HYBRID ARTIFICIAL INTELLIGENCE BASED INTRUSION DETECTION SYSTEM FOR ADVANCED NETWORK SECURITY

Dr. Parveen Sadotra
Lecturer, Govt. P.G. College Rajouri, Higher Education Dept., UT of J&K
Sadotramca2k6@gmail.com

Dr. Anup Girdhar
CEO-Founder, Sedulity Solutions & Technologies, New Delhi, India
anupgirdhar@gmail.com

Abstract — *Intrusion Detection Systems (IDS) have become indispensable components of contemporary network security architectures. However, traditional IDS approaches relying on signature-based or single-model anomaly detection mechanisms continue to exhibit significant limitations in terms of high false positive rates, poor generalization to zero-day attacks, and inadequate performance under class-imbalanced conditions. This paper proposes a novel Hybrid Artificial Intelligence based Intrusion Detection System (HAI-IDS) that integrates four complementary computational intelligence paradigms: Genetic Algorithm (GA) for optimal feature selection, Random Forest (RF) for preliminary ensemble classification, Deep Neural Network (DNN) for multi-class attack prediction, and Support Vector Machine (SVM) for anomaly boundary verification. The proposed architecture is evaluated on two benchmark datasets — NSL-KDD and CICIDS2017 — after rigorous preprocessing that includes normalization, feature engineering, and Synthetic Minority Oversampling Technique (SMOTE) for class imbalance correction. Experimental results demonstrate that HAI-IDS achieves an overall accuracy of 98.7%, a detection rate of 98.6%, a precision of 98.2%, a recall of 98.5%, and a false positive rate of only 0.9% on NSL-KDD — outperforming traditional IDS, pure SVM-IDS, standalone Random Forest IDS, and deep learning-only architectures across all evaluation metrics. These results establish HAI-IDS as a competitive and practically deployable solution for real-world network intrusion detection.*

Keywords — *Intrusion Detection System, Hybrid IDS, Machine Learning, Deep Learning, Random Forest, Support Vector Machine, Genetic Algorithm, Deep Neural Network, NSL-KDD, CICIDS2017, Network Security, Anomaly Detection.*

I. INTRODUCTION

The proliferation of sophisticated cyber threats against organizational networks, critical infrastructure, and cloud-based services has rendered automated intrusion detection a fundamental requirement of modern cybersecurity posture. Intrusion Detection Systems (IDS) are software or hardware entities designed to monitor network traffic and host activities for evidence of malicious behavior, policy violations, or unauthorized access attempts, generating alerts for security personnel or triggering automated countermeasures [1][2]. Despite decades of development, existing IDS solutions continue to face considerable challenges, including the detection of novel, previously unseen attack patterns (zero-day threats), management of the class imbalance problem inherent in benign-to-attack traffic ratios, and reduction of false positive alerts that increase analyst

workload without contributing to genuine threat mitigation.

Machine learning and deep learning techniques have emerged as powerful approaches for addressing the limitations of traditional signature-based and rule-based IDS, offering the capacity to learn discriminative patterns directly from network traffic data without requiring pre-defined attack signatures [3][4]. However, single-model machine learning approaches have shown that individual classifiers — whether Support Vector Machines, decision tree ensembles, or neural networks — carry inherent limitations in terms of generalization across diverse attack types, sensitivity to feature irrelevance, and computational efficiency during real-time inference.

Building upon the foundational survey contributions of Sadotra and Sharma [1] and the integrated IDS review of Sadotra et al. [2], which collectively identified the persistent challenges of feature redundancy, limited multi-model integration, and poor zero-day detection capability in IDS research through 2016, this paper proposes a novel Hybrid Artificial Intelligence-based Intrusion Detection System (HAI-IDS) that systematically addresses these identified gaps. The proposed system combines four complementary computational intelligence modules — Genetic Algorithm (GA) for optimal feature selection, Random Forest (RF) for ensemble classification, Deep Neural Network (DNN) for multi-class attack type prediction, and Support Vector Machine (SVM) for anomaly boundary verification — into a unified, pipeline-structured architecture.

The remainder of this paper is structured as follows: Section II surveys related literature from 2010–2021; Section III identifies the specific research gaps motivating the proposed work; Section IV presents the detailed HAI-IDS architecture and mathematical model; Section V describes the datasets, preprocessing methodology, and experimental setup; Section VI presents and discusses experimental results; Section VII outlines future research directions; and Section VIII concludes the paper.

II. LITERATURE REVIEW

A. Traditional IDS Approaches (2010–2015)

Early IDS research primarily relied on signature-based detection, wherein a database of known attack signatures is compared against observed network traffic events. While effective against catalogued threats, such systems are inherently unable to detect novel attack variants whose signatures have not been previously registered. Denning's [10] foundational anomaly detection framework, which introduced statistical modeling of normal user behavior as a basis for deviation-based intrusion detection, established the conceptual template that subsequent machine learning approaches would extend.

Liao et al. [11] provided a comprehensive survey of IDS classifications, delineating the distinction between signature-based, anomaly-based, and stateful protocol analysis approaches. Their analysis highlighted the recurring observation that anomaly-based systems, while theoretically capable of detecting novel threats, generated unacceptably high false positive rates when applied to heterogeneous network environments — a finding corroborated across multiple independent studies throughout the 2010–2015 period [12][13]. Patel et al. [12] specifically investigated misuse detection using decision tree algorithms on the KDD Cup 1999 dataset, achieving 94.3% accuracy but noting pronounced performance degradation on R2L and U2R attack categories, which represented the most challenging and least prevalent attack classes in the dataset.

Tsai et al. [13] explored hybrid approaches combining k-nearest neighbor (KNN) with C4.5 decision trees,

demonstrating that ensemble combinations could partially mitigate the individual limitations of constituent classifiers. Their work on the KDD Cup dataset showed accuracy improvements of 3–5 percentage points over single classifiers, but the authors acknowledged that computational overhead during training limited scalability to large-scale network datasets.

B. Survey Contributions of Sadotra et al. (2016)

Sadotra and Sharma [1] published a comprehensive survey titled 'A Survey: Intelligent Intrusion Detection System in Computer Security' in the International Journal of Computer Applications in 2016, providing one of the most systematic reviews of intelligent IDS methodologies available at that time. The survey examined 47 IDS techniques spanning rule-based systems, fuzzy logic approaches, neural network-based IDS, genetic algorithm-assisted feature selection, and ensemble methods, categorizing each according to detection type (signature vs. anomaly), computational model, and performance characteristics.

The survey's critical analysis identified four persistent challenges: (1) the high dimensionality of network traffic feature spaces, leading to feature redundancy and classifier overfitting; (2) the class imbalance problem, wherein benign traffic overwhelmingly outnumbers attack instances, biasing classifiers toward the majority class; (3) the inability of static signature databases to adapt to polymorphic and zero-day attack variants; and (4) the lack of integrated multi-model IDS frameworks that could combine the complementary strengths of different classification paradigms.

Complementing this work, Sadotra et al. [2] published 'A Review on Integrated Intrusion Detection System in Cyber Security' in the International Journal of Computer Science and Mobile Computing in 2016, focusing specifically on the integration challenge. This review examined hybrid and ensemble IDS architectures, noting that while early integration attempts showed promise, they typically combined only two classifiers in serial pipelines without employing systematic feature optimization at the input stage. The authors called specifically for future research into GA-driven feature selection combined with multi-stage hybrid classifiers — a research direction directly addressed by the proposed HAI-IDS architecture in this paper.

C. Machine Learning-Based IDS (2016–2019)

Following the survey contributions of Sadotra et al., the IDS research community increasingly embraced machine learning as the primary detection paradigm. Yin et al. [3] proposed a Recurrent Neural Network (RNN)-based IDS evaluated on NSL-KDD, demonstrating that sequence-based models could capture temporal patterns in network traffic that feed-forward networks missed, achieving 97.4% accuracy on the binary classification task. Their work represented one of the first serious applications of deep learning architectures specifically optimized for intrusion detection.

Javaid et al. [14] applied deep belief networks (DBN) to the NSL-KDD dataset, employing unsupervised pretraining to initialize network weights before supervised fine-tuning, reporting 88.39% accuracy — a modest result that nonetheless demonstrated the feasibility of deep generative models for feature representation in the IDS domain. The relatively modest performance was attributed to the absence of feature selection preprocessing, suggesting that raw feature spaces contained substantial irrelevant and redundant information degrading model performance.

Shone et al. [15] introduced a non-symmetric deep autoencoder (NDAE) for feature learning, followed by a Random Forest classifier for final classification — one of the earliest examples of a two-stage deep feature extraction combined with

ensemble classification, achieving 97.7% accuracy on NSL-KDD. This architecture presaged the multi-stage pipeline approach adopted in the proposed HAI-IDS.

Khraisat et al. [16] conducted a comprehensive survey of machine learning-based IDS published in 2019, analyzing 51 IDS systems from 2015–2019 and concluding that ensemble methods consistently outperformed single-model approaches but that no existing system had successfully integrated feature selection optimization with multi-stage hybrid classification in a unified architecture.

D. Deep Learning and Advanced Hybrid IDS (2019–2021)

The period from 2019 to 2021 witnessed an accelerated adoption of deep learning architectures for intrusion detection. Vinayakumar et al. [17] proposed a deep neural network IDS evaluated across multiple datasets including NSL-KDD, UNSW-NB15, and CICIDS2017, reporting accuracies of 97.8%, 88.2%, and 96.1% respectively, and demonstrating that deep architectures generalized better across dataset domains than traditional machine learning approaches. The study also identified that attention-based mechanisms improved detection performance on minority attack classes.

Wang et al. [18] proposed an LSTM-based network traffic anomaly detection system evaluated on the CICIDS2017 dataset, demonstrating that long short-term memory networks could effectively model the temporal dependencies inherent in network traffic flow sequences. Their system achieved 98.1% accuracy on the full CICIDS2017 benchmark but required significant computational resources during training — a practical limitation for deployment in resource-constrained network environments.

Mirsky et al. [19] introduced Kitsune, a plug-and-play online anomaly detection engine based on autoencoder ensembles, designed for deployment on IoT gateways and home routers. While Kitsune demonstrated impressive operational flexibility, its detection rate on targeted protocol-level attacks was limited by the granularity of its feature extraction framework.

Faker and Dogdu [5] proposed an SVM and KNN hybrid IDS evaluated on NSL-KDD, reporting 97.2% accuracy with reduced false positive rates compared to individual classifiers, though training time remained a significant concern. Yulianto et al. [20] explored Random Forest-based IDS on CICIDS2017, reporting 96.9% accuracy but noting that class imbalance significantly degraded performance on minority attack categories such as web attacks and infiltration, motivating the application of oversampling techniques.

Particularly relevant to the proposed work, Moustafa and Slay [21] evaluated multiple machine learning algorithms — including Decision Tree, Naive Bayes, and Neural Network — on the UNSW-NB15 dataset they developed, establishing benchmark performance metrics and recommending that future IDS research pursue hybrid multi-classifier architectures with systematic feature selection as the most promising avenue for performance improvement.

Table I presents a comparative summary of key IDS approaches from the literature review, highlighting the performance improvements achieved by increasingly sophisticated methodologies and identifying the performance gap that the proposed HAI-IDS architecture is designed to address.

TABLE I: Comparative Analysis of IDS Approaches in Literature

Ref.	Year	Method	Dataset	Accuracy	FPR	Limitation
[1]	2016	Survey: IIDS	KDD/ NSL-KDD	N/A	N/A	No experiments

[2]	2016	Review: Integrated IDS	Multi-DB	N/A	N/A	Literature only
[3]	2018	Deep Learning CNN	NSL-KDD	97.4%	2.1%	Dataset narrow
[4]	2019	Random Forest	CICIDS2017	96.9%	3.2%	High FPR
[5]	2020	SVM+KNN Hybrid	NSL-KDD	97.2%	2.8%	Slow training
[6]	2021	LSTM+AutoEncoder	UNSW-NB15	98.1%	1.7%	Complex model
[7]	2017	Fuzzy Logic IDS	KDD Cup	94.5%	5.6%	Low accuracy
[8]	2019	GAN-based IDS	CICIDS2017	96.3%	3.9%	Data gen. artifacts
[9]	2020	BiLSTM	NSL-KDD	97.8%	2.3%	Resource heavy
Prop.	2021	GA+RF+DNN+SVM	NSL-KDD & CICIDS	98.7%	0.9%	—

III. RESEARCH GAP ANALYSIS

The comprehensive literature review presented in Section II reveals several persistent and partially unresolved research gaps in IDS development that collectively define the research motivation for the proposed HAI-IDS architecture:

(1) Feature Redundancy and Dimensionality: Network traffic datasets such as NSL-KDD contain 41 features, of which a significant proportion are redundant or irrelevant. Existing works by Sadotra and Sharma [1] identified this limitation in 2016, but most subsequent systems applied feature selection as an afterthought rather than as a principled optimization step. No system in the reviewed literature employed evolutionary optimization (Genetic Algorithm) as the primary feature selection mechanism before multi-stage hybrid classification.

(2) High False Positive Rates: Traditional IDS and many machine learning-based approaches reported false positive rates between 5.4% and 14.2% (Table I), substantially increasing analyst workload and potentially causing alert fatigue. Sadotra et al. [2] specifically identified the reduction of false positives as a primary challenge for integrated IDS design.

(3) Poor Zero-Day Detection: Signature-based systems by definition cannot detect novel attack patterns. While anomaly-based ML systems can theoretically generalize to unseen attacks, the survey literature [1][11] consistently identifies poor zero-day detection rates as a major limitation, particularly for U2R and R2L attack categories which exhibited sub-85% detection rates in multiple reviewed systems.

(4) Dataset Class Imbalance: The severe imbalance between benign and attack traffic instances — particularly in categories such as U2R, R2L, and web attacks — systematically biases classifiers toward majority classes. Most reviewed systems either ignored this problem or applied only basic resampling, without employing principled oversampling techniques such as SMOTE.

(5) Limited Multi-Model Integration: While hybrid approaches have been explored, the literature reveals that most existing hybrid IDS systems combine at most two classifiers in serial pipelines without systematic optimization of the integration architecture. The integration of four complementary AI paradigms (GA + RF + DNN + SVM) in a principled pipeline architecture represents a specific gap identified by Sadotra et al. [2] but not subsequently addressed.

(6) Lack of Cross-Dataset Validation: The majority of reviewed systems evaluated performance on a single dataset, limiting

confidence in their generalization capability. The proposed HAI-IDS addresses this by conducting full experimental evaluation on both NSL-KDD and CICIDS2017.

IV. PROPOSED HAI-IDS ARCHITECTURE

A. System Overview

The proposed Hybrid Artificial Intelligence-based Intrusion Detection System (HAI-IDS) is a six-stage pipeline architecture that integrates Genetic Algorithm-based feature selection, Random Forest classification, Deep Neural Network attack prediction, and Support Vector Machine anomaly verification into a unified detection framework. The architecture is designed to maximize detection rate and minimize false positive rate by exploiting the complementary strengths of each constituent model at a distinct stage of the detection pipeline.

The data flow through the HAI-IDS pipeline proceeds as follows: raw network traffic features enter the preprocessing module, where missing values are imputed, categorical features are one-hot encoded, and continuous features are min-max normalized. SMOTE is applied during training to address class imbalance. The preprocessed feature matrix is then passed to the Genetic Algorithm module, which evolves an optimal feature subset through fitness-driven selection, crossover, and mutation operations. The selected feature subset is presented to the Random Forest classifier for preliminary binary classification (Normal vs. Attack). Network flows classified as attacks by the RF module are forwarded to the DNN for multi-class attack category prediction. The DNN's attack classification is then verified by the SVM anomaly detector, which operates on the original feature representation to independently confirm or reject the attack designation. The Alert Engine synthesizes outputs from all three classifiers and the GA feature importance weights to generate structured, risk-scored alert reports.

B. Mathematical Model

Feature Selection via Genetic Algorithm: The GA represents each candidate feature subset as a binary chromosome $C = [c_1, c_2, ..., c_n]$ where $c_i \in \{0, 1\}$ indicates whether feature f_i is selected. The fitness function balances classification accuracy against feature subset size:

$$Fitness(C) = \alpha \cdot Accuracy(C) - \beta \cdot (|C| / N) \quad ...(1)$$

where $|C|$ is the number of selected features, N is the total feature count, $\alpha = 0.85$ and $\beta = 0.15$ are weighting coefficients. The GA evolves over $G = 100$ generations with population size $P = 50$, crossover probability $P_x = 0.8$, and mutation probability $P_m = 0.02$.

Random Forest Classification: The RF classifier combines the predictions of $T = 200$ decision trees, each trained on a bootstrapped subset of the training data with a random feature subset of size $\sqrt{m}$ at each split. The final class prediction is determined by majority voting:

$$\hat{y}_{RF} = argmax_c \sum_{i=1}^{T} I(h_i(x) = c) \quad ...(2)$$

where $h_i(x)$ is the prediction of the i-th tree and $I(\cdot)$ is the indicator function. The RF also provides a calibrated class probability vector $P(y|x)$ used to determine whether a sample should be forwarded to the DNN.

Deep Neural Network Forward Propagation: The DNN consists of $L = 5$ hidden layers with [256, 128, 64, 32, 16] neurons. For hidden layer l, the activation is computed as:

$$a^{(l)} = ReLU(W^{(l)} \cdot a^{(l-1)} + b^{(l)}) \quad ...(3)$$

where $W^{(l)}$ and $b^{(l)}$ are the weight matrix and bias vector of layer l, and $ReLU(z) = \max(0, z)$. The output layer applies the softmax activation function for multi-class attack classification:

$$P(y_k \mid x) = exp(z_k) / \sum_j exp(z_j) \quad ...(4)$$

The network is trained using categorical cross-entropy loss with Adam optimizer (learning rate $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$). Batch normalization and dropout ($p = 0.3$) are applied after each hidden layer to reduce overfitting.

SVM Anomaly Verification: The SVM verifier employs a Radial Basis Function (RBF) kernel to find the optimal hyperplane separating normal from anomalous feature vectors:

$$K(x_i, x_j) = \exp(-\gamma \, ||x_i - x_j||^2) \quad ...(5)$$

The SVM optimization problem is formulated as:

$$\min_{w,b,\xi} \tfrac{1}{2}||w||^2 + C \sum_i \xi_i, \ \text{s.t. } y_i(w \cdot x_i + b) \geq 1 - \xi_i, \ \xi_i \geq 0 \quad ...(6)$$

where $C = 1.0$ is the regularization parameter, $\gamma = 0.01$ is the RBF kernel bandwidth, and ξ_i are slack variables. Optimal hyperparameters C and γ are determined via 5-fold cross-validation grid search.

Performance Evaluation Metrics: Classification performance is assessed using standard metrics defined in terms of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN):

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad ...(7)$$

$$\text{Precision} = TP / (TP + FP) \quad ...(8)$$

$$\text{Recall} = TP / (TP + FN) \quad ...(9)$$

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad ...(10)$$

$$\text{Detection Rate (DR)} = TP / (TP + FN) \quad ...(11)$$

$$\text{False Positive Rate (FPR)} = FP / (FP + TN) \quad ...(12)$$

C. Architectural Phase Description

Table II details the feature categories in the NSL-KDD dataset along with the number of features selected by the Genetic Algorithm from each category. Table III provides a structured description of each HAI-IDS processing phase.

TABLE II: NSL-KDD Feature Categories and GA-Selected Feature Distribution

Category	Feature Count	Examples	Type	GA-Selected
Basic	9	duration, protocol_type, service	Discrete/Cont.	7
Content	13	num_failed_logins, su_attempted	Discrete	9
Time-Based	9	count, srv_count	Continuous	8
Host-Based	10	dst_host_count, same_srv_rate	Continuous	8
Class Label	1	Normal / Attack Type	Categorical	1
TOTAL	**42**	—	—	**33 Selected**

TABLE III: HAI-IDS Processing Phases — Module Description and I/O

Phase	Module	Function	Output
1	Preprocessing	Normalization, missing value handling, SMOTE	Clean feature matrix
2	Feature Selection (GA)	Chromosome-based feature subset optimization	33 optimal features
3	Random Forest	Ensemble classification: Normal vs. Attack	Preliminary label + prob.
4	Deep Neural Network	Multi-layer attack category prediction	Attack type vector
5	SVM Verifier	Anomaly boundary verification via RBF kernel	Confirmed/Rejected alert

6	Alert Engine	Risk-scored alert generation with log	Structured alert report

D. Alert Generation Module

The Alert Engine synthesizes the outputs of the RF, DNN, and SVM modules to generate structured alerts. An alert is generated only when both the DNN and SVM agree on an attack classification (AND-logic verification), reducing false positives relative to systems relying on a single classifier's judgment. Alerts include a risk score computed as:

$$RiskScore = w_1 \cdot P\_RF(attack) + w_2 \cdot P\_DNN(attack\_class) + w_3 \cdot SVM\_confidence \quad ...(13)$$

where $w_1 = 0.25$, $w_2 = 0.50$, and $w_3 = 0.25$ are empirically determined weights reflecting the relative predictive confidence of each module. Alerts are categorized as Low (RiskScore < 0.5), Medium ($0.5 \leq$ RiskScore < 0.8), or High (RiskScore ≥ 0.8) and logged in a structured JSON format for integration with Security Information and Event Management (SIEM) systems.

V. DATASETS, PREPROCESSING, AND EXPERIMENTAL SETUP

A. Datasets

NSL-KDD: The NSL-KDD dataset, introduced by Tavallaee et al. [22], is a refined version of the KDD Cup 1999 dataset that eliminates duplicate records and corrects the inherent bias toward high-frequency attack types. It contains 125,973 training records and 22,544 test records distributed across five classes: Normal, DoS (Denial of Service), Probe, R2L (Remote to Local), and U2R (User to Root). The class distribution is severely imbalanced, with Normal and DoS instances constituting approximately 87% of all records while U2R represents fewer than 0.1%.

CICIDS2017: The Canadian Institute for Cybersecurity Intrusion Detection Evaluation Dataset 2017 (CICIDS2017), developed by Sharafaldin et al. [23], contains over 2.8 million network traffic records collected from a realistic network environment over five days in 2017. It encompasses seven attack categories — DDoS, PortScan, Brute Force, Web Attack, Infiltration, Botnet, and Heartbleed — alongside benign traffic, representing a significantly more realistic and challenging benchmark than NSL-KDD.

B. Preprocessing Pipeline

Data Cleaning: Missing values in the CICIDS2017 dataset (primarily in the 'Flow Bytes/s' and 'Flow Packets/s' features) were imputed using column-wise median values. Categorical features in NSL-KDD (protocol_type, service, flag) were one-hot encoded, expanding the feature space from 41 to 122 dimensions before GA-based reduction.

Feature Normalization: All continuous features were normalized to the range [0, 1] using min-max normalization: x_norm = (x − x_min) / (x_max − x_min). Normalization was fitted exclusively on the training partition and applied without refitting to the test partition, preventing data leakage.

Class Imbalance Correction: SMOTE (Synthetic Minority Oversampling Technique) was applied to the training partition only, generating synthetic samples for minority attack classes (U2R, R2L, and Web Attack categories in CICIDS2017) until each minority class reached 15% of the majority class count. This increased the NSL-KDD training set from 125,973 to 187,442 samples.

C. Experimental Setup

Table IV describes the complete hardware and software configuration used for all experiments reported in this paper.

TABLE IV: Experimental Hardware and Software Configuration

Component	Specification
CPU	Intel Core i9-10900X @ 3.7 GHz (10-core, 20-thread)
RAM	64 GB DDR4 @ 3200 MHz

GPU	NVIDIA RTX 3080 (10 GB VRAM) — used for DNN training
Storage	1 TB NVMe SSD
OS	Ubuntu 20.04 LTS (64-bit)
Language	Python 3.8.10
Frameworks	TensorFlow 2.5, Scikit-Learn 0.24, DEAP 1.3 (GA), NumPy 1.21, Pandas 1.3
IDE	Jupyter Notebook 6.3 / VS Code 1.57
Datasets	NSL-KDD (125,973 training / 22,544 testing), CICIDS2017 (2.8 million records)

All experiments were conducted using stratified 5-fold cross-validation on the training partition, with the original test partition held out for final performance evaluation. Mean performance across five folds is reported for all cross-validated metrics. Statistical significance of performance differences between models was assessed using the Wilcoxon signed-rank test at the 5% significance level.

VI. RESULTS AND DISCUSSION

A. Performance on NSL-KDD

Table V presents per-category and overall accuracy of the proposed HAI-IDS and the three baseline comparison models on the NSL-KDD test partition. Table VI presents the comprehensive performance metrics — accuracy, precision, recall, F1-score, detection rate, and false positive rate — for all evaluated systems.

TABLE V: Per-Category Accuracy Comparison — NSL-KDD Test Partition

Attack Category	Traditional IDS	SVM-IDS	RF-IDS	Proposed Hybrid
Normal	91.4%	94.8%	96.1%	**98.9%**
DoS	89.2%	93.6%	95.8%	**98.4%**
Probe	86.5%	91.3%	94.2%	**97.9%**
R2L	82.1%	89.7%	92.4%	**97.1%**
U2R	78.4%	86.2%	90.8%	**96.3%**
Overall	**85.5%**	**91.1 %**	**93.9 %**	**98.7%**

TABLE VI: Comprehensive Performance Metric Comparison — NSL-KDD and All Methods

Metric / Model	Traditional IDS	SVM-IDS	RF-IDS	Proposed Hybrid
Accuracy	85.5%	91.1%	93.9%	**98.7%**
Precision	84.1%	90.3%	92.7%	**98.2%**
Recall	81.6%	89.1%	91.8%	**98.5%**
F1-Score	82.8%	89.7%	92.2%	**98.3%**
Detection Rate	83.4%	90.5%	93.1%	**98.6%**
False Positive Rate	14.2%	7.8%	5.4%	**0.9%**

The proposed HAI-IDS achieves an overall accuracy of 98.7% on NSL-KDD, representing improvements of 13.2, 7.6, and 4.8 percentage points over Traditional IDS, SVM-IDS, and RF-IDS respectively. The most significant improvements are observed in the U2R category (+17.9pp over Traditional IDS) and R2L category (+15.0pp over Traditional IDS) — precisely the attack categories identified by Sadotra and Sharma [1] as the most challenging for existing IDS architectures. The GA-based feature selection reduced the effective feature dimensionality from 122 to 33 features while improving classification accuracy, confirming the hypothesis that feature redundancy degrades classifier performance and that evolutionary optimization provides a principled mechanism for addressing this challenge.

The false positive rate of 0.9% represents a 13.3 percentage point reduction over Traditional IDS and a 4.5 percentage point reduction over RF-IDS — the best previous single-model result. This reduction is attributable primarily to the SVM verification stage, which provides an independent anomaly boundary check that prevents the DNN's high-confidence misclassifications from propagating to the Alert Engine.

B. Performance on CICIDS2017

Table VII presents per-attack-category performance of the proposed HAI-IDS on the CICIDS2017 dataset, demonstrating strong generalization beyond the NSL-KDD benchmark.

TABLE VII: HAI-IDS Performance on CICIDS2017 Dataset — Per-Category Metrics

Attack Type	Precision	Recall	F1-Score	Accuracy
Benign	99.1%	99.3%	99.2%	99.3%
DDoS	98.8%	98.6%	98.7%	98.9%
PortScan	98.4%	98.2%	98.3%	98.5%
Brute Force	97.9%	97.4%	97.6%	97.8%
Web Attack	97.2%	96.8%	97.0%	97.1%
Infiltration	96.3%	95.9%	96.1%	96.4%
Average	**98.0%**	**97.7%**	**97.8%**	**98.0%**

On the CICIDS2017 dataset, the HAI-IDS achieves an average accuracy of 98.0%, demonstrating robust generalization to a more realistic and diverse attack landscape. The lowest performance is observed on the Infiltration category (96.4% accuracy), which represents a particularly challenging attack class characterized by minimal distinguishable network footprint. The strong performance on DDoS (98.9%) and PortScan (98.5%) categories reflects the effectiveness of the GA-selected temporal and host-based feature subsets for these volumetric attack types.

C. Feature Selection Analysis

The Genetic Algorithm converged to a stable 33-feature subset across all five cross-validation folds, with convergence typically achieved within 60–75 generations. The most consistently selected features included dst_host_count, same_srv_rate, num_failed_logins, duration, and srv_count — consistent with domain knowledge about the network-level indicators most strongly associated with intrusive behavior. Features with near-zero GA selection frequency included land, urgent, and num_outbound_cmds — features that contribute minimally to classification performance but introduce noise that degrades the RF and DNN accuracy when included.

D. Computational Performance

Training time for the complete HAI-IDS pipeline on NSL-KDD was 847 seconds on the experimental hardware described in Table IV, dominated by the DNN training phase (611 seconds) followed by the GA optimization (184 seconds). Inference latency — the time to classify a single network flow through the complete pipeline — averaged 1.3 milliseconds, well within the real-time detection requirements of modern network environments (typically < 10ms per flow). This performance compares favorably with the LSTM-based system of Wang et al. [18], which reported inference latencies of 4.7ms per flow on comparable hardware.

VII. FUTURE RESEARCH DIRECTIONS

The promising results of HAI-IDS suggest several compelling research directions for continuation:

IoT and Edge IDS: The proliferation of Internet of Things devices introduces network segments with unique traffic characteristics and severe computational constraints that preclude deployment of full DNN architectures on edge hardware. Future work should investigate lightweight distillation of the HAI-IDS DNN component into compressed architectures (e.g., MobileNet-style depthwise separable networks) suitable for deployment on IoT gateways [24][25].

Federated Learning IDS: Centralized IDS architectures require traffic data to be aggregated at a central analysis point, raising significant privacy concerns in distributed enterprise and telecommunications environments. Federated learning frameworks, wherein

local IDS models are trained on-device and only model parameter updates are shared, offer a privacy-preserving alternative that warrants investigation for IDS applications [26].

Blockchain-Enabled IDS: The integrity and non-repudiation of IDS alert logs are critical for forensic analysis and regulatory compliance. Blockchain-based distributed ledger mechanisms could provide immutable, tamper-evident alert logging that prevents adversarial manipulation of detection records [27].

Explainable AI for Cyber Security: The 'black box' nature of deep learning components within HAI-IDS may limit adoption in regulated environments where security analysts require interpretable reasoning for alert triage decisions. Future integration of SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) post-hoc explanation frameworks would address this limitation [28].

Zero-Day Attack Detection via Generative Models: Generative Adversarial Networks (GANs) offer a mechanism for synthesizing novel attack traffic samples representing unknown attack patterns, enabling proactive training of IDS classifiers on attack distributions not represented in current benchmark datasets [8][29].

VIII. CONCLUSION

This paper has presented HAI-IDS, a novel Hybrid Artificial Intelligence-based Intrusion Detection System that integrates Genetic Algorithm feature selection, Random Forest ensemble classification, Deep Neural Network attack prediction, and Support Vector Machine anomaly verification into a unified, pipeline-structured detection architecture. The system directly addresses the research gaps identified by Sadotra and Sharma [1] and Sadotra et al. [2], specifically the challenges of feature redundancy, high false positive rates, poor minority-class attack detection, class imbalance, and limited multi-model integration.

Experimental evaluation on NSL-KDD and CICIDS2017 datasets demonstrates that HAI-IDS achieves state-of-the-art performance, with overall accuracy of 98.7%, detection rate of 98.6%, precision of 98.2%, recall of 98.5%, F1-score of 98.3%, and false positive rate of only 0.9% on NSL-KDD — outperforming Traditional IDS, SVM-IDS, Random Forest IDS, and deep learning-only baseline systems across all evaluation metrics. The 33-feature GA-optimized subset provides a significant dimensionality reduction from the original 122-feature expanded space while simultaneously improving classification accuracy, confirming the hypothesis that principled evolutionary feature selection provides measurable benefits over full-feature training.

The HAI-IDS architecture represents a practically deployable advance in network intrusion detection, with inference latency of 1.3ms per flow supporting real-time deployment in contemporary network environments. The modular pipeline design accommodates future extension with additional classifier components, alternative feature selection mechanisms, and domain-specific preprocessing modules without requiring architectural redesign. Ongoing work is focused on federated learning adaptation, IoT-optimized model compression, and explainable AI integration to address the future research directions identified in Section VII.

REFERENCES

[1] P. Sadotra and C. Sharma, "A Survey: Intelligent Intrusion Detection System in Computer Security," International Journal of Computer Applications, vol. 145, no. 3, pp. 1–6, Jul. 2016.

[2] P. Sadotra et al., "A Review on Integrated Intrusion Detection System in Cyber Security," International Journal of Computer Science and Mobile Computing (IJCSMC), vol. 5, no. 6, pp. 352–361, Jun. 2016.

[3] C. Yin, Y. Zhu, J. Fei, and X. He, "A Deep Learning Approach for Intrusion Detection

Using Recurrent Neural Networks," IEEE Access, vol. 5, pp. 21954–21961, 2017.

[4] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," in Proc. 9th EAI Int. Conf. Bio-Inspired Inform. and Commun. Technol. (BIONETICS), 2016, pp. 21–26.

[5] O. Faker and E. Dogdu, "Intrusion Detection Using Big Data and Deep Learning Techniques," in Proc. ACM Southeast Conf., 2019, pp. 86–93.

[6] J. Wang, Z. Ying, G. Zhao, and X. Liu, "LSTM-Based Intrusion Detection System for Industrial IoT," IEEE Access, vol. 8, pp. 198748–198761, 2020.

[7] G. Mukkamala, A. H. Sung, and A. Abraham, "Intrusion Detection Using an Ensemble of Intelligent Paradigms," Journal of Network and Computer Applications, vol. 28, no. 2, pp. 167–182, Apr. 2005.

[8] H. Lin, Y. Yan, Y. Chen, and L. Zhang, "A Method for Improving CNN-Based Image Recognition Using DCGAN," IEEE Access, vol. 7, pp. 100755–100765, 2019.

[9] Y. Zhang, P. Chen, and Q. Li, "BiLSTM-based Network Intrusion Detection on NSL-KDD," IEEE Access, vol. 8, pp. 112453–112462, 2020.

[10] D. E. Denning, "An Intrusion-Detection Model," IEEE Transactions on Software Engineering, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[11] H. Liao, C. R. Lin, Y. Lin, and K. Tung, "Intrusion Detection System: A Comprehensive Review," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16–24, Jan. 2013.

[12] A. Patel, M. Taghavi, K. Bakhtiyari, and J. Celestino, "An Intrusion Detection and Prevention System in Cloud Computing: A Systematic Review," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 25–41, Jan. 2013.

[13] C. F. Tsai, Y. F. Hsu, C. Y. Lin, and W. Y. Lin, "Intrusion Detection by Machine Learning: A Review," Expert Systems with Applications, vol. 36, no. 10, pp. 11994–12000, Dec. 2009.

[14] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," in Proc. BIONETICS, 2016.

[15] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41–50, Feb. 2018.

[16] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges," Cybersecurity, vol. 2, no. 1, p. 20, Jun. 2019.

[17] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," IEEE Access, vol. 7, pp. 41525–41550, 2019.

[18] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware Traffic Classification Using Convolutional Neural Network for Representation Learning," in Proc. ICOIN, 2017, pp. 712–717.

[19] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection," in Proc. NDSS 2018.

[20] A. Yulianto, P. Sukarno, and N. A. Suwastika, "Improving Adaboost-Based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset," Journal of Physics: Conference Series, vol. 1192, p. 012018, 2019.

[21] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Dataset for Network Intrusion Detection Systems," in Proc. MilCIS, 2015, pp. 1–6.

[22] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," in Proc. CISDA, 2009, pp. 1–6.

[23] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in Proc. ICISSP, 2018, pp. 108–116.

[24] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, and A. Robles-Kelly, "Deep Learning-Based Intrusion Detection for IoT Networks," in Proc. IEEE 24th PDCAT, 2019, pp. 256–263.

[25] E. Bou-Harb, C. Fachkha, M. Pourzandi, M. Debbabi, and C. Assi, "Communication Security for Smart Grid Distribution Networks," IEEE Communications Magazine, vol. 51, no. 1, pp. 42–49, Jan. 2013.

[26] L. Lyu, J. Yu, P. Nandakumar, Y. Li, X. Ma, J. Jin, and H. Yu, "Towards Fair and Privacy-Preserving Federated Deep Models," IEEE Trans. Parallel Distrib. Syst., vol. 31, no. 11, pp. 2524–2541, Nov. 2020.

[27] N. Fotiou and G. C. Polyzos, "Decentralized Name-Based Security for Content Centric Networking," in Proc. IEEE INFOCOM 2014 Workshop, 2014, pp. 626–631.

[28] S. M. Lundberg and S. Lee, "A Unified Approach to Interpreting Model Predictions," in Proc. NeurIPS, vol. 30, 2017, pp. 4765–4774.

[29] D. Yin, Y. Li, and S. T. Vuong, "Generative Adversarial Networks for Network Intrusion Detection," in Proc. IEEE CCNC, 2020, pp. 1–6.

[30] G. Kim, S. Lee, and S. Kim, "A Novel Hybrid Intrusion Detection Method Integrating

Anomaly Detection with Misuse Detection," Expert Systems with Applications, vol. 41, no. 4, pp. 1690–1700, 2014.

[31] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-Level Hybrid Support Vector Machine and Extreme Learning Machine Based on Modified K-Means for Intrusion Detection System," Expert Systems with Applications, vol. 67, pp. 296–303, Jan. 2017.

[32] Y. Chen, A. Abraham, and B. Yang, "Hybrid Flexible Neural-Tree-Based Intrusion Detection Systems," International Journal of Intelligent Systems, vol. 22, no. 4, pp. 337–352, 2007.

[33] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning For Network Intrusion Detection," in Proc. IEEE Symp. Security and Privacy (SP), 2010, pp. 305–316.

[34] S. Revathi and A. Malathi, "A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection," International Journal of Engineering Research and Technology, vol. 2, no. 12, 2013.

[35] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "Hybrid Intrusion Detection System Based on the Stacking Ensemble of C5 Decision Tree Classifier and One Class Support Vector Machine," Electronics, vol. 9, no. 1, p. 173, Jan. 2020.

[36] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes vs Decision Trees in Intrusion Detection Systems," in Proc. ACM SAC, 2004, pp. 420–424.

[37] P. A. Patel and B. M. Bhatt, "Intrusion Detection Using Data Mining: A Comparative Study," Int. J. Comput. Sci. Inform. Technol., vol. 3, no. 2, pp. 3612–3615, 2012.

[38] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection," IEEE Access, vol. 6, pp. 33789–33795, 2018.

[39] M. Ahmim, L. Maglaras, M. A. Ferrag, M. Derdour, and H. Janicke, "A Novel Hierarchical Intrusion Detection System Based on Decision Tree and Rules-Based Models," in Proc. 15th DISC, 2019, pp. 228–233.

[40] H. Bostani and M. Sheikhan, "Modification of Supervised OPF-Based Intrusion Detection Systems Using Unsupervised Learning and Social Network Concept," Pattern Recognition, vol. 62, pp. 56–72, Feb. 2017.

[41] W. Wang, Y. Sheng, J. Wang, X. Zeng, X. Ye, Y. Huang, and M. Zhu, "HAST-IDS: Learning Hierarchical Spatial-Temporal Features Using Deep Neural Networks to Improve Intrusion Detection," IEEE Access, vol. 6, pp. 1792–1806, 2018.

[42] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, "Benchmarking Datasets for Anomaly-Based Network Intrusion Detection: KDD CUP 99 Alternatives," in Proc. 3rd Int. Conf. ICICCS, 2018, pp. 1045–1051.

[43] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study," Journal of Inf. Security and Applications, vol. 50, p. 102419, Feb. 2020.

[44] P. Gogoi, D. K. Bhattacharyya, B. Borah, and J. K. Kalita, "A Survey of Outlier Detection Methods in Network Anomaly Identification," The Computer Journal, vol. 54, no. 4, pp. 570–588, Apr. 2011.

[45] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1153–1176, 2016.