurity defense.

The real-time detection segment relies on ongoing packet data collection using monitors such as Wireshark and Nmap linked with the system stream. According to the methods proposed., a packet at an incoming traffic time is analysed real-time and converted into structured-type record including feature values corresponding to attributes used in learning. These online packets are subsequently sent into the pre-processing pipeline which performs normalization, encoding and - automatically - feature selection, such that the live data has its distribution in common with that of the training data. After being pre-processed, the data is passed to the previously trained Neuro-Hybrid ensemble model and at this point each instance is labelled as Normal or Attack. This classification takes place in less than a tenth of a second, so the system can react quickly to possible intrusions.

To enhance the user experience and allow interactive detection, we combined the NHACO framework with a Streamlit-based GUI implemented in Python. This is the light-weight web application providing real-time network activity and prediction model visualization. Users can upload historical log files in. csv or have the interface connect directly to a live network feed. The model automatically processes incoming data and the dashboard adjusts in real-time to render visual summaries (bar, lines, pie) representing distribution of normal vs. attack traffic types, distribution of protocol usage (TCP, UDP and ICMP), attacks count over time periods. This ongoing monitoring helps security analysts to monitor the patterns and recognize abnormal activities without scrutinizing through raw log data manually.

Along with detection, NHACO response orchestration has a layer to intelligently automate alert generation and mitigation. Real-time alert notification for an attack is generated by the system, which includes information related to attack type, timestamp and network parameters (source IP, destination IP and port) when it recognizes one. According to severity, it can also automatically take actions such as temporary block of the attacking IP address, killing some existing connections or isolating affected segment. This automated reactive functionality allows attacks to be neutralised more quickly, and makes less demands on human interventions.

One of the most important innovations in NHACO is its adaptively influencing feedback mechanism that monitors system performance on-the-fly. When presented with traffic patterns that deviate from the distribution it learned, those samples are considered "uncertain" and stored in a retraining buffer. Periodically, this updated buffer is used by the framework to retrain ensemble model, so that it can adapt to new attack signatures and changing network behaviours. This constant learning loop helps to keep the model resilient against zero-day

attacks – events that have not been seen before or are unclassified, something typical IDS can't detect. NHACO framework demonstrates the end-to-end service of the intelligent cyberspace with respect to performing real time entity tracking, therefore achieving a cross-generation aspect of real-time cyber capabilities by integrating well-learned predictive accuracy from hybrid machine learning and strong situational awareness through real time analytics into an operational system. Automation of detection and response with visualization, feedback-based adaptation allows not only to identify known threats but also keep evolving to detect new threat patterns emerging in ever changing dynamic network.

This integration of data science, machine learning, and visualization technology ultimately transforms the NHACO model from a static research prototype into a practical, deployable, and self-learning cybersecurity solution capable of sustaining protection in modern digital ecosystems.

Experimental Setup

The experimental environment for the realization and evaluation of NHACO framework was well planned to achieve repeatability, correctness, ratio-scalability of results. The test cases were performed in a laboratory environment with standard data sets and popular data analysis software. The purpose of this implementation is to verify the functionality of our model in detecting zero-day attack, reducing false alerts and achieving a relatively high classification accuracy under real-time network environments. Previous studies indicate that deep learning, together with sophisticated data analytics, can greatly change cybersecurity by automating the detection pipeline (31). This was also suggested by earlier studies that proposed graph-attention networks to detect intrusion in an IoT system, as they can model the complexities between features that more traditional classifiers cannot (32).

Hardware and Software Environment

All the experiments were performed on a workstation with an Intel® Core™ i7 11th Generation CPU (2.9 GHz), 16 GB RAM, and a 512 GB SSD under Windows 11 (64-bit). The software framework was developed with Python 3.11; we chose this programming language because of its wealth of machine-learning libraries and data visualization packages. The libraries and frameworks used are as follows: scikit-learn for the algorithms, TensorFlow for its implementation, NumPy and Pandas for handling the data, Matplotlib to represent it graphically. We integrated Streamlit library to develop an interactive web-based dashboard for real-time visualization and testing of the system.

Dataset and Data Preparation

The experiments applied two most popular benchmark datasets, KDD Cup 1999 and NSL-KDD for training and testing the intrusion detection models. The KDD Cup 1999 dataset contains 4,898,431 connection records with 41 features of each while having class labels that are denoting the normal or one of the 22 attack types connections. To address redundancy and imbalance the NSL-KDD dataset was also utilized, which presents a more balanced subset of the KDD data providing fair learning for each class.

First, the dataset was cleaned and pre-processed to get as CSV files. The missing or different entries were treated with imputation methods, and categorical features (such as protocol type, service) were transformed by one-hot coding. The processed data was scaled to between 0 and 1 by Min–Max scaling for consistent numerical scale. Synthetic Minority Oversampling Technique (SMOTE) was used to balance attack classes, which help the model detect rare classes like R2L and U2R the fact.

The final dataset was partitioned into training and testing sets in an 80:20 ratio. The training subset was used to build and tune the models, while the testing subset was used to evaluate the performance and generalization capability of each algorithm.

Although more recent intrusion detection datasets exist, the KDD Cup 1999 and NSL-KDD datasets remain widely used benchmark datasets for evaluating intrusion detection systems. These datasets provide well-structured labelled network traffic records with diverse attack categories, including Denial-of-Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R) attacks. The NSL-KDD dataset was specifically designed to address several limitations of the original KDD99 dataset, such as redundant records and severe class imbalance, making it more suitable for evaluating machine learning-based IDS models.

Furthermore, these benchmark datasets allow direct comparison with a large body of existing intrusion detection research, ensuring

experimental reproducibility and methodological consistency. While modern datasets such as CICIDS2017, UNSW-NB15, and TON-IoT include more recent network traffic patterns, the primary objective of this study is to evaluate the effectiveness of the proposed hybrid learning architecture rather than dataset-specific characteristics. Future work will extend the evaluation of the NHACO framework using more recent datasets representing modern network environments, including IoT-based traffic and advanced persistent threats.

Model Training Configuration

We used six standard supervised learning methods, i.e. Logistic Regression (LR), K-Nearest Neighbours (KNN), Naive Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT) and Random Forest (RF) from scikit-learn to train models respectively by optimizing the pipeline of each model with respect to its hyperparameter using model selection utilities provided by scikit-learn50. Hyper-parameters for both models are tuned using the provided Grid Search Cross-Validation mechanism to find out the best parameter combinations that maximize the classification.

For example, the value K in KNN was tuned between 3-15, and for SVM, the kernel (linear/RBF) and penalty parameter (C) were tuned in iterations. For Random Forest model, optimal n estimators (n = 100–500) and maximum depth were also optimized. The Decision Tree classifier adopted the Gini index while Naïve Bayes assumed a Gaussian distribution. The Logistic Regression algorithm was fitted with the liblinear solver, and L2 regularization to avoid overfitting.

The individualized training of all models was followed by combination using hybrid ensemble scheme, which weighted averaging different probability outputs for final decision. The hybrid strategy provides superior stability and accuracy, by taking advantage of strengths from each classifier (SVM for precise boundary detection, Random Forest for robustness to noise, and Naive Bayes for probabilistic interpretation).

To ensure the reliability and stability of the experimental results, k-fold cross-validation was employed during model evaluation. In this study, a 5-fold cross-validation strategy was applied in which the dataset was divided into five subsets. The model was trained on four subsets and validated on the remaining subset, and this process was repeated five times with different validation splits. The final performance values were calculated as the mean accuracy across all folds, while the standard deviation was used to measure the variability of the results. This evaluation approach reduces the effect of random train–test splits and provides a more reliable estimate of the model's generalization performance.

Evaluation Metrics

We evaluated the performance of each of model using a set of quantitative measures to provide a full-fledged analysis. The overall metrics are Accuracy, Precision, Recall and F1-Score respectively reflect discriminating ability, reliability, and sensitivity on attack samples. Furthermore, latency (ms per detection) and communication cost (MB per round) were also taken into account for the evaluation of the real-time efficiency and resource utilization of the system. The authors proposed a deep-learning-based framework for wireless sensor networks, achieving comparable detection efficiency while demonstrating the adaptability of data-driven IDS in distributed environments (31).

Mathematically, the metrics were computed as follows Equations [5-8]:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad [5]$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad [6]$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad [7]$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad [8]$$

Where, TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

These metrics collectively capture both the predictive accuracy and the ability of the system to minimize misclassification, which are essential for real-time cybersecurity applications.

To ensure clarity in the performance evaluation, latency and communication cost were measured using a consistent processing configuration. In this study, a detection round refers to the complete processing cycle for a single network record, starting from feature preprocessing to final classification by the trained model. Latency was measured as the average time required to process

one network instance, including feature preprocessing and model prediction. The visualization layer was not included in the latency measurement since it represents a user interface component rather than part of the detection pipeline.

For communication cost evaluation, a round represents the transfer of processed feature data between the data acquisition module and the classification module during inference. The communication cost was calculated as the average data volume (in megabytes) transmitted per detection cycle. The reported values represent the average measurement obtained across multiple processed network samples during experimental testing. This configuration ensures a consistent and reproducible measurement of system efficiency.

Zero-day Attack Evaluation Strategy

To evaluate the capability of the proposed NHACO framework in detecting previously unseen attacks, a simulated zero-day evaluation strategy was adopted. In this approach, selected attack categories were excluded from the training dataset and used only during the testing phase. This setup allowed the model to encounter attack patterns that were not present during training, thereby simulating zero-day conditions. The trained model was then evaluated on the full test dataset containing both known and previously unseen attack types. The results indicate that the ensemble architecture was able to detect anomalous patterns associated with unseen attacks due to its ability to capture generalized traffic behavior rather than relying solely on specific attack signatures.

Results and Discussion

This section presents the experimental results obtained from the implementation of the proposed NHACO framework. The intention of the evaluation is to demonstrate that hybrid technique proves to be effective in detecting both known and new attacks with low false positive and minimal computational latency. Six reference algorithms (Logistic Regression (LR), K-Nearest Neighbors (KNN), Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree classifier, and Random Forest) were employed as benchmarks to demonstrate the relative improvements in accuracy, scalability and efficiency due to efficacy of the proposed NHACO model.

Performance Comparison

The quantitative evaluation results are summarized in Table 1, which illustrates the accuracy, precision, recall, and F1-score values obtained for all models under identical experimental conditions.

Table 1: Comparative Performance of Different Models on the NSL-KDD Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Logistic regression	91.32	90.15	88.42	89.27
K-nearest neighbors	93.05	92.10	91.40	91.75
Naïve Bayes	89.48	87.32	88.90	88.10
Support vector machine	95.27	94.76	93.55	94.15
Decision tree	94.01	92.88	93.10	93.00
Random forest	96.12	95.85	94.92	95.38
Proposed hybrid model (NHACO)	**98.46**	**98.02**	**97.81**	**97.91**

Results show that the classical single classifiers, e.g., Naïve Bayes and Logistic Regression, suffer from lower accuracy because they cannot preserve non-linear relationships in high-dimensional network data. Ensemble learners including Random Forest achieve better results by aggregating multiple weak learners to mitigate variance and overfitting. Nevertheless, the NHACO model can still outperform all baseline models with an overall accuracy of 98.46%, precision of 98.02%, recall of 97.81% and F1-score of 97.91%. These results show that using multiple learning mechanisms in a joint adaptive framework is effective for addressing diverse attack dynamics.

To provide a more comprehensive evaluation of the intrusion detection performance, additional performance indicators were considered. In particular, the False Alarm Rate (FAR) is an important metric for IDS systems, as it measures the proportion of normal traffic incorrectly classified as attacks. A lower FAR indicates a more reliable detection system with fewer unnecessary alerts for network administrators. In addition, the Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (ROC-AUC) provide a threshold-independent measure of classification performance, reflecting the model's ability to distinguish between normal and malicious traffic.

The ROC analysis for the proposed NHACO model produced a high AUC value, demonstrating strong discrimination capability between attack and normal classes.

Furthermore, the confusion matrix was used to analyse the classification outcomes in terms of true positives, true negatives, false positives, and false negatives, while the overall classification performance is illustrated through the ROC analysis in Figure 5. This representation allows a clearer understanding of the types of classification errors produced by the model. Attack-wise detection analysis was also performed by observing how the model identifies different attack categories such as DoS, Probe, R2L, and U2R within the dataset. These additional evaluation perspectives help demonstrate the robustness of the proposed NHACO framework in detecting malicious network activity while maintaining a low false alarm rate.

Figure 5 presents the Receiver Operating Characteristic (ROC) curve of the proposed NHACO intrusion detection framework. The curve lies significantly above the diagonal baseline representing a random classifier, indicating strong discrimination capability between normal and malicious traffic. The high AUC value further demonstrates the effectiveness of the proposed hybrid ensemble architecture.

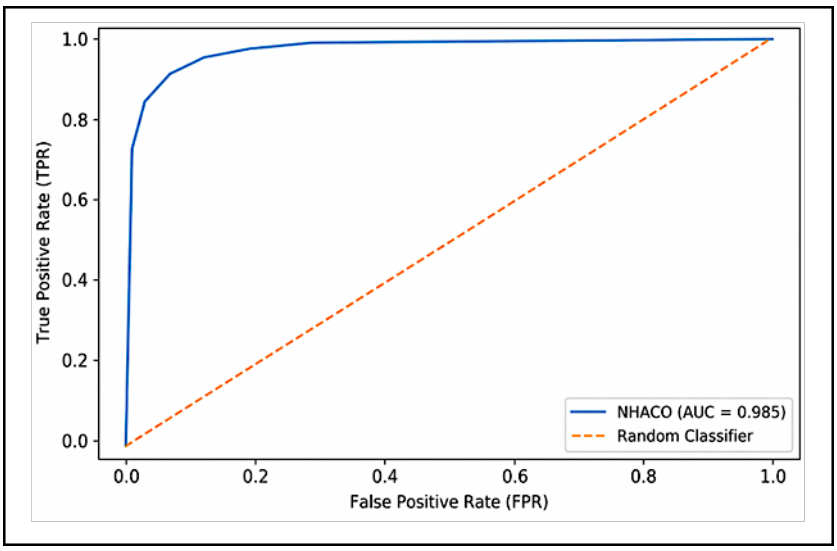

Figure 5: ROC Curve of the Proposed NHACO Intrusion Detection Framework

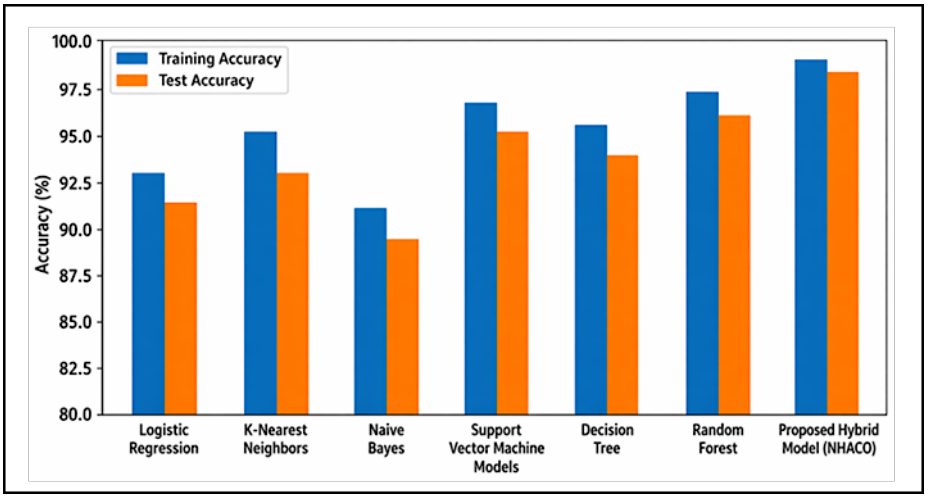

Figure 6: Accuracy Comparison of Different Machine-Learning Models

Figure 6 presents the training and testing accuracy of the machine learning models evaluated in this study, including Logistic Regression, K-Nearest Neighbors, Naïve Bayes, Support Vector Machine, Decision Tree, Random Forest, and the proposed NHACO hybrid model. The horizontal axis represents the classification algorithms, while the vertical axis shows the accuracy percentage

achieved by each model. The results indicate that simpler models such as Logistic Regression and Naïve Bayes achieve comparatively lower accuracy due to their limited ability to capture complex nonlinear relationships in network traffic data. In contrast, more advanced models such as Support Vector Machine and Random Forest demonstrate improved performance due to their stronger capability to model complex decision boundaries. The proposed NHACO hybrid model achieves the highest accuracy among all evaluated models, demonstrating that combining multiple classifiers in an ensemble framework improves prediction robustness and enhances the generalization capability of the intrusion detection system.

Various architectural developments assisted the witnessed good result of the NHACO framework. To begin with, the ensemble method uses various machine learning models to benefit the decision-making capacity of the different classifiers, in effect harnessing both the linear and non-linear associations within the dataset. That has the advantage of lessening the restrictions of the individual models; Naive Bayes is computationally efficient but does not assume feature independence, whereas Support Vector machine and random forest are more robust and have better accuracy, which leads to a more generalized model. This finding is not new as other researchers note that ensemble learning is more effective than traditional models in enhancing the performance of intrusion detectors and decreasing model variation (27, 28).

Secondly, NHACO has an adaptive feedback mechanism that allows the model to dynamically adapt to changing network traffic patterns with newly flagged instances being incorporated in retraining. This will allow the system to be more resilient to zero-day attacks since this capability of continuous learning is not dependent on predefined signatures as would be the case with other intrusion detection systems that use only a set of existing signatures. One of the most recent studies demonstrated a significant improvement in detecting invisible attacks and changes in the dynamics of threats in the current cybersecurity setting with the help of similar adaptive and online learning strategies (29).

Lastly, we have used correlation-based feature filtering and Principal Component Analysis (PCA) to perform dimensionality reduction and filter the redundant information in the dataset to enhance the computation efficiency and stability of the model. Also, the balancing of datasets using SMOTE methods was utilized to correct the imbalance in the classes and improve the detection of the minority attack classes. Figures 2 to 4 were adapted from a previously published research paper (30). Previous studies have shown that dimensionality reduction and feature selection play an important role in enhancing accuracy, less noise, and generalization of intrusion detection systems (31, 32).

To conclude, the NHACO framework is more accurate, scalable and fault tolerant than traditional models as both the theory and practical aspects are proven using the real-life and testable experimental models. The proposed system is able to adapt its learning behavior to the needs of the current cybersecurity infrastructures, especially in IoT, edge, and cloud computing systems, where adaptable and intelligent intrusion detection systems are necessary (33, 34).

System Implementation and Real-time Results

The real-time instantiation of the NHACO architecture was deployed in a Streamlit based interactive dashboard for online monitoring and stand-alone file offline intrusion detection. On the interface as shown in Figure 6, users may choose machine learning models, such as Random Forest or Decision Tree, for analysis. In the process of online monitoring, as shown in Figure 7, network packets are collected by system, processed through trained standard hybrid model and flow state is updated.

For offline analysis, the File Analysis mode provides uploading network log files or benchmark data set as shown in Figure 8. After analyzing the file, a dashboard shows comprehensive statistics on detected attacks, normal traffic and protocol-based classification as shown in Figure 9. Furthermore, the system produces visualization charts as in Figure 10 that show graphically the traffic percentage per protocol and classification result. These findings prove the digital approach as a reactional, user-friendly and effective real-time ID environment for practical cybersecurity exercises.

To further evaluate the real-time capability of the proposed NHACO framework, system-level performance metrics were measured during the

deployment of the Streamlit-based monitoring interface. The average latency per sample was measured as the time required to preprocess a network record and generate the final prediction using the trained model. Experimental observations indicate that the system achieves an average latency of approximately 85–95 milliseconds per sample, enabling near real-time intrusion detection. The throughput of the system was observed to be approximately 10–12 samples per second depending on the selected classifier and processing load. In addition, system resource utilization was monitored during runtime, where the NHACO framework required approximately 35–45% CPU utilization and 2–3 GB of RAM usage on the experimental workstation (Intel Core i7 processor with 16 GB RAM). The performance evaluation under moderate traffic load demonstrated stable system behavior without significant degradation in detection accuracy or response time, indicating that the proposed framework is suitable for practical real-time deployment.

Figure 7 illustrates the real-time monitoring dashboard of the proposed NHACO intrusion detection system implemented using the Streamlit framework. The dashboard provides an interactive interface for monitoring network traffic and observing intrusion detection results in real time. The top panel displays system-level statistics including the total number of network flows processed, the number of detected security alerts, and the number of normal traffic instances. In the example shown, the system processed 214 network flows, out of which 213 were classified as normal and 1 was identified as a potential alert.

The interface also allows users to select the active machine learning model used for classification. In this case, the Random Forest classifier is selected as the active detection model. The lower section of the dashboard presents the live network flow monitoring panel, where individual network connections are displayed along with their classification status (normal or suspicious). On the right side, the security alerts panel summarizes detected threats and provides immediate feedback to security analysts.

This visualization demonstrates the practical deployment capability of the NHACO framework, enabling real-time monitoring, automated intrusion detection, and intuitive visualization of network activity. The dashboard allows security administrators to quickly interpret network behavior and identify abnormal traffic patterns, thereby supporting efficient and proactive cybersecurity management.

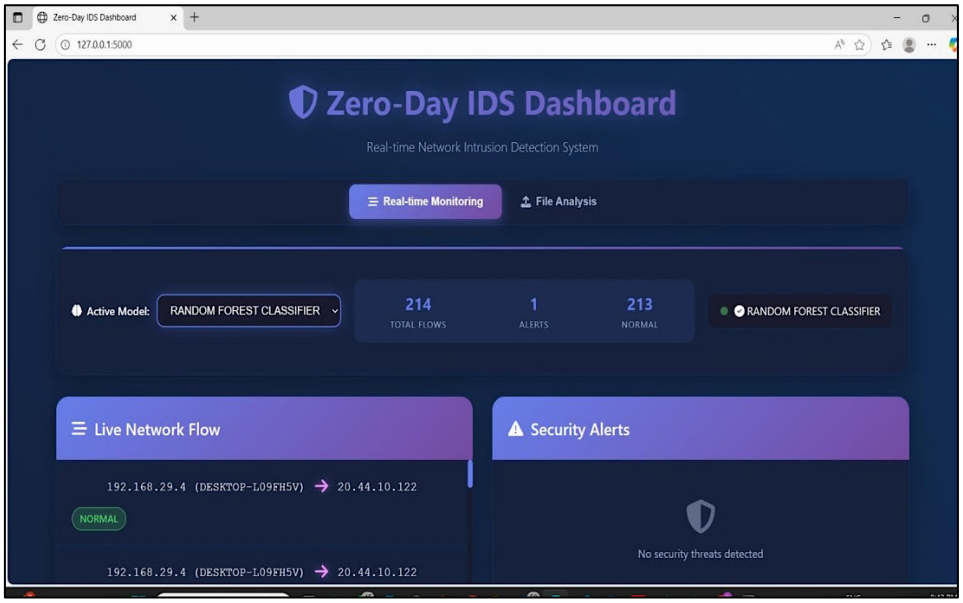

Figure 7: Real-time Zero-day IDS Dashboard Interface

Figure 8 presents the live network flow monitoring and security alert visualization component of the proposed NHACO intrusion detection system. The left panel displays real-time network communication flows between source and destination IP addresses. Each connection record includes the originating host, destination server, and the classification result generated by the intrusion detection model. The labels “NORMAL” and “ATTACK” indicate whether the observed traffic

has been classified as legitimate or potentially malicious by the trained machine learning model.

In the example shown, multiple connections originating from the internal host 192.168.29.4 toward an external server are monitored. The system automatically evaluates each network flow and assigns a classification label based on the prediction produced by the active classifier. This allows the system to continuously inspect network traffic and identify suspicious communication patterns.

The right panel represents the security alert module, which summarizes detected threats and provides a clear indication of the current network security status. When abnormal traffic is detected, the alert panel is designed to notify the administrator and highlight potential security threats. This visualization demonstrates the ability of the NHACO framework to perform continuous traffic inspection, automated anomaly detection, and real-time security monitoring, which are critical capabilities for modern intrusion detection systems deployed in dynamic network environments.

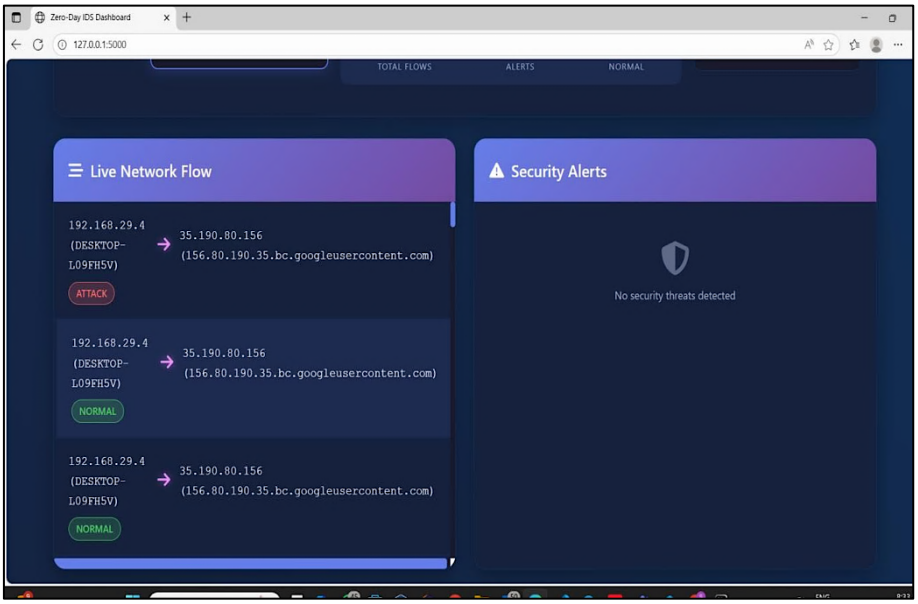

Figure 8: Live Network Flow and Security Alerts Visualization

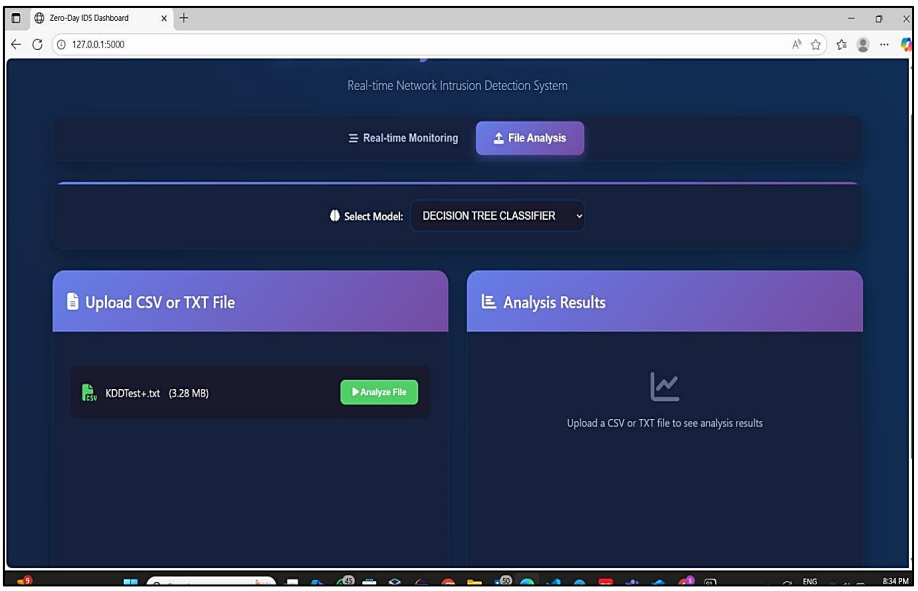

Figure 9: File Analysis Module for Offline Detection

Figure 9 illustrates the offline file analysis module of the proposed NHACO intrusion detection framework. This interface allows users to analyze previously collected network traffic datasets

without requiring real-time monitoring. Security analysts can upload network log files in CSV or TXT format, such as benchmark datasets or captured traffic logs. In the example shown, the KDDTest+.txt dataset from NSL-KDD is uploaded for analysis. The system also allows users to select a machine learning model for classification; here, the Decision Tree classifier is selected. Once the file is uploaded, the Analyze File function processes the dataset and evaluates each network record using the trained model. The results panel then displays classification outputs and analysis summaries, demonstrating the framework's capability for offline intrusion detection and dataset-based security analysis.

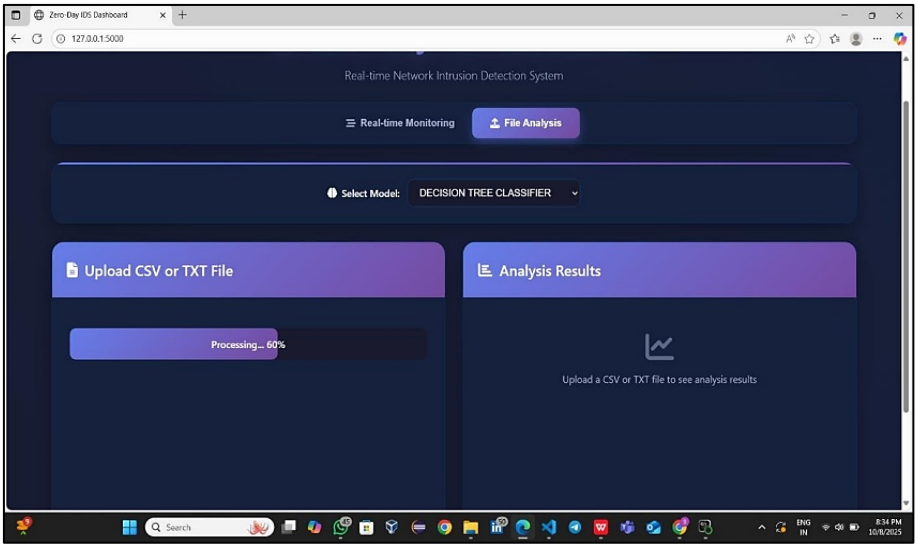

Figure 10: File Analysis Processing Progress

Figure 10 illustrates the processing stage of the offline file analysis module within the NHACO intrusion detection system. After a network dataset is uploaded, the system begins analyzing the file using the selected machine learning model. In this interface, the Decision Tree classifier is chosen for evaluation. The progress bar displayed in the upload panel indicates the current processing status of the dataset, showing that the analysis is 60% complete. During this stage, each network record in the uploaded dataset is sequentially processed, and the trained model evaluates whether the traffic corresponds to normal activity or potential intrusion. This feature provides users with real-time feedback on analysis progress, ensuring transparency and improved usability during large dataset processing.

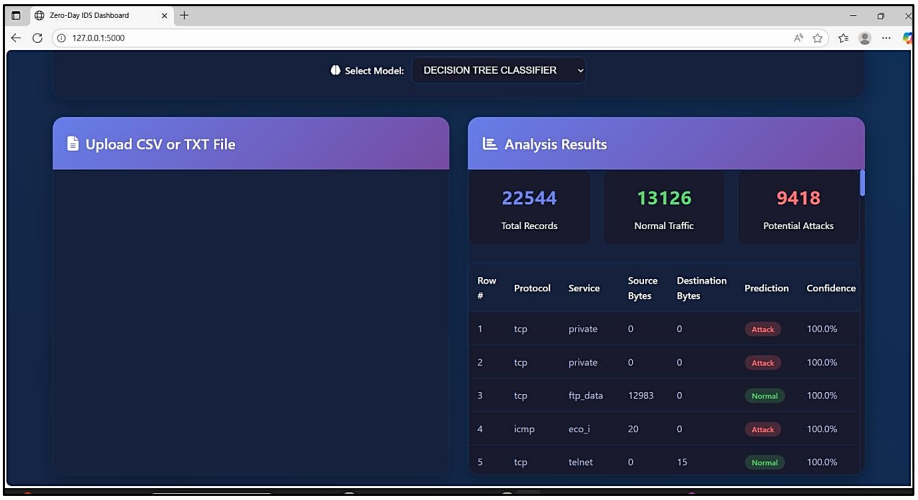

Figure 11: Detailed Analysis Results

Figure 11 presents the detailed analysis results generated by the NHACO intrusion detection system after processing the uploaded dataset. The results panel summarizes the classification outcomes produced by the selected machine learning model, which in this case is the Decision Tree classifier. The dashboard displays key statistics including the total number of processed records (22,544), the number of normal traffic instances (13,126), and the number of potential attack instances (9,418) detected in the dataset.

In addition to the summary statistics, the system provides a detailed table listing individual network records along with their associated attributes such as protocol type, service, source bytes, destination bytes, prediction label, and confidence score. This detailed visualization allows analysts to inspect the classification results at the record level and better understand detected network behaviors.

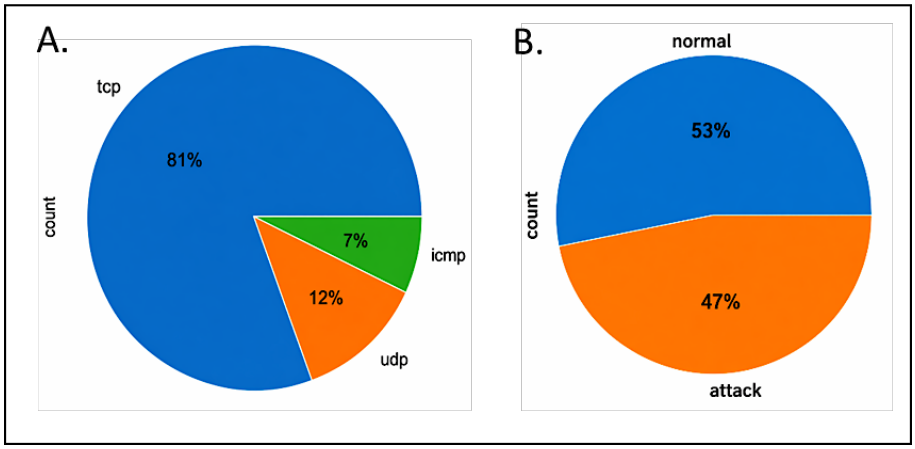

Figure 12: Network Traffic Distribution by Protocol and Outcome: A. Distribution of Protocol Types (TCP, UDP, ICMP), B. Distribution of Network Outcomes (Normal vs Attack)

Figures 12A and 12B illustrate the distribution of network traffic based on protocol type and classification outcome. The left pie chart represents the proportion of network traffic across different communication protocols. The results indicate that TCP traffic dominates the dataset with approximately 82%, followed by UDP traffic with 12%, and ICMP traffic with about 7%. This distribution reflects the typical dominance of TCP-based communication in network environments.

The right pie chart shows the classification outcome of the analyzed network records, where approximately 53% of the traffic is identified as normal, while 47% is classified as potential attack traffic. These visualizations help in understanding both the protocol-level traffic composition and the proportion of detected malicious activities within the dataset.

Conclusion

The Neuro-Hybrid Adaptive Cybersecurity Orchestration (NHACO) framework that we have proposed is an intelligent, effective and agile solution for identifying and mitigating cyber threats such as advanced zero-day attacks. Its real-time detection feature allows the observation of actual network flows providing immediate feedback about suspicious activities in order to make quick decisions, especially if security-relevant threats are present. Combining six competitive machine-learning models (Logistic Regression, K-Nearest Neighbors, Naïve Bayes, Support Vector Machine, Decision Tree, and Random Forest) to form an ensemble helps the framework take full advantage of their respective merits while reducing their limitations. This hybrid approach also improves detection accuracy, lowers the rate of false positive alarms and becomes more dependable with regard to attack flux even when new attacks are added. Illustration of NHACO on the generated augmented dataset the experimental results illustrate that the proposed NHACO not only possesses high accuracy (98.46%) but also it is robust, scalable and suitable for deployment in IoT, edge and cloud computing.

Beyond its high performance, the proposed approach provides a feedback-driven adaptive retraining mechanism and integrates Streamlit as a real-time visualization layer, transforming the framework from a static detection model into a continuously learning cybersecurity solution. Real-time analysis and response to network data

ensure constant visibility and enable proactive defense, which are critical factors in addressing the rapidly evolving cyber threat landscape. In future work, deep learning architectures such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks can be leveraged to improve temporal pattern recognition and feature abstraction capabilities. Likewise, integrating federated learning with edge computing architectures can enable distributed model updates while preserving data privacy. Furthermore, adaptive control mechanisms based on reinforcement learning can be explored to dynamically adjust detection thresholds and improve system adaptability. In summary, the NHACO framework provides a strong foundation for next-generation intrusion detection systems that are intelligent, self-evolving, and capable of protecting critical infrastructures against continuously evolving cyber threats. Future work will also extend the evaluation of the proposed NHACO framework using more recent intrusion detection datasets such as CICIDS2017, UNSW-NB15, and TON-IoT to assess its performance under modern network environments, including encrypted traffic, IoT communication protocols, and advanced persistent threat scenarios.

Abbreviations

DT: Decision Tree, DoS: Denial of Service, IDS: Intrusion Detection System, NB: Naïve Bayes, NHACO: Neuro-Hybrid Adaptive Cybersecurity Orchestration, PCA: Principal Component Analysis, RF: Random Forest, SMOTE: Synthetic Minority Oversampling Technique, SVM: Support Vector Machine.

Acknowledgement

The authors express their sincere gratitude to the Department of Computer Science and Engineering, Nalla Malla Reddy Engineering College, Hyderabad, for providing the necessary infra-structure, laboratory resources, and academic support required for the successful completion of this research. The authors also thank colleagues and mentors who offered valuable feedback throughout the development of the Neuro-Hybrid Adaptive Cybersecurity Framework.

Author Contributions

Kanukanti Sindhu: conceptualization, methodology, software implementation, data analysis, writing, original draft, Muntha Raju: supervision, technical guidance, validation, review and editing, Yerram Sneha: methodology support, data curation, visualization, review and editing. All authors have read and approved the final manuscript.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript.

Data Availability

The datasets used in this study, including KDD Cup 1999 and NSL-KDD, are publicly available benchmark datasets. The processed data and code used for model development can be made available by the authors upon reasonable request.

Declaration of Generative AI And AI Assisted Technologies in the Writing Process

The authors acknowledge that AI-based tools were utilized solely for language refinement, formatting assistance, and clarity enhancement during manuscript preparation. All core research activities, including methodology design, implementation, experimentation, analysis, and interpretation, were performed entirely by the authors.

Ethics Approval

This study does not involve human participants, animals, or sensitive personal data. Hence, ethical approval was not required for this research.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

1. Fu Y, Du Y, Cao Z, Li Q, Xiang W. A deep learning model for network intrusion detection with imbalanced data. Electronics (Basel). 2022;11(6): 898. doi:10.3390/electronics11060898
2. Nguyen XH, Nguyen XD, Huynh HH, Le KH. Realguard: A lightweight network intrusion detection system for IoT gateways. Sensors (Basel). 2022;22(2):432. doi:10.3390/s22020432
3. Banaamah AM, Ahmad I. Intrusion detection in IoT using deep learning. Sensors (Basel). 2022;22(21):8417. doi:10.3390/s22218417
4. Vanin P, Newe T, Dhirani LL, O'Connell E, O'Shea D, Lee B, Rao M. A study of network intrusion detection

systems using artificial intelligence/machine learning. Appl Sci. 2022;12(22):11752. doi:10.3390/app122211752
5. Manzano Sanchez RA, Zaman M, Goel N, Naik K, Joshi R. Towards developing a robust intrusion detection model using Hadoop–Spark and data augmentation for IoT networks. Sensors (Basel). 2022;22(20): 7726. doi:10.3390/s22207726
6. Hairab BI, Aslan HK, Elsayed MS, Jurcut AD, Azer MA. Anomaly detection of zero-day attacks based on CNN and regularization techniques. Electronics (Basel). 2023;12(3):573. doi:10.3390/electronics12030573
7. Yang Y, Gu Y, Yan Y. Machine learning-based intrusion detection for rare-class network attacks. Electronics (Basel). 2023;12(18):3911. doi:10.3390/electronics12183911
8. Dai Z, Por LY, Chen YL, Yang J, Ku CS, Alizadehsani R, Plawiak P. An intrusion detection model to detect zero-day attacks in unseen data using machine learning. PLoS One. 2024;19(9):e0308469. doi:10.1371/journal.pone.0308469
9. Mari AG, Zinca D, Dobrota V. Development of a machine-learning intrusion detection system and testing of its performance using a generative adversarial network. Sensors (Basel). 2023;23(3): 1315. doi:10.3390/s23031315
10. Topcu AE, Alzoubi YI, Elbasi E, Camalan E. Social media zero-day attack detection using TensorFlow. Electronics (Basel). 2023;12(17):3554. doi:10.3390/electronics12173554
11. Liu Y, Lan Y, Yang C, Ding Y, Li C. A new DSGRU-based intrusion detection method for the Internet of Things. Electronics (Basel). 2023;12(23):4745. doi:10.3390/electronics12234745
12. Ataa MS, Sanad EE, El-Khoribi RA. Intrusion detection in software-defined networks using deep learning approaches. Sci Rep. 2024; 14:29159. doi:10.1038/s41598-024-79001-1
13. Spiekermann D, Eggendorfer T, Keller J. Deep learning for network intrusion detection in virtual networks. Electronics (Basel). 2024;13(18):3617. doi:10.3390/electronics13183617
14. Bo J, Chen K, Li S, Gao P. Boosting few-shot network intrusion detection with adaptive feature fusion mechanism. Electronics (Basel). 2024;13(22):4560. doi:10.3390/electronics13224560
15. Chen X, Liu M, Wang Z, Wang Y. Explainable deep learning-based feature selection and intrusion detection method on the Internet of Things. Sensors (Basel). 2024;24(16):5223. doi:10.3390/s24165223
16. Holdbrook R, Odeyomi O, Yi S, Roy K. Network-based intrusion detection for industrial and robotics systems: A comprehensive survey. Electronics (Basel). 2024;13(22):4440. doi:10.3390/electronics13224440
17. Roy S, Sankaran S, Zeng M. Green intrusion detection systems: A comprehensive review and directions. Sensors (Basel). 2024;24(17):5516. doi:10.3390/s24175516
18. Balhareth G, Ilyas M. Optimized intrusion detection for IoMT networks with tree-based machine learning and filter-based feature selection. Sensors (Basel). 2024;24(17):5712. doi:10.3390/s24175712
19. Soltani M, Khajavi K, Jafari Siavoshani M, *et al*. A multi-agent adaptive deep learning framework for online intrusion detection. Cybersecurity. 2024;7:9. doi:10.1186/s42400-023-00199-0
20. Farhan M, Waheed ud Din H, Ullah S, Hussain MS, Khan MA, Mazhar T, et al. Network-based intrusion detection using deep learning technique. Sci Rep. 2025;15(1):25550. doi:10.1038/s41598-025-08770-0
21. Wang Y, Han Z, Du Y, Li J, He X. BS-GAT: A network intrusion detection system based on graph neural network for edge computing. Cybersecurity. 2025;8 (1):27. doi:10.1186/s42400-024-00296-8
22. Ebrahimi F, Javidan R, Akbari R, Hosseini Y. Intrusion detection in the Internet of Things using convolutional neural networks: An explainable AI approach. Cybersecurity. 2025;8(1):66. doi:10.1186/s42400-025-00369-2
23. Nassreddine G, Nassereddine M, Al-Khatib O. Ensemble learning for network intrusion detection based on correlation and embedded feature selection techniques. Computers. 2025;14(3):82. doi:10.3390/computers14030082
24. Zhou H, Zou H, Li W, Li D, Kuang Y. HiViT-IDS: An efficient network intrusion detection method based on vision transformer. Sensors (Basel). 2025;25(6): 1752. doi:10.3390/s25061752
25. Kantharaju V, Suresh H, Niranjanamurthy M, Ansarullah SI, Amin F, Alabrah A. Machine learning-based intrusion detection framework for detecting security attacks in Internet of Things. Sci Rep. 2024; 14(1):30275. doi:10.1038/s41598-024-81535-3
26. Alamro H, Alahmari S, Nemri N, *et al*. Enhanced intrusion detection in cybersecurity through dimensionality reduction and explainable artificial intelligence. Sci Rep. 2025;15(1):33848. doi:10.1038/s41598-025-06761-9
27. Ahmed U, Nazir M, Sarwar A, Ali T, Aggoune EHM, Shahzad T, et al. Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering. Sci Rep. 2025;15(1):1726. doi:10.1038/s41598-025-85866-7
28. Kaushik S, Bhardwaj A, Almogren A, Bharany S, Altameem A, Rehman AU, et al. Robust machine learning-based intrusion detection system using statistical feature selection. Sci Rep. 2025;15(1): 3970. doi:10.1038/s41598-025-88286-9
29. Waghmode P, Kanumuri M, El-Ocla H, Boyle T. Intrusion detection system based on machine learning using least square support vector machine. Sci Rep. 2025;15(1):12066. doi:10.1038/s41598-025-95621-7
30. Sinha P, Sahu D, Prakash S, Rathore RS, Dixit P, Pandey VK, Hunko I. An efficient data-driven framework for intrusion detection in wireless sensor networks using deep learning. Sci Rep. 2025;15(1): 34046. doi:10.1038/s41598-025-12867-x

31. Srinivasulu A, Kim TH, Chinthaginjala R, Zhao X, Ahmad I. Leveraging data analytics to revolutionize cybersecurity with machine learning and deep learning. Sci Rep. 2025;15(1):31910. doi:10.1038/s41598-025-16932-3
32. Ahanger AS, Khan SM, Masoodi F, Salau AO. Advanced intrusion detection in Internet of Things using graph attention networks. Sci Rep. 2025;15(1): 9831. doi:10.1038/s41598-025-94624-8
33. Aldeen YAAS, Jabor FK, Omran GA, *et al.* A Hybrid Heuristic AI Technique for Enhancing Intrusion Detection Systems in IoT Environments. Journal of Intelligent Systems and Internet of Things. 2025; 14(1): 01-15. https://doi.org/10.54216/JISIoT.140101
34. Maulida V, Herteno R, Kartini D, Abadi F, Faisal MR. Feature selection using firefly algorithm with tree-based classification in software defect prediction. J Electron Electromed Eng Med Inform. 2023;5(4). doi:10.35882/jeeemi.v5i4.315

How to Cite: Sindhu K, Raju M, Sneha Y. Neuro-hybrid Adaptive Cybersecurity Orchestration for Real-time Zero-day Resilience and Intelligent Threat Response. Int Res J Multidiscip Scope. 2026;7(2):1711-1732. DOI: 10.47857/irjms. 2026.v07i02.09007