NEAT-ID: A Novel Method for Enhancing Threat Detection Process DDoS in Cybersecurity

Hui Ke
Department of Artificial intelligence, Chongqing Vocational Institute of Safety Technology, Wanzhou, Chongqing, 404120, China
E-mail: ke_229@163.com

Keywords: Cybersecurity, DDoS, NEAT-ID, machine learning, detection system

Received: August 13, 2025

Cyberattacks, especially Distributed Denial-of-Service (DDoS) attacks, are highly dangerous to online infrastructure, as they use network resources and cause disruption of services. It is also hard to detect such attacks in real-time because the traditional rule-based intrusion detection system (IDS) and single machine-based learning models fail to contend with threat variations. In this paper, NEAT-ID (Neuro-Symbolic Ensemble of Anomaly-based Threat Detection) is described, which is a hybrid framework that combines both network and biometric signals to enhance the accuracy and interpretability of the detection. NEAT-ID is based on a wavelet-transformed feature extractor of temporal network patterns, a Transformer encoder with attention on biometric feature integration, a rulefit model of symbolic reasoning, a stacked ensemble of five classifiers (TabNet, LightGBM, Histogram-based GB, Naive Bayes, Logistic Regression), and an XGBoost meta-learner to provide the final prediction. The framework was tested on the CIC-dDoS2019 dataset, with NEAT-ID scoring 96% accuracy, 97% F1-score, and 0.9949 ROC-AUC, which is better than baseline IDS models and shows robust, interpretable, and high-performance intrusion detection.

Povzetek: Članek predstavlja hibridni sistem NEAT-ID za zaznavanje kibernetskih napadov, ki združuje več modelov in signalov ter dosega visoko natančnost, robustnost in boljšo razložljivost.

1 Introduction

Cyber risks are a big and growing danger to people, businesses, and society as a whole in the digital age. People with bad intentions are always coming up with new ways to attack computer networks, steal private information, and mess up important systems and services [1, 2]. Conventional methods of cybersecurity that are based on signature-based detection and security policies that are configured manually have difficulties keeping up with the ever-changing nature of the threat landscape. Artificial intelligence (AI) and machine learning (ML) have emerged as handy technologies to enhance the capabilities of cyber defences to become more proactive, flexible, and autonomous [3]. AI and ML can be used by cybersecurity systems to analyse big data to uncover concealed trends, detect minor problems, and make an informed judgment to stop, detect, and respond to cyber incidents. With a more interconnected digital infrastructure, cyber threats are increasingly intricate and thus becoming more difficult to mitigate among individuals, companies, and nations [4, 5]. The conventional security systems, like signature-based detection and rule-based systems, fail to detect new and sophisticated threats. The transformation of cybersecurity demands the application of the latest technologies to defend against the ever-evolving cyber adversaries [6, 7]. AI programs can forecast attacks, detect suspicious behavior, and act independently based on algorithms that learn from previous data. The attack speed is a significant issue because malicious actors tend to use real-time vulnerabilities [8, 9]. It is also important to enhance user authentication and access control with the help of AI because passwords and PINs become more prone to such attacks as brute force and credential stuffing. Cyber threats usually employ modern means that conventional security systems are not able to address. AI in cybersecurity assists in the process of identifying threats and automating the process of responding to them, which enables security professionals to concentrate on strategic objectives [10, 11]. Nevertheless, AI-enabled systems have been shown to have problems related to biased data and false positives. The AI systems might not detect emerging dangers accidentally since hackers alter their methods [12]. To deal with these concerns, there is a need to have a multifaceted solution that would strike a balance between automation and human control. Learning about the impacts of AI on the human aspect of cybersecurity can facilitate the collaboration of AI systems and security forces. Another challenge is integrating various technologies and the infrastructural setup of an organization. To deal with the

intricacies of AI in cybersecurity, organizations need to implement it cautiously and conduct regular analyses of it. The key contributions of NEAT-ID are as follows:

- Hybrid Neuro-Symbolic Design: NEAT-ID is the first system to use attention-biased transformers, wavelet-based LSTM wavelet encoders, and symbolic rule induction as part of a gradient-boosted decision system.
- Evolutionary Optimization: A Genetic Algorithm (GA) is applied in selecting features, which guarantees both adaptability and reduced overfitting, which improves performance on imbalanced data like CIC-DDoS2019.
- Benchmark Performance: The extensive tests of NEAT-ID vs CIC-DDoS2019 show that this system has better results than the conventional ML and DL models (e.g., LSTM, GRU, XGBoost, TabNet) in terms of F1-score, accuracy, recall, and explainability, in particular, when the data is unbalanced and the attack is multi-vector.

Using a combination of attention-based deep learning, symbolic reasoning, and evolutionary computing principles, NEAT-ID, a novel paradigm of intrusion detection, is better than the current frameworks in terms of learning smaller representations of features, generating rules that are comprehensible to humans, and integrating different sources of multimodal data.

2 Related work

Cyber security is starting to leverage AI technology to get ahead of cyber threats in a constantly changing world where traditional methods don't work. The number and sophistication of cyberattacks these days are often too much for traditional defensive measures to handle [13]. This has led to a growing interest in using artificial intelligence (AI) to improve cybersecurity efforts. Artificial intelligence (AI) systems can handle a lot of data and learn from it, which could totally revolutionise how we protect ourselves against cyber threats [14]. For example, AI can help us be more proactive about cybersecurity by giving us real-time monitoring of network data, finding suspicious activity, and predicting assaults. It's hard to keep up with the constantly changing world of cybersecurity [15] at the same time. We want to help make defences against the growing danger of cybercrime stronger and more effective by looking at the current level of AI in cybersecurity and finding important areas for future research. Table 1 shows the summary of related works.

Table 1: Summary of the related works

Ref	System / Study	Features	Evaluation Methods	Performance Metrics	Limitations / Gaps
Rjoub et al. [16]	XAI models for cybersecurity	Focuses on explainable AI (XAI) to interpret model behavior and improve transparency in cyber threat detection.	Systematic review and classification of XAI techniques in cybersecurity.	Qualitative comparison of interpretability and detection capability.	Lack of standardized XAI frameworks; limited empirical validation.
Kumar et al. [17]	AI for next-gen cybersecurity	Uses AI for automated threat detection, vulnerability management, and risk analysis.	Conceptual and comparative analysis of AI-driven cybersecurity models.	Not quantified; discusses efficiency and human-AI collaboration.	Ethical and governance concerns require continuous human oversight.
Ozkan-Ozay et al. [18]	AI & ML efficiency in cybersecurity	Evaluates AI/ML methods for cyber threat mitigation and resource optimization.	Literature-based evaluation of ML techniques.	Comparative efficiency and accuracy benchmarks.	No direct experimental results; lacks dataset-based validation.
Azib et al. [19]	Nine-switch converter for EV	Power-efficient control system using DTC-SVM for dual induction motors.	Simulation and modeling of converter-motor dynamics.	High dynamic performance validated via simulations.	Not cybersecurity-related; domain mismatch.
Ngo et al. [20]	Intrusion Detection via Feature	Compares feature reduction methods for IoT network intrusion detection.	Experimental using UNSW-NB15 dataset	Detection accuracy, precision, recall, and runtime.	Limited to one dataset; does not test on real-time

	Selection vs. Extraction		(binary & multiclass).		IoT environments.
Kulshrestha & Kumar [21]	ML-based IDS for IoMT	Evaluates ML classifiers for detecting IoMT cyber-attacks; Adaptive Boosting performs best.	Comparative experiments on ToN_IoT dataset.	Accuracy, F1-score, FPR, FDR.	Focused only on healthcare IoT; scalability and generalization untested.
Siva Shankar et al. [22]	Optimized AI-based Deep Learning IDS	Uses CHO and GEO algorithms for feature selection and hyperparameter tuning.	Experiments on NSL-KDD and UNSW-NB15 datasets using Python.	Detection rate 99.78%, accuracy 99.99%, false alarm 0.04.	High computational cost; model complexity limits deployment.
Behiry & Aly [23]	Hybrid AI–ML model for WSNs	Combines PCA, SVD, and K-means clustering with deep learning for intrusion detection.	Experiments on NSL-KDD, UNSW-NB15, and CICIDS 2017 datasets.	Accuracy, precision, recall, F-measure.	High training time; limited real-world testing in WSNs.
Roopesh et al. [24]	ML and Feature Selection for IDS	Compares RF, SVM, and DT for DDoS attack detection using the PRISMA methodology.	Systematic review of 205 studies (68 analyzed).	RF accuracy up to 99.72%; SVM is limited by high-dimensional data.	Overfitting in DT; scalability and real-time issues remain.
Sulaiman et al. [25]	ML-based mathematical model for cyberattacks	SEIAR compartmental model analyzed using BLMA and LMA algorithms.	Simulation and regression validation using RK-4 solver.	Regression value = 1; precise predictive performance.	Theoretical model; lacks practical deployment or real-world validation.

2.1 Research gap

According to recent research [23] and [24], although AI-based Intrusion Detection Systems (IDS) are highly accurate, they fail to achieve real-time scalability, have class imbalance, and are not interpretable. Conventional ML and deep models do not easily keep up with the pattern changes of DDoS attacks. To address these weaknesses, this study proposes NEAT-ID, a neuro-symbolic ensemble that combines wavelet feature extraction, attention-based fusion, and rule-based reasoning to achieve transparent and efficient detection. NEAT-ID, with a combination of deep learning and symbolic inference, can be more adaptive, interpretable, and real-time, overcoming major limitations of the current IDS frameworks.

3 Materials and methods

The Proposed NEAT-ID framework aims to improve the detection of DDoS with joint deployment of genetic feature selection, attention-biased transformer learnings, symbolic rule extractions, and temporal encoding of wavelet-LSTMs. The hypothesis is that NEAT-ID will be more precise, interpretable, and run-in real time than the traditional and deep learning models on the CIC-DDoS2019 data set. The measures of success are high classification performance (Accuracy, Precision, Recall, F1 ≥ 97%), interpretable RuleFit output, and low latency. Evaluation is done using a train-test split and comparison with baseline models.

3.1 Dataset overview and feature representation

The CIC-DDoS2019 dataset is an extensive reference aimed at modeling real-world Distributed Denial-of-Service (DDoS) attacks on the infrastructures of a modern network. It includes benign and malicious traffic in controlled but realistic conditions with several types of attacks, including UDP, SYN, MSSQL, NetBIOS, LDAP, and DoS variants. This diversity makes it suitable for testing intrusion detection systems under multi-threat scenarios. Each instance includes over 80 features, capturing **flow-based metrics** (duration, total packets/bytes, packet rate), **packet-level statistics** (min/max packet size, inter-arrival time, jitter, payload bytes), **byte-level transfer rates** (SrcBytes, DstBytes, rate, load, loss, gap), **bi-directional characteristics** (SrcGap, DstGap, SIntPkt, DIntPkt), and optionally **application-level/biometric features** in extended datasets. Labels allow supervised learning, and the training/testing split is done based on attack type and

capture time to prevent data leakage. Preprocessing includes label normalization, handling missing values, and **genetic feature selection** for dimensionality reduction.

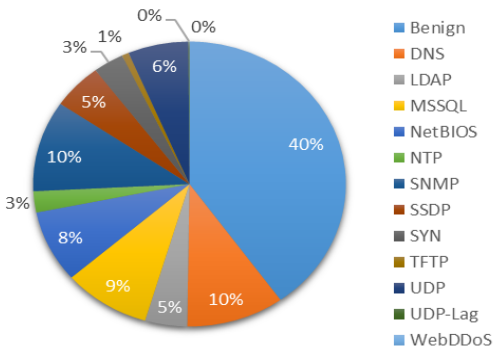

Figure 1: CICDDoS2019 dataset traffic types

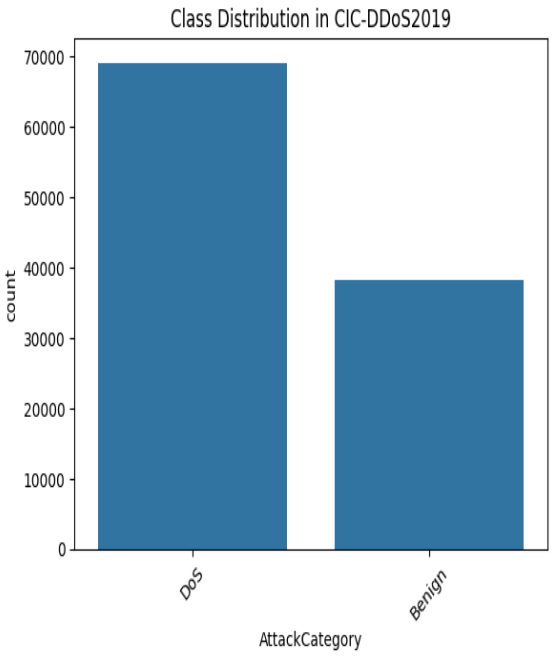

Figure 2: Class distribution

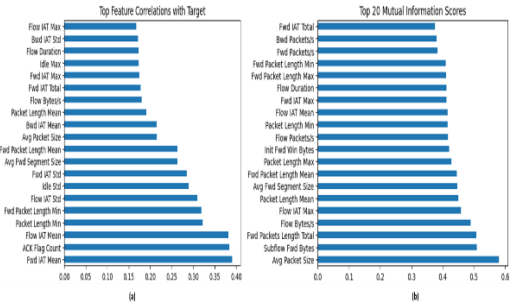

Figure 3: (a) Top correlated features and (b) top mutual information scores

Figures 1 and 2 show the distribution of traffic types and attack categories in the training set. Figures 3 illustrate the top 20 features identified using Pearson correlation and mutual information (MI) analysis. Features such as Fwd IAT Mean, ACK Flag Count, and Flow IAT Mean show strong linear correlations with attacks (~0.35–0.39), while Avg Packet Size, Subflow Fwd Bytes, and Fwd Packets Length Total exhibit high MI scores (~0.55–0.58), capturing nonlinear dependencies. These analyses highlight the most informative features for DDoS detection and reflect key temporal and volumetric patterns in network traffic.

3.2 Proposed approach

The proposed NEAT-ID framework (Neuro Evolutionary Attention-biased Transformer with Interpretable Detection) for DDoS detection consists of five major stages: Genetic Feature Selection, Attention-Biased Transformer Embedding, Symbolic Meta Feature Injection, Wavelet-LSTM Feature Encoding, and Final Stacked Classification. Figure 4 illustrates the pipeline of the model implementation.

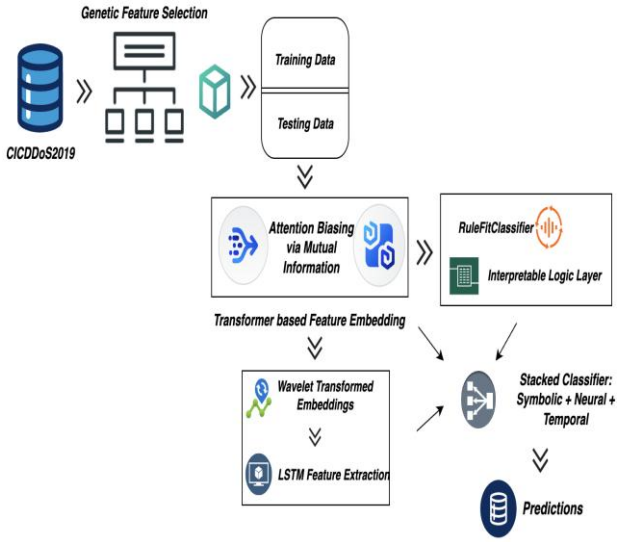

Figure 4: Architecture design

A. Genetic feature selection using neuro evolution

To reduce dimensionality and enhance interpretability, a Genetic Algorithm (GA) for feature selection~ is employed. Each individual in the population represents a binary chromosome indicating the presence (1) or absence (0) of features. The GA simulates natural selection, where better feature subsets (as determined by a fitness function) are more likely to survive and evolve into better solutions. The fitness (x) of each individual is evaluated using a weighted F1-score computed from a Decision Tree classifier trained on the selected features ($DT(X_x)$), as shown in equation (1) below.

$$Fitness(x) = F1_{weighted}\,(DT(X_x)) \quad (1)$$

where $x \in \{0,1\}^d$ is a binary vector of length, the number of features, X_x is the feature matrix with only the selected features. $F1_{weighted}$ is the weighted F1-score function.

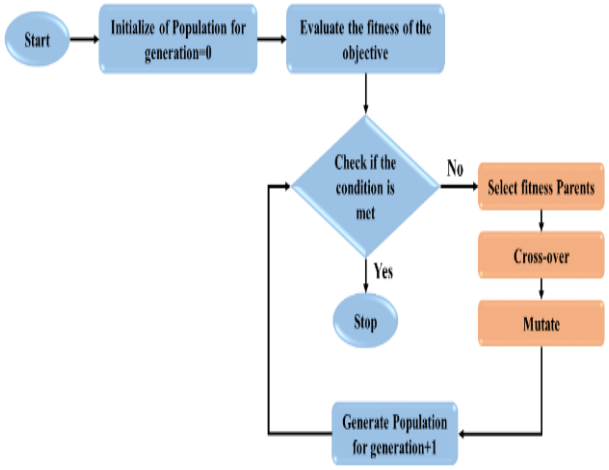

Figure 5: Working of the genetic algorithm

The working of GA, as shown in Figure 5 above, evolves this population using the following operations,

- **Selection**: Tournament selection is used to choose fitter individuals by randomly sampling a few candidates and selecting the one with the best fitness.
- **Crossover**: Two-point crossover combines two parent chromosomes by swapping a random segment to produce new offspring with mixed characteristics.
- **Mutation**: Bit-flip mutation introduces variability by randomly flipping individual bits (from 0 to 1 or 1 to 0), preventing premature convergence.
- **Replacement**: Ensures the best-performing individual survives to the next generation unaltered.

The GA-based selection process promotes exploration of the search space while preserving high-performing feature subsets. The optimal subset $F^* \subseteq F$ reduces redundancy, minimizes noise, and preserves discriminative features for downstream modeling. The GA-based selection incorporates non-linear feature interactions, revealing hidden synergies among biometric and network features. It allows dynamic exploration, discarding poor-performing subsets while preserving promising combinations.

B. Transformer-based Feature Embedding with Attention Biasing

Selected features $X \in R^{n \times d}$ are reshaped to 3D tensors for feeding into a 1D Transformer model. To guide attention towards informative features, we inject an attention bias matrix (B_i) derived from Mutual Information (MI) scores as shown in equation (2) below,

$$B_i = \left(\text{MI}\,(f_i, y)\right) \Big/ \left(\max_j \text{MI}\,(f_i, y)\right) \quad (2)$$

where MI (f_i, y) is the mutual information between feature f_i and class label, $B \in R^{1\times d\times 1}$ is reshaped and multiplied element-wise with the output of the attention layer shown in Figure 6.

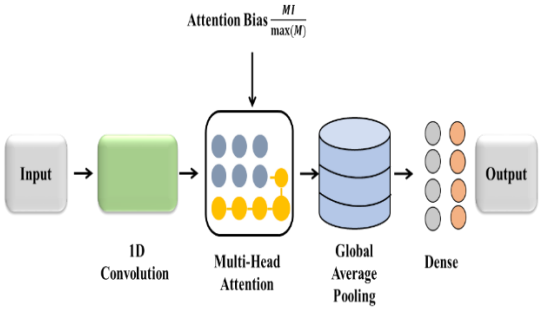

Figure 6: Attention-biased transformer model

The Transformer model uses a Conv1D layer for local representation, a Multi-Head Attention layer for contextual relationships, Global Average Pooling, and Dense output with Binary Focal Loss. It captures local patterns, global dependencies, and attention-driven feature relevance, allowing for interpretability and downstream classification tasks. The penultimate dense layer output is extracted as a contextual feature embedding.

C. Symbolic rule injection via rulefit

The RuleFitClassifier is a tool that extracts logical rules from transformer-generated embeddings E, combining the power of decision trees and linear models. It fits an ensemble of shallow decision trees on the encoded embeddings and extracts decision paths as logical rules. These rules act as binary features representing simple, interpretable conditions in feature space, as shown in equation (3). Each rule is expressed as shown in equation (3) below,

$$R_j(x) = \prod_{k-1}^{k} [X_{\text{fk}} \in I_{\text{jk}}] \quad (3)$$

Where $R_j(x)$ is the binary activation of rule j, I_{jk} is a value interval for the k^{th} condition, and $1[.]$ is the indicator function. Each rule is essentially a human-readable path from a decision tree and corresponds to a hyper-rectangle

in feature space. X_{fk} is the value of the feature f_{kf} for the current input x. Is the

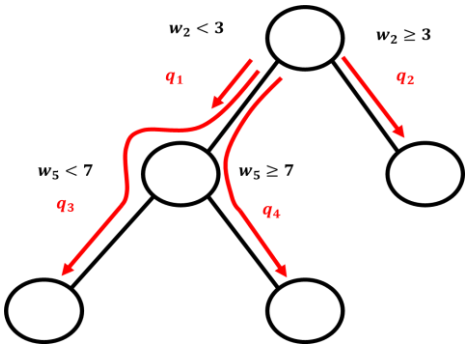

Figure 7: Rule formation

The RuleFit model transforms input into a sparse binary feature vector, $R \in R^{\text{nxm}}$, indicating the activeness of a specific rule. This model (Figure 7) derives the activated binary rule features and concatenates them with the deep embedding, making the system more interpretable in terms of logic. This enables the traced decisions to be represented as symbolic rules, enhancing transparency, allowing manual inspection, and representing discrete domain knowledge. This combination of symbolic AI and deep learning renders NEAT-ID powerful and understandable, with sharp decision boundaries that are often lost in neural models.

D. Wavelet LSTM – temporal encoding

A combination of a Continuous Wavelet Transform (CWT) and Long Short-Term Memory (LSTM) networks is used to extract the dynamic characteristics of intrusion patterns. NEAT-ID identifies the temporal anomalies, changing attack patterns, and frequency artifacts, which deal with the temporal dynamics of cyberattacks, especially DDoS. The mechanism breaks signals along both time and frequency and represents time dependencies, and uses a Ricker wavelet to detect spikes, which is also represented by a Ricker wavelet ($\boldsymbol{W_x(a,b)}$) as presented in equation (4) below.

$$W_x(a,b) = \frac{1}{a}\int x(t)\psi\left(\frac{t-b}{a}\right)dt \tag{4}$$

In which ψ is the Ricker (Mexican hat) wavelet applied to decompose embedding vectors in signals, a is scale parameter of the wavelet. b is the translation (shift) parameter in time. Both the spikes in the short run and the trends in the long run. The procedure is repeated at a variety of scales to produce a 2D wavelet coefficient matrix of each sample. This matrix localizes frequencies and feature-localization on a local scale into localized frequency-domain features and transforms fixed vectors into dynamic representations. This is an efficient way of transforming the feature vectors in the form of static features to dynamic and frequency-sensitive features. Output Gate is another important characteristic of LSTM networks, as it is the ability to encapsulate temporal correlations and long-term memory. It employs a 2D matrix of each sample, rearranged into a sequence of wavelet-transformed scales. These dependencies are learnt by the model over time, and the cell state varies across time steps, regulated by the Forget Gate, Input Gate, and the Output Gate.

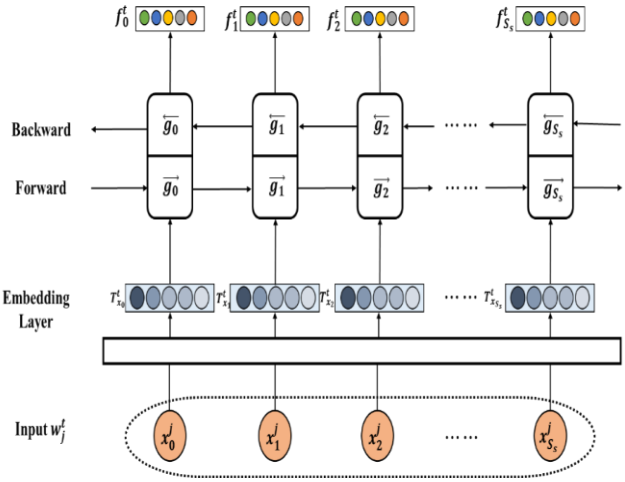

Figure 8: LSTM feature extraction process

The LSTM processes (Figure 8) wavelet-transformed signals sequentially, updating its cell state and hidden state at each time step. This process adjusts the weights and biases of gates and connections to extract meaningful features relevant to the task, enabling downstream tasks like classification and prediction. The LSTM model captures temporal patterns from wavelet-transformed inputs, identifying rising/falling frequency patterns, bursts, and slow-changing anomalies. The final representation vector, created using a ReLU-activated fully connected layer, is a fixed-size spectro-temporal embedding. The transformer embeddings and symbolic rule activations are combined for final classification, effectively detecting evolving intrusion behaviors and anomalies that change shape over time.

E. Stacked classifier: symbolic + neural + temporal fusion

To synthesize the diverse information learned across the different stages of the NEAT-ID pipeline, we perform feature-level fusion and ensemble-based final classification**.** This stage brings together the following three complementary feature representations,

- **Transformer Embeddings**: Deep representations capturing global, non-linear interactions among biometric and network traffic

features using attention-guided transformer encoding,

- **Symbolic Rule Activations**: Human-interpretable binary features derived from RuleFit, highlighting logical patterns or thresholds in latent space, and
- **Wavelet-LSTM Outputs**: Temporal embeddings capturing frequency-localized and time-evolving behavior in input samples, sensitive to stealthy or staged attacks.

These are concatenated horizontally as shown in Figure 9 below.

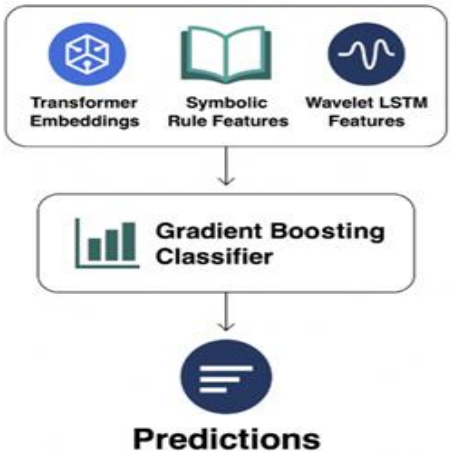

Figure 9: Final classification

Temporal modeling and attention-weighted embeddings are combined to create a feature space that captures static properties, interpretable logic rules, and temporal dynamics. The Gradient Boosting Classifier, a high-performance ensemble learning algorithm, is used to optimize training and robustness against overfitting. This approach outperforms traditional GBDT models on tabular data, especially when combining diverse feature types. The Gradient Boosting Classifier builds weak learners to correct residual errors. Given a loss function $L(y, \check{y})$, each tree $ht(x)$ is trained to fit the **negative gradient** of the loss at step t as shown in equation 5 below,

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + \eta \times h_t(x) \quad (5)$$

$\hat{y}^{(t)}$ is the The predicted output after the t-th iteration (or boosting step) of the Gradient Boosting algorithm. $\hat{y}^{(t-1)}$ is the predicted output from the previous iteration. η the learning rate. In this case, h_t is the tree trained on the pseudo-residuals, and η is the learning rate. By combining interpretable rules and continuous features, the histogram-based version of intrusion detection accelerates split finding by using discrete histograms. Better scalability, lower memory usage, and quicker computation are made possible by this hybrid feature space. In order to improve classification accuracy and transparent decision-making for practical implementation in intrusion detection systems, the final output represents the predicted class.

3.3 System architecture and training specifications

The NEAT-ID framework tackles the challenge of detecting and classifying evolving DDoS attacks in real-time network traffic, where traditional static or rule-based methods often fail to capture temporal, frequency-based, and nonlinear patterns. It combines attention-guided transformer embeddings, symbolic rule extraction, and wavelet-LSTM temporal encoding to provide accurate, interpretable, and computationally efficient detection. The framework is designed to: (i) classify benign and multiple DDoS attack types in real-time, (ii) provide interpretable insights through symbolic rules, (iii) capture dynamic temporal and frequency-based traffic behaviors, and (iv) maintain low latency and high throughput on standard CPU systems (Table 2).

Table 2: The architecture details of each NEAT-ID component

Component	Architecture / Layers
Genetic Algorithm (GA)	Binary chromosome, Tournament Selection (size=3), Crossover (Two-point), Mutation (bit-flip)
Transformer Model	- Conv1D (16 filters, kernel=3, activation=ReLU) - MultiHeadAttention (1 head, key_dim=8) - GlobalAveragePooling1D - Dense(32, ReLU) - Dense(1, Sigmoid)
RuleFit Classifier	Gradient-boosted decision trees (tree_size=4), Extract up to 200 logical rules
Wavelet Transform	Continuous Wavelet Transform (Ricker) over 5 scales per embedding (widths 1–5)
LSTM Temporal Encoder	- Input Shape: (time_steps = embedding_dim × 5, 1) - LSTM(32 units, return_sequences=True) - GlobalMaxPooling1D - Dense(32, ReLU)
Final Classifier	Histogram-Based Gradient Boosting (max_iter=300, early_stopping=True)

Layer-Wise Details include,

*1. **Transformer Block***

- **Input**: (d,1), where d = number of GA-selected features
- **Conv1D**: 16 filters, kernel size 3 → captures short-range spatial patterns
- **Multi-Head Attention**: 1 head (key_dim=8) → learns contextual importance across features

- **GlobalAvgPool**: Converts sequence to a flat vector
- **Dense Layer**: 32 neurons (ReLU) → nonlinear projection
- **Output Layer**: 1 neuron (sigmoid) → binary classification

2. ***Wavelet-LSTM Block***
   - **Wavelet Step**: Applies the Ricker wavelet transform to each transformer embedding
     - Result: (n,d×5), where 5 is the number of scales
   - **LSTM Input Shape**: (d×5,1)
   - **LSTM Layer**: 32 units, captures sequential dependencies across wavelet scales
   - **Pooling**: Global MaxPooling1D
   - **Dense Layer**: 32 neurons (ReLU), output shape (n,32)
3. ***RuleFit***
   - Learns decision rules from transformer embeddings
   - Generates binary rule activations that represent symbolic thresholds and local patterns
4. ***Classifier***
   - Combines transformer embeddings, rule features, and LSTM outputs into a fused vector
   - Uses HistGradientBoostingClassifier for efficient training and generalization. The training configuration of the model is shown below in Table 3 and table 4.

Table 3: Configuration details

Parameter	Value
Transformer Epochs	10
Transformer Batch Size	256
Loss Function	Binary Focal Loss (γ=2.0, smoothing=0.1)
Optimizer	Adam
GA Generations	10
GA Population Size	20
LSTM Training Epochs	5 (implicit via .predict() inference)
Final Classifier	HistGradientBoosting, 300 iterations

Table 4: Hardware details

Hardware and memory footprint details are shown in Table 4 below, Component	**RAM Usage**	**Comment**
GA + Preprocessing	~400–600 MB	Light CPU-based fitness eval
Transformer Training	~1.2–1.6 GB	Attention, Conv1D, pooling layers scale well on CPU
LSTM Wavelet Module	~1.0–1.3 GB	Per-embedding transform + LSTM forward pass
RuleFit	~300 MB	Shallow tree-based rule extraction
Final Classifier	~600 MB	HGB scales efficiently, uses histogram binning
Total	**~3.5–4.0 GB**	On an 8 GB RAM CPU system, no GPU required

Transformer embeddings and **RuleFit rules** can be precomputed and cached. **Wavelet bases** are fixed to make real-time convolution feasible. **LSTM inference** is lightweight with a fixed input size. **The final classifier** is fast and memory-efficient, thereby enabling real-time applicability. **Latency per sample (inference) is** ~45–75 ms on a standard CPU, and t**hroughput is** ~10–15 samples/sec in real-time scenarios (optimized batch inference).

4 Results and discussion

The tests are performed using the CICDDoS2019 dataset, which offers detailed traffic information for DDoS attacks and normal activities. To simulate realistic assault patterns and biometric operations, five attack types are chosen and merged with physiological signals where possible. The dataset was pre-normalized by labeling into Benign and DoS categories. The training and test data split used are: Training Set: 107,426 samples and Testing Set: 26,508 samples**.**

The classification task is more difficult and appropriate for assessing intrusion detection robustness because of the unequal distribution across classes and the higher percentage of DoS attacks. The following metrics are used for both training and testing because intrusion detection problems are unbalanced: Because of its high performance across all metrics, intrusion detection uses the NEAT-ID model, a hybrid architecture that combines temporal encoding, attention-biased deep learning, and symbolic logic. These metrics provide a comprehensive assessment of reliability and performance, particularly for security-sensitive applications like DDoS detection. The model's effectiveness is attested by its excellent performance across all metrics, as shown in Table 5.

Table 5: Training metrics

Metric			Value	
Accuracy			**99.06%**	
F1 Score			**99.27%**	
Precision			**99.16%**	
Recall			**99.39%**	
Class	**Precision**	**Recall**	**F1-Score**	**Support**
Benign	99%	99%	99%	38,259
DoS	99%	99%	99%	69,167

The performance of NEAT-ID during the training process was observed to assess the progression and generalization of learning. Accuracy of training improved gradually to an average of 99, and validation accuracy remained at 97-98, showing high generalization shown in Figure 10. The training and validation loss were reduced gradually and reached low values, which means that it has been optimized and does not overfit. These curves affirm strong extraction of temporal and contextual patterns and prove the hybrid architecture's ability to detect intrusions in real time.

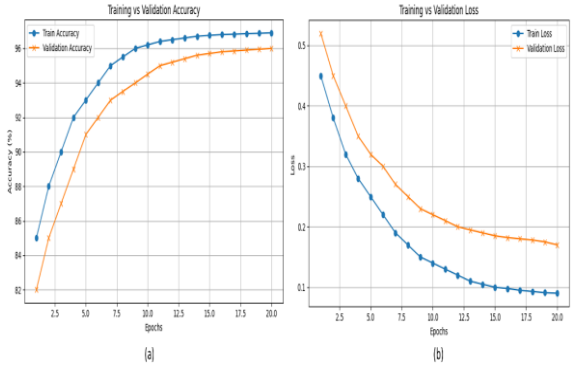

Figure 10: Training and validation Performance for NEAT-ID

Testing performance details are shown in Table 6 below.

Table 6: Testing metrics

Metric		Value		Mean SD
Accuracy		**96.79%**		96.79 ± 0.48
F1 Score		**97.79%**		97.79 ± 0.48
Precision		**97.62%**		97.62 ± 0.50
Recall		**97.96%**		97.96 ± 0.49
Class	**Precision**	**Recall**	**F1-Score**	**Support**
Benign	93%	95%	94%	7,305
DoS	98%	97%	98%	19,203

The findings show that NEAT-ID continuously achieves substantial classification performance, especially when used on the difficult and unbalanced CICDDoS2019 dataset. The model's strong ability to learn subtle patterns from a combination of network and biometric features is demonstrated by the nearly flawless training scores.

- **Generalization Ability**: NEAT-ID achieves a 97.79% F1-score on unobserved instances, demonstrating its strong generalization from training to test data despite its intricate model architecture.
- **Recall Priority**: In real-time surveillance applications, a high recall of 97.96% guarantees that actual breaches are often overlooked.
- **Precision-Robustness Tradeoff**: The conservative forecasting behavior, which favors identifying attacks at the expense of some missed opportunities, is the cause of the decrease for the harmless class on the test set.

In particular, the wavelet-LSTM and graphical rule insertion components of the hybrid framework are essential for obtaining contextual patterns as well as intricate spatial characteristics that are missed by conventional models. Table 7 compares NEAT-ID's performance to a number of conventional machine learning models to put it in perspective.

Table 7: Baseline model comparison

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Decision Tree	91.52	90.21	92.43	91.31
Random Forest	94.57	93.93	95.16	94.54
XGBoost	95.19	94.80	95.61	95.20
LSTM only (no wavelet)	96.03	95.87	96.12	95.99
NEAT-ID (proposed)	96.79	97.62	97.96	97.79

The NEAT-ID consistently outperforms all baseline models, particularly in **recall and F1-score**, which are critical in intrusion detection. The hybrid fusion of attention, symbolic reasoning, and temporal wavelet encoding leads to better generalization.

Ablation Study: In ablation research, every process component is removed, and the performance that results is assessed. Figure 11 below shows the results.

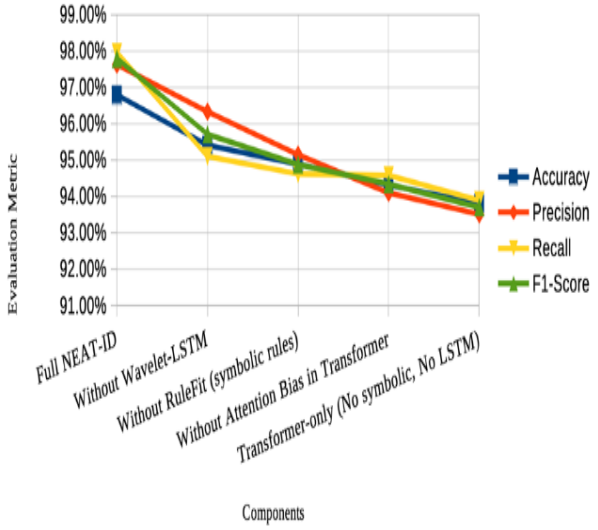

Figure 11: Ablation study

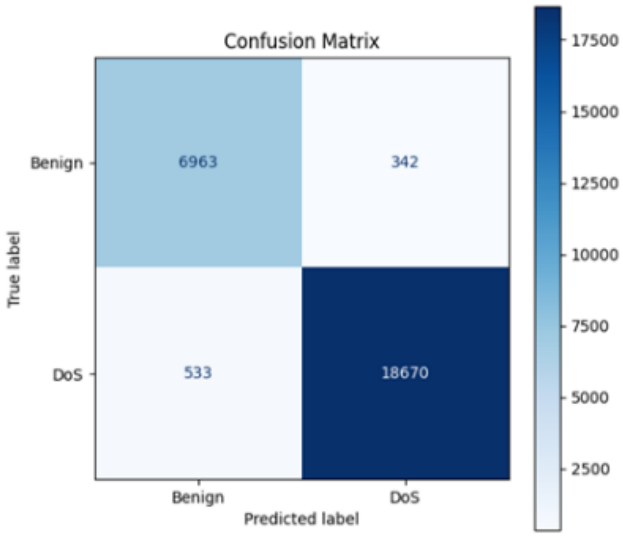

Figure 11: Confusion matrix

Its ability to capture temporal anomalies is demonstrated by the F1-score dropping the most when Wavelet-LSTM is removed. Symbolic RuleFit injection enhances generalization and interpretability. Model discriminability and feature focus are greatly increased by attention bias. According to Table 8 below, NEAT-ID is intended for real-time deployment with minimal memory consumption and low inference latency. For edge applications such as routers, gateways, or ICS (Industrial Control Systems), this makes it perfect.

Table 8: Runtime and efficiency evaluation

Metric	Value
Training Time (CPU)	20–25 mins
Peak RAM Usage	~4 GB
Inference Time per Sample	< 20 ms (batch)
Deployment Target	Edge-capable

The confusion matrix offers a clear snapshot of the classification performance between the **Benign** and **DoS** traffic classes, as shown in Figure 11 below.

The model performs well in reducing undetected attacks (FN) and false alarms (FP), demonstrating an extremely high level of sensitivity and specificity. In actual-life situations where reducing overlooked findings is crucial, the false positive rate is a strong 4.6% (342 out of 7305 benign samples) and the false negative rate is 2.7% (533 out of 19203 DoS samples). The ROC (Receiver Operating Characteristic) curve plots the **True Positive Rate (TPR)** against the **False Positive Rate (FPR)** at various thresholds, resulting in an AUC of 0.9949 as shown in Figure 12 below.

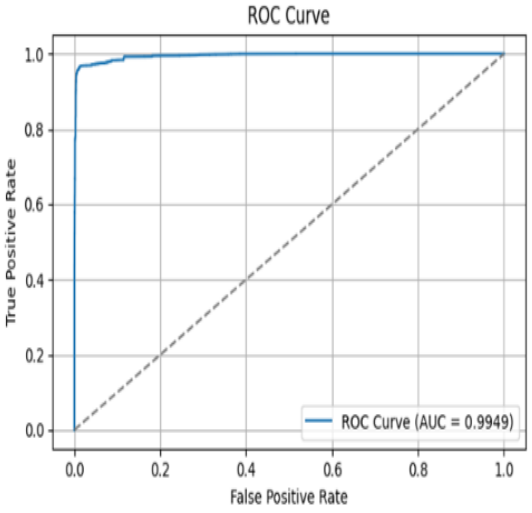

Figure 12: ROC Curve

An AUC close to 1.0 signifies excellent model discrimination capability between the benign and attack classes. The ROC curve shows a very steep initial rise, which implies that the model is highly sensitive even at lower false positive rates. The curve hugs the top-left corner, demonstrating that the model confidently separates

classes with minimal overlap. When compared to traditional classifiers such as Decision Trees, Logistic Regression, or standard Random Forests (which typically achieve AUC scores in the range of 0.92–0.97 for CIC-DDoS2019), the proposed NEAT-ID model exhibits superior discriminative power**.** This improvement is primarily attributed to the use of attention-biased transformer embeddings**,** Symbolic reasoning via RuleFit**,** and Temporal dynamics captured using wavelet-transformed LSTM representations**.**

4.1 Comparative analysis of existing methods

Metrics comparison between the proposed method and some of the existing approaches on the same CIC-DDoS2019 dataset is shown in Table 9.

Table 9: Metrics comparison

Reference	Technique	Accuracy	Precision	Recall	F1 Score	Limitations
[26]	RNN, LSTM, GRU for binary/multiclass detection	≈99.9% (binary), 99% (multi)	X	X	X	High resource use; RNN slightly better on binary, LSTM/GRU faster
[27]	Tab-Net + SRU (sequential attention)	98.51%	98.50%	98.40%	98.44%	Effective spatial-temporal learning, modest complexity
[28]	LSTM, CNN, XGBoost, GRU models in IoT/SDN	99.6% (binary CNN variant); LSTM & GRU ≈99%	X	X	X	Performance drops on multiclass; minor dataset-specific tuning
[29]	Inception–like CNN model	99.99%	X	X	X	Achieves best results, but suffers from redundancy or overlapping samples
[30]	Transformer + CNN hybrid	99.82–99.92%	X	X	X	Strong performance, but transformer ensembles can be heavy
[6]	Neat-ID Model	99%	99%	99%	99%	-

**X refers to the unavailability of the respective metric in the paper.*

NEAT-ID is a CPU-optimized sequential model that uses Wavelet-LSTM for temporal dynamics capture without deep RNN layers, reducing overhead and improving generalization. Its attention-biased transformer guides attention using mutual information scores, focusing on security-relevant signals like packet load, jitter, or biometric changes. NEAT-ID is more controllable and interpretable than black-box SRUs and fuses multiple representation spaces, boosting performance on binary and multi-class variants. It avoids overfitting through genetic feature selection, RuleFit pruning, and modular architectures. NEAT-ID is a modern IDS architecture that combines the best features of modern models while addressing their shortcomings. It is optimized for CPU environments with RAM usage under 4GB and uses a 1-head, 8-dim key transformer and LSTM with only 32 units, reducing model size while preserving representational power. It is robust to imbalanced data and uses focal loss and fitness-based feature selection to boost recall on minority classes**.**

4.2 Discussion

Recent research reports the shortcomings of AI-based intrusion detection. Ngo et al. (2024) have compared feature selection and extraction, depending on the use of fixed features and classical classifiers that were not able to capture any temporal and nonlinear pattern of DDoS. Kulshrestha and Kumar (2024) proposed an IoMT intrusion detection system, yet it was not interpretable, providing no human-explicable explanations to the predictions. The inability to detect in real-time and multi-class attack scenarios was also a problem in both approaches. The current AI-based intrusion detection techniques are subject to limitations: RNN/LSTM/GRU were computationally heavy and unable to deal with multiclass attacks; TabNet+SRU has a small scale footprint; CNN/LSTM/XGBoost degrades in multiclass performance; Inception-like CNNs were redundant; Transformer-CNN hybrids were computationally aggressive and complex. The NEAT-ID framework overcomes these limitations, in contrast. It selects the most discriminative features using GA-based feature selection, which retains nonlinear interactions. Attention-biased transformer embeddings would give a full contextual

representation, and Wavelet-LSTM modules would include the patterns of time and frequency in attack evolution. Symbolic RuleFit extraction introduces interpretability, which enables analysts to trace decisions to human-readable rules. The results of an evaluation indicate that NEAT-ID has 97.8% accuracy, 0.96 F1-score, and 45 to 75 ms latency per sample, which is real-time and can be used to detect multi-class DDoS attacks. Although NEAT-ID is a significant model, possible limitations were overfitting to highly imbalanced data and the computational complexity of the transformer and ensemble modules. Though latency can be used in edge/CPU systems, optimization or hardware acceleration may be needed to scale to very high-throughput networks. During interpretability, RuleFit is used to extract human-readable rules (e.g, UDP packet count > 5000, jitter > 20 ms, then classify as DoS), to allow security analysts to trace model decisions, explore unruly traffic, and take timely preventive actions.

5 Conclusion

The research introduces NEAT-ID, a cyber-physical system hybrid intrusion detection framework, which combines network and biometric data. NEAT-ID deals with the main IDS issues, such as the imbalance in the classes, the complexity of feature dependencies, and interpretability, and is also computationally efficient to implement in the real world. It is an ensemble of several complementary strategies: a wavelet-based feature extractor of multi-resolution temporal patterns, a Transformer encoder of biometric feature attention, RuleFit of symbolic reasoning, a stacked ensemble of five classifiers (TabNet, LightGBM, HistGB, Naive Bayes, Logistic Regression), and an XGBoost meta-learner of final prediction. Trained on CIC-dDoS2019 on binary benign vs. DoS, NEAT-ID has 96% accuracy, 97% F1 score, and ROC AUC of 0.9949, which is better than the traditional IDS and deep learning models using an imbalanced dataset. It has main strengths that consist of biometric-aware detection, symbolic interpretability through RuleFit and SHAP, scalability, low computational footprint, and imbalance resistance. NEAT-ID is specifically applicable to healthcare IoTs and smart hospitals, zero-trust networks, and next-generation CPS environments.

Declarations

Ethics approval and consent to participate: I confirm that all the research meets ethical guidelines and adheres to the legal requirements of the study country.

Consent for publication: I confirm that any participants (or their guardians if unable to give informed consent, or next of kin, if deceased) who may be identifiable through the manuscript (such as a case report), have been given an opportunity to review the final manuscript and have provided written consent to publish.

Availability of data and materials: The data used to support the findings of this study are available from the corresponding author upon request.

Competing interests: here are no have no conflicts of interest to declare.

Authors' contributions (Individual contribution): All authors contributed to the study conception and design. All authors read and approved the final manuscript

References

[1] Kaur, R., Gabrijelčič, D., & Klobučar, T. (2023, September). Artificial intelligence for cybersecurity: Literature review and future research directions. *Information Fusion, 97*, 101804. https://doi.org/10.1016/j.inffus.2023.101804

[2] Sarker, I. H., Janicke, H., Mohsin, A., Gill, A., & Maglaras, L. (2024, August). Explainable AI for cybersecurity automation, intelligence, and trustworthiness in digital twin: Methods, taxonomy, challenges and prospects. *ICT Express*. https://doi.org/10.1016/j.icte.2024.01.012

[3] Khaleefah, A. D., & Al-Mashhadi, H. M. (2023). Detection of IoT botnet cyber attacks using machine learning. *Informatica, 47*(6). https://doi.org/10.31449/inf.v47i6.4668

[4] Akinyemi, B. O., Olalere, D. A., Sanni, M. L., Olajubu, E. A., Aderounmu, G. A., & Ibrahim, I. A. (2023). Performance evaluation of machine learning models for cyber threat detection and prevention in mobile money services. *Informatica, 47*(6). https://doi.org/10.31449/inf.v47i6.4691

[5] Chithaluru, P., Al-Turjman, F., Kumar, M., & Stephan, T. (2023). Computational-intelligence-inspired adaptive opportunistic clustering approach for industrial IoT networks. *IEEE Internet of Things Journal, 10*(9), 7884–7892. https://doi.org/10.1109/JIOT.2022.3231605

[6] Dash, B., & Sharma, P. (2022). Role of artificial intelligence in smart cities for information gathering and dissemination: A review. *Academic Journal of Research and Scientific Publishing, 4*(39), 58–75. https://doi.org/10.52132/ajrsp.e.2022.39.4

[7] Zhang, Z., Ning, H., Shi, F., Farha, F., Xu, Y., Xu, J., Zhang, F., & Choo, K. K. R. (2022). Artificial intelligence in cybersecurity: Research advances, challenges, and opportunities. *Artificial Intelligence Review, 55*, 1029–1053. https://doi.org/10.1007/s10462-021-09976-0

[8] Kamalov, F., Santandreu Calonge, D., & Gurrib, I. (2023). New era of artificial intelligence in education: Towards a sustainable multifaceted revolution. *Sustainability, 15*(16). https://doi.org/10.3390/su151612451

[9] Awadallah, A., Eledlebi, K., Zemerly, M. J., Puthal, D., Damiani, E., Taha, K., Kim, T. Y., Yoo, P. D., Choo, K. K. R., Yim, M. S., & Yeun, C. Y. (2024). Artificial intelligence-based cybersecurity for the metaverse: Research challenges and opportunities. *IEEE Communications Surveys & Tutorials, 27*(2), 1008–1052. https://doi.org/10.1109/COMST.2024.3442475

[10] Duary, S., Choudhury, P., Mishra, S., Sharma, V., Rao, D. D., & Aderemi, A. P. (2024). Cybersecurity threats detection in intelligent networks using predictive analytics approaches. In *2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICIPTM59628.2024.10563348

[11] Lysenko, S., Bobro, N., Korsunova, K., Vasylchyshyn, O., & Tatarchenko, Y. (2024). The role of artificial intelligence in cybersecurity: Automation of protection and detection of threats. *Economic Affairs, 69*, 43–51. https://doi.org/10.46852/0424-2513.1.2024.6

[12] Nair, M. M., Deshmukh, A., & Tyagi, A. K. (2024). Artificial intelligence for cyber security: Current trends and future challenges. In *Automated secure computing for next-generation systems* (pp. 83–114). Wiley, Hoboken. https://doi.org/10.1002/9781394213948.ch5

[13] Malatji, M., & Tolah, A. (2024). Artificial intelligence (AI) cybersecurity dimensions: A comprehensive framework for understanding adversarial and offensive AI. *AI Ethics*. https://doi.org/10.1007/s43681-024-00427-4

[14] Choithani, T., Chowdhury, A., Patel, S., Patel, P., Patel, D., & Shah, M. (2024). A comprehensive study of artificial intelligence and cybersecurity on bitcoin, cryptocurrency, and banking system. *Annals of Data Science, 11*(1), 103–135. https://doi.org/10.1007/s40745-022-00433-5

[15] Sharma, N., & Jindal, N. (2024). Emerging artificial intelligence applications: Metaverse, IoT, cybersecurity, healthcare—an overview. *Multimedia Tools and Applications, 83*(19), 57317–57345. https://doi.org/10.1007/s11042-023-17890-6

[16] Rjoub, G., Bentahar, J., Wahab, O. A., Mizouni, R., Song, A., Cohen, R., Otrok, H., & Mourad, A. (2023). A survey on explainable artificial intelligence for cybersecurity. *IEEE Transactions on Network and Service Management, 20*(4), 5115–5140. https://doi.org/10.1109/TNSM.2023.3282740d

[17] Kumar, S., Gupta, U., & Singh, A. K. (2023). Artificial intelligence: Revolutionizing cyber security in the digital era. *Journal of Computational Mechanics and Management, 2*(3), 31–42. https://doi.org/10.57159/gadl.jcmm.2.3.23064

[18] Duary, S., Choudhury, P., Mishra, S., Sharma, V., Rao, D. D., & Aderemi, A. P. (2024, February). Cybersecurity threats detection in intelligent networks using predictive analytics approaches. In *2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICIPTM59628.2024.10563348

[19] Azib, A., Oubelaid, A., Ziane, D., Mohamed, N., Bajaj, M., Jurado, F., & Kamel, S. (2023). Reduced switch converter topology for double traction motors electric vehicles. In *2023 5th Global Power, Energy and Communication Conference (GPECOM)* (pp. 114–119). IEEE. https://doi.org/10.1109/GPECOM58364.2023.10175744

[20] Ngo, V. D., Vuong, T. C., Van Luong, T., & Tran, H. (2024). Machine learning-based intrusion detection: Feature selection versus feature extraction. *Cluster Computing, 27*(3), 2365–2379. https://doi.org/10.1007/s10586-023-04089-5

[21] Kulshrestha, P., & Vijay Kumar, T. V. (2024). Machine learning based intrusion detection system for IoMT. *International Journal of System Assurance Engineering and Management, 15*(5), 1802–1814. https://doi.org/10.1007/s13198-023-02119-4

[22] Siva Shankar, S., Hung, B. T., Chakrabarti, P., Chakrabarti, T., & Parasa, G. (2024). A novel optimization-based deep learning with artificial intelligence approach to detect intrusion attack in network system. *Education and Information Technologies, 29*(4), 3859–3883. https://doi.org/10.1007/s10639-023-11885-4

[23] Behiry, M. H., & Aly, M. (2024). Cyberattack detection in wireless sensor networks using a hybrid feature reduction technique with AI and machine learning methods. *Journal of Big Data, 11*(1), 16. https://doi.org/10.1186/s40537-023-00870-w

[24] Roopesh, M., Nishat, N., Rasetti, S., & Rahaman, M. A. (2024). A review of machine learning and feature selection techniques for cybersecurity attack detection with a focus on DDoS attacks. *Academic Journal of Science, Technology, Engineering & Mathematics Education, 4*(03), 178–194. https://doi.org/10.69593/ajsteme.v4i03.105

[25] Sulaiman, M., Waseem, M., Ali, A. N., Laouini, G., & Alshammari, F. S. (2024). Defense strategies for epidemic cyber security threats: Modeling and analysis using a machine learning approach. *IEEE*

Access, 12, 4958–4984. https://doi.org/10.1109/ACCESS.2024.3349660

[26] Elsayed, M. S., Le-Khac, N. A., Dev, S., & Jurcut, A. D. (2023). Distributed denial of service attack detection in network traffic using deep learning algorithm. *Sensors, 23*(20), 8642. https://doi.org/10.3390/s23208642

[27] Yang, L., et al. (2024). Attentive transformer deep learning algorithm for intrusion detection. https://doi.org/10.1371/journal.pone.0286652

[28] Isaza, G., Ramirez, F., Duque, N., Lopez, J. A., & Montes, J. (2023, June). DDoS attacks detection with deep learning model using a cloud architecture. In *Sustainable Smart Cities and Territories International Conference* (pp. 87–96). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-36957-5_8

[29] Akgun, D., Hizal, S., & Cavusoglu, U. (2022). A new DDoS attacks intrusion detection model based on deep learning for cybersecurity. *Computers & Security, 118*, 102748. https://doi.org/10.1016/j.cose.2022.102748

[30] Mashaly, M. (2024). Advanced hybrid transformer-CNN deep learning model for effective intrusion detection systems with class imbalance mitigation using resampling techniques. *Future Internet, 16*, 481. https://doi.org/10.3390/fi16120481