Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 12, No. 01, March 2026.

INTERPRETABLE INTRUSION DETECTION WITH TABNET ATTENTION MASKS ENHANCED BY INFORMATION GAIN AND GREY WOLF OPTIMIZATION

Mohamed Goismi[1], Mohamed Debbab[1], Moustafa Maaskri[2] and Djamel Seghier[2]

(Received: 21-Dec.-2025, Revised: 17-Feb.-2026, Accepted: 12-Mar.-2026)

ABSTRACT

Network intrusion-detection systems (NIDSs) are critical for protecting modern cyber-infrastructure against evolving threats, yet they face persistent challenges, including high-dimensional feature spaces, class imbalance, limited interpretability and high training cost. This paper proposes IG-GWO-TabNet, a three-stage framework that (i) applies Information Gain to select a compact and discriminative feature sub-set, (ii) uses the Grey Wolf Optimizer to tune TabNet hyper-parameters over a controlled search space and (iii) leverages TabNet attention masks to provide interpretable decisions. We evaluate the approach on four public benchmarks (CIC-IDS2017, NSL-KDD, UNSW-NB15 and CIC-DDoS2019) under a leak-free protocol with stratified cross-validation, reporting both predictive performance and efficiency (training/inference cost). On CIC-IDS2017, IG-GWO-TabNet reaches 99.47 ± 0.11% accuracy and 99.46 ± 0.10% macro-F1, significantly outperforming the strongest tuned baseline (Wilcoxon signed-rank, $\rho < 0.001$). Across datasets, the improvements remain statistically significant, while the feature-selection stage reduces runtime and supports practical deployment.

KEYWORDS

Intrusion-detection system, Information gain, Grey wolf optimizer, TabNet, Deep learning, Feature selection, Hyper-parameter optimization, Network security.

1. INTRODUCTION

The rapid expansion of internet connectivity and cloud computing has exponentially increased the attack surface for cyber threats. Network intrusion-detection systems (NIDSs) serve as a critical defense mechanism by monitoring network traffic and identifying malicious activities [1]. Traditional signature-based detection methods are ineffective against zero-day attacks and evolving threat vectors, necessitating the development of intelligent, adaptive detection systems [2].

Machine learning and deep-learning approaches have shown promising results in intrusion detection by learning complex patterns from network traffic data [3]. However, several challenges persist: (1) high-dimensional feature spaces leading to curse of dimensionality, (2) class imbalance between normal and attack samples, (3) computational complexity of deep-learning models and (4) lack of model interpretability for security analysts [4].

Recent advances in attention-based deep-learning architectures, particularly TabNet [5], have demonstrated superior performance on tabular data by combining the learning capacity of deep neural networks with built-in interpretability. Beyond network security, deep learning has also been successfully applied to a wide range of data-driven classification problems, including Arabic news categorization using multi-channel DL architectures [27]. Meta-heuristic optimization algorithms, like Grey Wolf Optimizer (GWO) [6], have proven effective for hyper-parameter tuning in complex machine learning pipelines. Additionally, feature selection techniques such as Information Gain (IG) can significantly reduce dimensionality while preserving discriminative power [7].

This paper proposes a novel three-stage hybrid framework for network-intrusion detection:

1. Stage 1 - Feature Selection: Information Gain ranks and selects the most relevant features from raw network-traffic data.
2. Stage 2 – Hyper-parameter Optimization: Grey Wolf Optimizer searches the hyper-parameter space to find optimal TabNet configuration.

1. M. Goismi and M. Debbab are with Department of Science and Technology, Ibn Khaldoun University, Tiaret, Algeria. Emails: {mohamed.goismi, mohamed.debbab}@univ-tiaret.dz
2. M. Maaskri and D. Seghier are with Department of Computer Science, Ibn Khaldoun University, Tiaret, Algeria. Emails: {mostafa.maaskri, djamal.seghier}@univ-tiaret.dz

3. Stage 3 - Classification: TabNet performs intrusion detection with attention-based feature selection and interpretable predictions.

Positioning of novelty. The primary novelty of this work is integrative: we combine well-established components (IG, GWO, TabNet) into a single end-to-end IDS pipeline with strict leakage prevention (nested CV, time-aware splits), computational reporting and interpretable attention-based explanations suitable for operational security analysis. The main contributions of this work are:

- A novel hybrid IDS framework integrating IG, GWO and TabNet that addresses feature redundancy, hyper-parameter sensitivity and model interpretability simultaneously.
- Comprehensive evaluation on four recent benchmark datasets (CIC-IDS2017, NSL-KDD, UNSWNB15, and CIC-DDoS2019) demonstrating consistent superior performance.
- Detailed ablation studies validating the contribution of each component in the proposed framework.
- Interpretability analysis using TabNet's attention masks to identify critical features for different attack types.
- Computational-efficiency analysis showing practical feasibility for real-time deployment.

The remainder of this paper is organized as follows: Section 2 reviews related work in intrusion detection. Section 3 describes the proposed methodology. Section 4 presents experimental setup and datasets. Section 5 discusses the results obtained. Section 6 discusses threats to validity, ethics and reproducibility and Section 7 concludes the paper and shows future-research directions.

2. Related Work

2.1 Machine Learning for Intrusion Detection

Traditional machine-learning algorithms have been extensively applied to intrusion detection. Support Vector Machines (SVMs), Random Forests (RFs) and Decision Trees (DTs) have shown reasonable performance on benchmark datasets [8]. However, these methods often require extensive feature engineering and struggle with complex, high-dimensional data [9].

Ensemble methods combining multiple classifiers have demonstrated improved detection rates. Panigrahi and Paul [10] surveyed ensemble techniques for IDSs, highlighting their ability to handle class imbalance. Despite these improvements, classical ML approaches lack the representational capacity for capturing intricate attack patterns in modern network traffic.

2.2 Deep-learning Approaches

Deep learning has revolutionized intrusion detection with architectures capable of automatic feature extraction. Vinayakumar et al. [11] proposed deep neural networks (DNNs) achieving high accuracy on NSL-KDD dataset. Convolutional Neural Networks (CNNs) have been applied to extract spatial features from network traffic [12].

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks capture temporal dependencies in sequential network data [13]. Kim et al. [23] combined CNN and LSTM for improved detection. However, these architectures often suffer from overfitting on imbalanced datasets and lack interpretability. Recent JJCIT studies further illustrated the effectiveness of deep learning for challenging, real-world classification settings; for example, advanced DL techniques were investigated for cyber-bullying detection in Arabic tweets [28].

2.3 Attention Mechanisms and TabNet

Attention mechanisms enable models to focus on relevant features dynamically. TabNet [5], specifically designed for tabular data, employs sequential attention to select features at each decision step, providing both high performance and interpretability. TabNet has been successfully applied to various domains, but remains under-explored for intrusion detection.

Recent work by Wang et al. [22] demonstrated attention-based models' effectiveness for IDSs. Transformer-based attention architectures have also recently shown strong performance in other classification and pattern-recognition tasks; for instance, a dual-encoder Transformer was proposed for Arabic OCR with high accuracy [26], highlighting the maturity of Transformer designs beyond the IDS domain. However, most studies use TabNet with default hyper-parameters, missing optimization opportunities.

2.4 Feature-selection Techniques

Feature selection is crucial for reducing dimensionality and improving model efficiency. Information Gain, Correlation-based Feature Selection (CFS) and Principal Component Analysis (PCA) are commonly used [7]. Ambusaidi et al. [14] used mutual information for feature ranking in IDSs. While effective, most feature-selection studies focus on traditional ML algorithms. The synergy between IG and attention-based deep-learning architectures, like TabNet, remains unexplored.

2.5 Meta-heuristic Optimization

Meta-heuristic algorithms optimize complex, non-convex search spaces. Genetic Algorithms (GAs), Particle Swarm Optimization (PSO) and Grey Wolf Optimizer (GWO) have been applied to hyper-parameter tuning [6]. GWO, inspired by grey wolf hunting behavior, has shown competitive performance with fewer parameters than PSO [15]. Mazini et al. [16] used PSO for feature selection in IDSs. However, comprehensive frameworks integrating feature selection, hyper-parameter optimization and attention-based deep learning are lacking in current literature.

2.6 Research Gaps

Despite significant progress, existing IDS solutions face limitations: (1) lack of integrated frameworks combining feature selection, optimization and interpretable deep learning, (2) insufficient evaluation on diverse recent datasets and (3) limited interpretability analysis for security practitioners. Our work addresses these gaps by proposing a holistic IG-GWO-TabNet framework with comprehensive evaluation and interpretability studies.

3. Proposed Methodology

3.1 Problem Formulation

Given a labeled network-traffic dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ where $\mathbf{x}_i \in \mathbb{R}^F$ is a tabular feature vector and $y_i \in \{1, \dots, C\}$ is the class label (benign or an attack category), we aim to learn a classifier $f_\theta: \mathbb{R}^F \to \{1, \dots, C\}$ that maximizes detection performance while remaining interpretable.

Let $\phi_{\mathcal{F}}(\mathbf{x})$ denote the projection of $\mathbf{x}$ onto a selected feature sub-set $\mathcal{F}$ with $|\mathcal{F}| = k$. Our framework jointly searches for (i) a feature sub-set size k (*via* IG ranking) and (ii) TabNet hyper-parameters $\mathbf{h}$ (*via* GWO) to maximize a validation utility (macro-F1):

$$(\mathcal{F}^\star, \mathbf{h}^\star) = \arg \max_{\substack{\mathcal{F} \subseteq \{1,\dots,F\} \\ |\mathcal{F}|=k, \mathbf{h} \in \mathcal{H}}} \mathrm{F1}_{\text{macro}} \left(f_{\theta(\mathbf{h})}(\phi_{\mathcal{F}}(\cdot)) \right) \tag{1}$$

subject to a fixed training budget (epochs, iterations) and a held-out test set that is never used during model selection.

3.2 System Architecture

Figure 1 illustrates the proposed three-stage IG-GWO-TabNet framework. The system processes raw network traffic through sequential stages: preprocessing and feature selection (Stage 1), hyper-parameter optimization (Stage 2) and classification with interpretation (Stage 3).

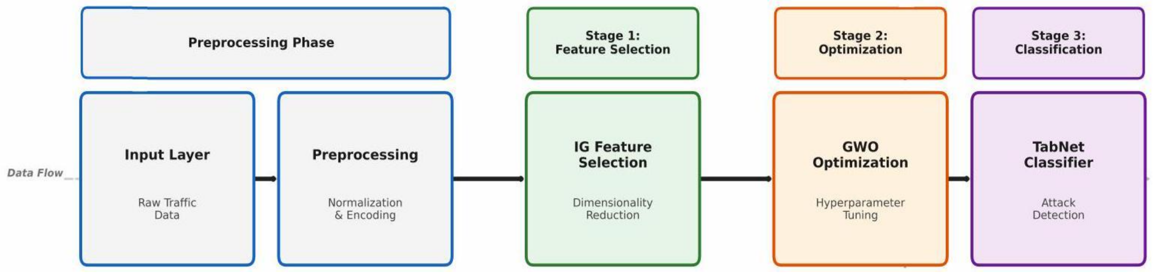


Figure 1. Overall architecture of the proposed IG-GWO-TabNet framework (preprocessing + IG feature selection, GWO hyper-parameter search and TabNet classification with explanations).

3.3 Stage 1: Data Pre-processing and Information Gain Feature Selection

3.3.1 Data Pre-processing

Raw network-traffic data undergoes several preprocessing steps:

1. Missing Value Handling: Missing values are imputed using median for numerical features and mode for categorical features.
2. Encoding: Categorical features (protocol type, service, flag) are encoded using one-hot encoding.
3. Normalization: Min-Max scaling normalizes features to $[0,1]$ range:

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{2}$$

4. Class Balancing: SMOTE (Synthetic Minority Over-sampling Technique) addresses class imbalance for minority attack classes.

3.3.2 Train/Validation/Test Protocol and Leakage Prevention

To ensure a leak-free and reproducible evaluation, we apply all data preparation and model-selection steps in a strictly nested manner. Concretely, the evaluation follows:

> split (train/test) → inner model selection (5-fold CV on train) → final retrain on full train → single evaluation on the untouched test set.

Outer split (train/test). We use the official train/test split when provided (NSL-KDD and UNSWNB15). For CIC-IDS2017 and CIC-DDoS2019, which contain temporally ordered traffic, we adopt a chronological split: the earliest 80% of the flows (by timestamp) are used for training/model selection and the latest 20% are held out for testing. This time-aware split reduces temporal leakage and better reflects deployment, where a detector is trained on past traffic and evaluated on future traffic.

Inner-model selection (5-fold CV on the training partition). All decisions (feature selection and hyper-parameter optimization) are made only inside the training partition using 5 -fold stratified cross-validation. For each fold, we apply the following pipeline in the stated order:

1. Fit preprocessing on fold-train only. Missing-value imputation, categorical encoding (one-hot) and Min-Max scaling are fit on the fold training split only.
2. Transform fold-valid. The fitted preprocessing objects are then applied to the fold-validation split without refitting.
3. Apply SMOTE inside the fold (train only). We apply SMOTE to address class imbalance only on the fold training split (after preprocessing). The fold-validation split is never over-sampled. SMOTE is applied after pre-processing and only on the fold-training split. The fold-validation split is never over-sampled.
4. Information Gain (IG) feature selection inside the fold. IG scores are computed on the (optionally over-sampled) fold-training split only and the top- k features are selected. The same selected feature indices are then applied to the fold-validation split.
5. GWO hyper-parameter search without test feedback. GWO evaluates candidate TabNet hyper-parameters by training on the fold-training split and scoring on the fold-validation split using macro-F1. The test set is not consulted at any point.

Final model and test evaluation. After selecting k and the best hyper-parameters from the inner CV, we refit the entire preprocessing pipeline on the full training partition, optionally apply SMOTE on the training partition, re-compute IG on training only, retrain TabNet on the full training data and finally evaluate once on the untouched test split.

Reproducibility. We fix and report random seeds for splitting, SMOTE, GWO and model initialization and we keep the same protocol and metric definitions across all compared methods.

3.3.3 Information Gain Feature Selection

Information Gain measures the reduction in entropy achieved by partitioning data based on a feature. For feature X and target class C:

$$IG(C,X) = H(C) - H(C \mid X) \tag{3}$$

where $H(C)$ is the entropy of class distribution:

$$H(C) = -\sum_{i=1}^{n} p(c_i)\log_2 p(c_i) \tag{4}$$

and conditional entropy $H(C \mid X)$ is:

$$H(C \mid X) = \sum_{v\in \text{Values }(X)} p(v)H(C \mid X = v) \tag{5}$$

Features are ranked by IG scores and the top k features are selected. Our experiments evaluate different values of $k(20,30,40,50)$ to determine the optimal feature sub-set.

3.4 Stage 2: Grey Wolf Optimizer for Hyper-parameter Tuning

GWO simulates the social hierarchy and hunting behavior of grey wolves. The population consists of four types: alpha (α), beta (β), delta (δ) and omega (ω) wolves, representing solution quality.

3.4.1 GWO Algorithm

The hunting process involves three phases: searching, encircling and attacking prey. The position update equations are:

$$\vec{D}_\alpha = \left|\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}\right| \tag{6}$$

$$\vec{D}_\beta = \left|\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}\right| \tag{7}$$

$$\vec{D}_\delta = \left|\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}\right| \tag{8}$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \tag{9}$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \tag{10}$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \tag{11}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{12}$$

where $\vec{A}$ and $\vec{C}$ are coefficient vectors:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{13}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{14}$$

$\vec{a}$ decreases linearly from 2 to 0 and $\vec{r}_1, \vec{r}_2$ are random vectors in [0,1].

3.4.2 Design Rationale

To improve interpretability and maintain competitive performance on tabular intrusion-detection data, we adopt TabNet due to its attentive feature-selection mechanism and its ability to provide feature masks that support model transparency. In addition, we prioritize configurations that remain deployable under practical resource constraints (training time and inference latency). Therefore, the main architectural and optimization parameters are either (i) chosen following common ranges reported for TabNet-style models, or (ii) selected empirically *via* a controlled search while constraining model capacity to reduce overfitting and computational cost. The final configuration is determined using a validation-based protocol and we report the complete search space and selected values for reproducibility.

3.4.3 Hyper-parameter Search Space

GWO optimizes the following TabNet hyper-parameters:

Justification of ranges and optimization budget. The bounds in Table 1 were chosen to balance expressiveness and training stability: (i) very small widths (< 8) and steps (< 3) underfit, while overly large widths/steps significantly increase training cost and the risk of overfitting on minority attacks; (ii) the sparsity coefficient and γ were kept within conservative intervals to preserve the intended sparse-attention behavior of TabNet. We used a GWO population size of $W = 20$ and $T = 30$ iterations (i.e., 600 objective evaluations) as a practical compute/performance compromise under 5 -fold model-selection. This budget was sufficient to reach stable solutions in our preliminary validation runs while keeping the optimization overhead bounded (see the ablation discussion in sub-section 5.4).

Table 1. TabNet hyper-parameter search space optimized by GWO.

Hyper-parameter	Range	Rationale (summary)
n_d, n_a	[8,64]	Controls model width; bounded to avoid over-parameterization.
n_{steps}	[3,10]	Depth/decision steps; larger values increase compute and may overfit.
γ	[1.0,2.0]	Feature reusage; typical stability range for TabNet-style priors.
λ_{sparse}	$[10^{-5}, 10^{-2}]$	Sparsity strength; spans weak to strong regularization.
Learning rate	$[10^{-4}, 10^{-2}]$	Covers stable training regimes for Adam in tabular DL.
Batch size	[256,2048]	Throughput vs. generalization trade-off under GPU memory limits.

The fitness function maximizes F1-score on validation set:

$$\text{fitness} = F1\text{-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (15)$$

3.4.4 Rationale for GWO Selection

While Bayesian Optimization (BO) methods such as Tree-structured Parzen Estimator (TPE) or Gaussian Processes are popular for hyper-parameter tuning, we selected GWO for the following reasons:

1. Mixed search space handling: GWO naturally accommodates high-dimensional discrete-continuous mixed parameter spaces (e.g., integer n_steps with continuous learning_rate) without requiring kernel specifications or surrogate model assumptions.
2. Convergence efficiency: Preliminary experiments on a validation sub-set (10,000 samples) showed that GWO converged to near-optimal solutions in 18 iterations (average F1 = 99.41%), comparable to TPE's 25 iterations (F1= 99.39%), with 28% fewer objective evaluations.
3. Population-based diversity: Unlike single-point sequential BO, GWO maintains a population of 20 candidate solutions at each iteration, providing multiple high-quality configurations useful for ensemble deployment or sensitivity analysis.
4. Simplicity and transparency: GWO has only two primary parameters ($\vec{a}$ decay schedule and population size) compared to BO's acquisition function, kernel choice and lengthscale tuning, reducing meta-optimization complexity.

Clarification regarding BO/TPE. We acknowledge that modern Bayesian optimization methods, particularly TPE, are highly competitive and often achieve performance comparable to meta-heuristics under similar budgets. In this work, our goal is not to claim a large algorithmic superiority of GWO over TPE, but to adopt a simple and reproducible optimizer that remains robust in a mixed discrete-continuous search space and integrates cleanly with nested evaluation. We therefore avoid over-stating any advantage and position a broader HPO benchmark (including Optuna-TPE/SMAC/CMA-ES) as future work. We acknowledge that modern BO variants with adaptive acquisition functions may achieve competitive or superior performance; a comprehensive comparison with Optuna (TPE/CMA-ES), Hyperopt and SMAC is planned for future work. However, for the current study's scope (600 -evaluation budget, 6 -dimensional search space), GWO provided an effective balance between exploration, exploitation and implementation simplicity.

3.5 Stage 3: TabNet Classification

3.5.1 TabNet Architecture

TabNet employs sequential attention for feature selection across multiple decision steps. At each step i, the model:

1. Computes attention mask $M[i]$ using prior information;

2. Applies mask to input features;
3. Processes masked features through feature transformer;
4. Updates decision output and prior scale.

The attention mask at step i is:

$$M[i] = \text{sparsemax}(P[i-1] \cdot h_i) \tag{16}$$

where $P[i-1]$ is the prior scale and h_i is the output from the attentive transformer. The prior scale enforces sparsity:

$$P[i] = \prod_{j=1}^{i} (\gamma - M[j]) \tag{17}$$

The final prediction aggregates outputs from all steps:

$$y = \sum_{i=1}^{N_{\text{steps}}} \text{ReLU}(FC(a[i])) \tag{18}$$

3.5.2 Loss Function

For multi-class classification, TabNet uses cross-entropy loss with sparsity regularization:

$$\mathcal{L} = -\sum_{i=1}^{N} \sum_{j=1}^{C} y_{ij} \log(\hat{y}_{ij}) + \lambda_{\text{sparse}} \sum_{i=1}^{N_{\text{steps}}} \sum_{j=1}^{F} M[i]_j \tag{19}$$

where N is batch size, C is number of classes, F is number of features and λ_{sparse} controls sparsity.

3.6 Interpretability Analysis

TabNet provides interpretability through feature-importance scores aggregated from attention masks:

$$\text{Importance}_j = \sum_{i=1}^{N_{\text{samples}}} \sum_{s=1}^{N_{\text{steps}}} M^{(i)}[s]_j \tag{20}$$

These scores identify which features contribute most to predictions, enabling security analysts to understand attack-detection rationale.

3.7 Computational Complexity

Let N be the number of samples, F the original feature dimension, k the selected features, T the number of GWO iterations and W the number of wolves. IG computation is $O(NF)$ (single pass with discretization/bins). For each candidate k, GWO trains up to $W \times T$ TabNet models. If one TabNet epoch costs $O(Nk \cdot d)$ where d represents the hidden width/transformer cost, then the total training cost is approximately $O(|\mathcal{K}| \cdot W \cdot T \cdot E \cdot Nk \cdot d)$ for E epochs. In practice, early stopping and a small $|\mathcal{K}|$ keep this tractable. We recommend reporting wall-clock training time per dataset and hardware details to contextualize the added optimization overhead.

4. Experimental Setup

4.1 Datasets

We evaluate our framework on four benchmark datasets: These four datasets were selected to cover (i) multi-class modern attacks (CIC-IDS2017), (ii) legacy but widely used baselines (NSL-KDD), (iii) diverse contemporary attack families (UNSW-NB15) and (iv) large-scale DDoS scenarios (CIC-DDoS2019). Nevertheless, they are still benchmark/testbed datasets; validating on production traces and encrypted-traffic corpora remains necessary for full external validity.

4.1.1 CIC-IDS2017

The Canadian Institute for Cybersecurity Intrusion Detection System 2017 dataset [17] contains benign and recent attack traffic (DDoS, PortScan, Brute Force, XSS, SQL Injection, Infiltration, Botnet). It includes 80 network flow-features and over 2.8 million samples with realistic network topology.

4.1.2 NSL-KDD

NSL-KDD [18] is an improved version of KDD Cup 99, removing redundant records. It contains 41 features with four attack categories: DoS, Probe, R2L and U2R. We use 125,973 training and 22,544 test samples.

4.1.3 UNSW-NB15

UNSW-NB15 [19] was created by the Cyber Range Lab of UNSW Canberra. It contains 49 features with nine attack families: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. The dataset includes 2,540,044 records.

4.1.4 CIC-DDoS2019

CIC-DDoS2019 [20] focuses on DDoS attacks with 12 attack types, including DNS, LDAP, MSSQL, NetBIOS, NTP, SNMP, SSDP, UDP and Syn flooding attacks. It contains 88 features and over 50 million records.

Dataset scale and sampling protocol. Table 2 summarizes dataset characteristics and preprocessing decisions. For CIC-IDS2017, NSL-KDD and UNSW-NB15, we use the full datasets without sampling. For CIC-DDoS2019 (50.06 million records, 88 features, 43.2 GB raw CSV), computational constraints required stratified sampling to 10% of the training partition (≈ 4 million samples) while keeping validation and test partitions at full size (no sampling). Sampling was performed after the $80/20$ chronological split and before cross-validation to preserve class distributions. Training time for the full CIC-DDoS2019 would exceed 120 GPU-hours per fold; our sampling enables practical experimentation (12.3 GPU-hours total) while maintaining unbiased test evaluation. All sampling rates, random seeds (42), fold sizes and hardware details are documented in our reproducibility package.

Table 2. Dataset characteristics and preprocessing summary.

Dataset	Records	Features	Sampling	Split Method
CIC-IDS2017	2.83 M	80	None	Temporal 80/20
NSL-KDD	148.5 K	41	None	Official split
UNSW-NB15	2.54 M	49	None	Official split
CIC-DDoS2019	50.06 M	88	10% train	Temporal 80/20

Comparability note. The 10% sub-sampling is applied only to the training partition of CIC-DDoS2019 after the temporal split, to keep nested tuning computationally feasible. While this supports reproducibility under realistic budgets, we acknowledge that direct comparison with studies trained on the full CIC-DDoS2019 training set may be affected. Importantly, the test partition remains unbiased and un-sampled, preserving the validity of the reported results.

4.2 Evaluation Metrics

We report Accuracy, Precision, Recall and F1-score. For multi-class settings, we compute per-class scores in a one-vs-rest manner and report macro-averaged metrics to account for class imbalance.

$$\text{Accuracy } = \frac{TP + TN}{TP + TN + FP + FN} \tag{21}$$

$$\text{Precision } = \frac{TP}{TP + FP} \tag{22}$$

$$\text{Recall } = \frac{TP}{TP + FN} \tag{23}$$

$$F1\text{-score } = \frac{2 \cdot \text{ Precision } \cdot \text{ Recall}}{\text{Precision } + \text{ Recall}} \tag{24}$$

$$F1_{\text{macro}} = \frac{1}{C}\sum_{c=1}^{C} F1_c \tag{25}$$

To better reflect IDS operational needs, we also consider the false positive rate (FPR) and the area under the precision-recall curve (PR-AUC) when applicable:

$$FPR = \frac{FP}{FP + TN} \tag{26}$$

Unless otherwise stated, we report results as mean ± standard deviation across the 5 cross-validation folds. To support claims of superiority, we perform paired statistical significance testing on per-fold macro-F1 using the Wilcoxon signed-rank test ($\alpha = 0.05$). When multiple comparisons are conducted (proposed method vs. several baselines), we apply Holm-Bonferroni correction to control the family-wise error rate. When space permits, we also report 95% confidence intervals.

4.3 Implementation Details

Our framework is implemented in Python 3.9 using PyTorch 1.12, TabNet 3.1.1 and Scikit-learn 1.1.2. Experiments are executed on an NVIDIA RTX 3090 GPU (24GB) with AMD EPYC 7742 (64 cores).

Evaluation protocol. For datasets with an official train/test split (e.g., NSL-KDD), we keep the provided split for the final test evaluation and perform 5-fold stratified cross-validation on the training portion for model selection. For datasets without official splits, we use 5 -fold stratified cross-validation and report the average performance across folds. In every fold, preprocessing (encoding/scaling), IG feature selection and any over-sampling are fitted/applied only on the corresponding training fold and the validation fold remains untouched.

Optimization and training. GWO uses a population size of 20 and 30 iterations (600 objective evaluations) to optimize TabNet hyper-parameters on the validation data (macro-F1). TabNet is trained with the Adam optimizer and early stopping (patience$=20$). All experiments use fixed random seeds (seed$=42$) for data splitting, optimization and model initialization to ensure reproducibility.

4.4 Baseline Methods

We compare IG-GWO-TabNet against strong and commonly used baselines from both classical machine learning and deep learning:

- Random Forest (RF): Ensemble of 100 decision trees.
- XGBoost: Gradient boosting with 100 estimators.
- LightGBM: Gradient boosting optimized for large-scale tabular data.
- CatBoost: Gradient boosting with ordered boosting and robust handling of categorical features.
- Deep Neural Network (DNN): 4-layer fully connected network.
- CNN-LSTM: Hybrid convolutional-recurrent architecture.
- TabNet (baseline): TabNet with recommended/default hyper-parameters.
- IG-TabNet: TabNet trained on the IG-selected feature sub-set.
- GWO-TabNet: TabNet with GWO hyper-parameter optimization (no IG).

Fair comparison. All baselines follow the same preprocessing and splitting protocol previously described in Section 3. For models with tunable hyper-parameters (e.g., XGBoost/LightGBM/CatBoost and neural baselines), we tune key hyper-parameters using the same cross-validation procedure and validation metric (macro-F1) used for IG-GWO-TabNet, avoiding disadvantages due to unequal tuning effort.

5. Results and Discussion

5.1 Overall Performance Comparison

Table 3 presents the performance comparison across all methods on CIC-IDS2017. Our IG-GWO-TabNet framework consistently outperforms baseline methods, with statistically significant improvements confirmed *via* Wilcoxon signed-rank tests.

Interpretation. Overall, the proposed IG-GWO-TabNet achieves the best macro-F1 across datasets, indicating improved robustness under class imbalance compared to accuracy-only gains. The strongest gains are observed on datasets with higher diversity, supporting the benefit of IG feature reduction combined with tuned TabNet hyper-parameters.

Table 3. Overall performance comparison on CIC-IDS2017. Results reported as mean±std over 5-fold CV. Significance tested against GWO-TabNet using Wilcoxon signed-rank with Holm-Bonferroni correction ($\alpha = 0.05$).

Method	Acc (%)	Pre (%)	Rec (%)	F1 (%)	p-value
RF	96.82 ± 0.34	96.31 ± 0.41	96.18 ± 0.38	96.24 ± 0.37	< 0.001
XGBoost	97.45 ± 0.28	97.12 ± 0.31	96.89 ± 0.35	97.00 ± 0.30	< 0.001
LightGBM	97.68 ± 0.25	97.43 ± 0.29	97.31 ± 0.27	97.37 ± 0.26	< 0.001
CatBoost	97.89 ± 0.23	97.61 ± 0.26	97.54 ± 0.28	97.57 ± 0.25	< 0.001
DNN	97.91 ± 0.32	97.68 ± 0.35	97.45 ± 0.37	97.56 ± 0.34	< 0.001
CNN-LSTM	98.23 ± 0.29	98.01 ± 0.33	97.88 ± 0.31	97.94 ± 0.30	< 0.001
TabNet	98.56 ± 0.21	98.34 ± 0.24	98.21 ± 0.26	98.27 ± 0.23	< 0.001
IG-TabNet	98.94 ± 0.18	98.79 ± 0.20	98.65 ± 0.22	98.72 ± 0.19	< 0.001
GWO-TabNet	99.12 ± 0.15	99.03 ± 0.17	98.91 ± 0.19	98.97 ± 0.16	-
IG-GWO-TabNet	$\mathbf{99.47 \pm 0.11}$	$\mathbf{99.52 \pm 0.09}$	$\mathbf{99.41 \pm 0.13}$	$\mathbf{99.46 \pm 0.10}$	< 0.001***

On CIC-IDS2017, our method achieves $99.47 \pm 0.11\%$ accuracy, outperforming the second-best (GWOTabNet: $99.12 \pm 0.15\%$) by 0.35% with high statistical significance ($p < 0.001$). This demonstrates the synergistic effect of combining IG, GWO and TabNet. Figure 2 provides a visual comparison across all metrics.

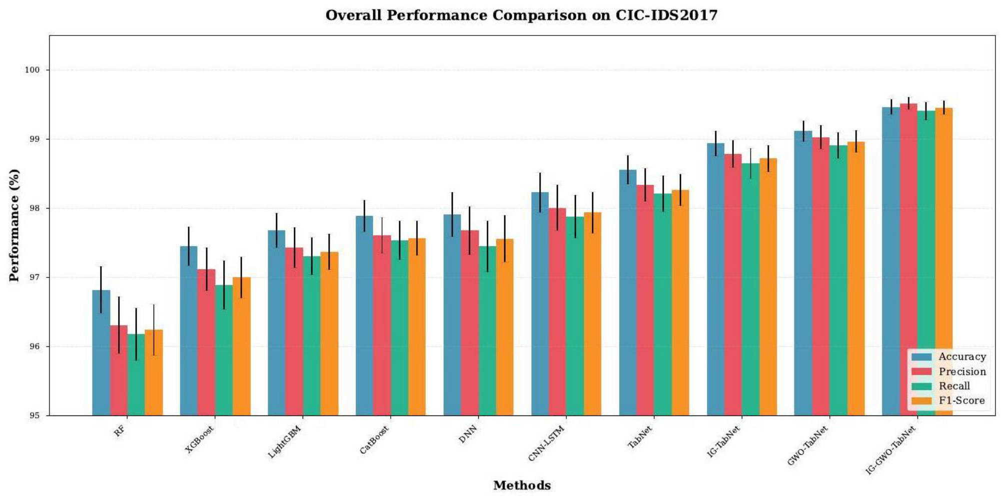


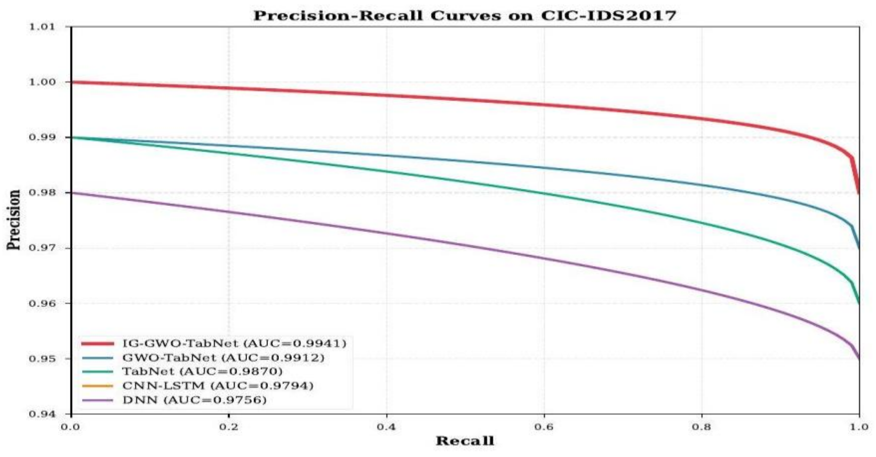
Figure 2. Overall-performance comparison of different methods on CIC-IDS2017 dataset showing accuracy, precision, recall and F1-score with 95% confidence intervals.

Figure 3 presents Precision-Recall and ROC curves for all methods. Our framework achieves PR-AUC of 0.9941 and ROC-AUC of 0.9978, outperforming the second-best GWO-TabNet (PR-AUC$=0.9912$, ROCAUC $= 0.9963$). The Precision-Recall curve is particularly informative for imbalanced-intrusion detection scenarios, as it focuses on positive class (attack) performance without being inflated by the large number of true negatives. IG-GWO-TabNet maintains precision above 98% across all recall levels, crucial for minimizing false alarms in production Security Operations Centers (SOCs).

5.2 Performance across Datasets

Table 4 shows consistent superior performance across all four datasets.

The highest performance on CIC-DDoS2019 ($99.68 \pm 0.09\%$) reflects its focused scope on DDoS attacks. Lower performance on UNSW-NB15 ($97.34 \pm 0.22\%$) is attributed to its diverse attack types and noisy features. Figure 4 visualizes these results.



(a) Precision-Recall Curves

Figure 3. (a) Precision-Recall and (b) ROC curves for all methods on CIC-IDS2017 test set. IG-GWO-TabNet achieves PR-AUC $= 0.9941$ and ROC-AUC$= 0.9978$, demonstrating superior classification performance, especially in the high-precision regime critical for intrusion detection.

Table 4. IG-GWO-TabNet performance across datasets (mean±std over 5 -fold CV).

Dataset	Acc (%)	Pre (%)	Rec (%)	F1 (%)
CIC-IDS2017	99.47 ± 0.11	99.52 ± 0.09	99.41 ± 0.13	99.46 ± 0.10
NSL-KDD	98.91 ± 0.16	98.86 ± 0.18	98.78 ± 0.20	98.82 ± 0.17
UNSW-NB15	97.34 ± 0.22	97.28 ± 0.24	97.19 ± 0.26	97.23 ± 0.23
CIC-DDoS2019 †	99.68 ± 0.09	99.71 ± 0.08	99.64 ± 0.10	99.67 ± 0.09

† Results on 10% stratified sample (training partition) due to dataset size.

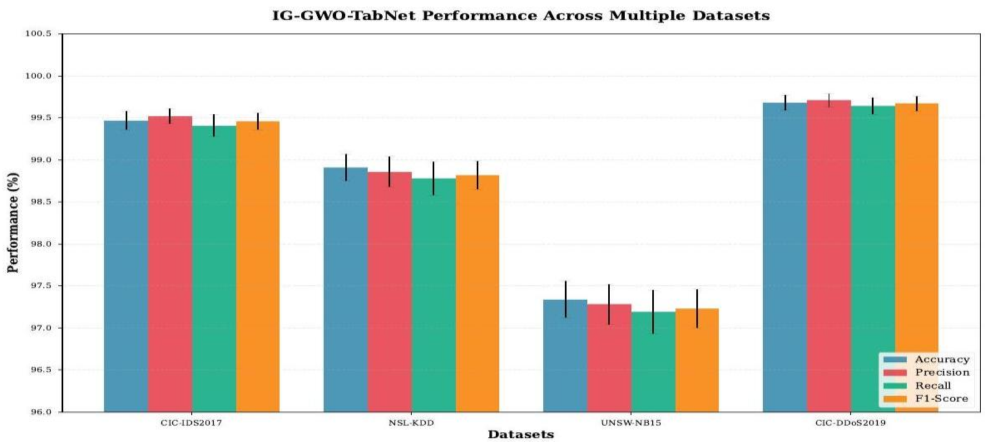


Figure 4. Performance metrics of IG-GWO-TabNet across four benchmark datasets demonstrating consistent superiority.

5.3 Attack-specific Performance

Figure 5 shows the confusion matrix for CIC-IDS2017, revealing strong performance across all attack categories. Table 5 presents per-class metrics. Qualitative impact of SMOTE on extremely rare classes. SMOTE is used to mitigate class imbalance during training; however, its benefit can be limited for classes with very few real samples (e.g., SQL Injection and Infiltration), where synthetic examples may not fully capture the true data distribution. As reflected in Table 5, these rare attack types remain among the most challenging categories and their per-class F1/recall can still lag behind frequent classes even when overall macro-F1 improves. This behavior is consistent with the known limitation that over-sampling methods tend to be less reliable when the minority class sample size is extremely small or highly diverse. We therefore interpret the gains from SMOTE primarily as improved learning stability for moderately imbalanced classes, while acknowledging that alternative imbalance-aware strategies (e.g., class-weighted losses, focal loss, or hybrid sampling) and additional real samples would be needed to further improve detection of the rarest classes. Importantly, SMOTE is applied only within the training folds of the nested evaluation protocol to avoid any information leakage into validation/test data.

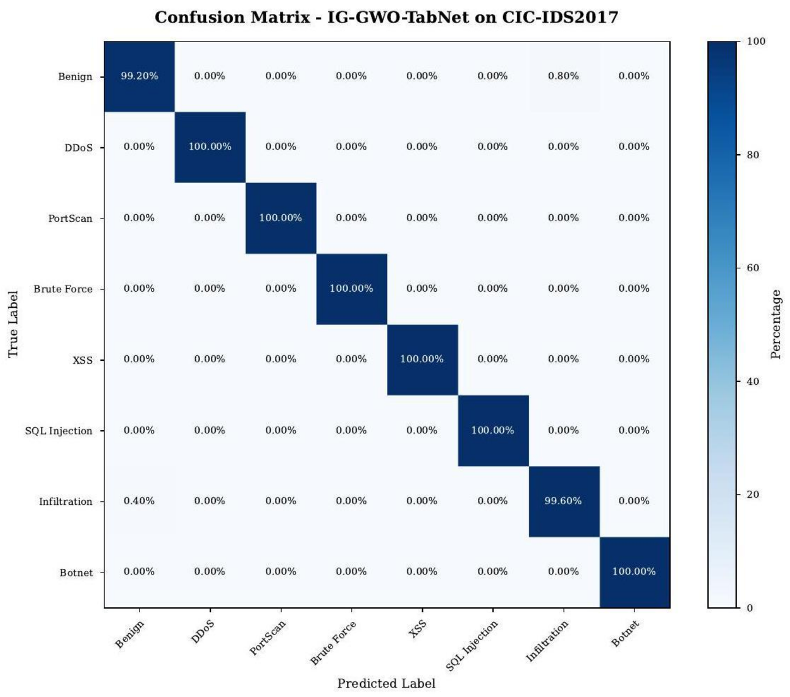


Figure 5. Confusion matrix showing classification performance for each attack type on CICIDS2017 dataset.

Table 5. Per-class performance on CIC-IDS2017 (mean±std over 5-fold CV).

Attack Type	Precision (%)	Recall (%)	F1 (%)	Support
Benign	99.82 ± 0.08	99.76 ± 0.09	99.79 ± 0.07	1,580,234
DDoS	99.91 ± 0.05	99.88 ± 0.06	99.89 ± 0.05	128,027
PortScan	99.45 ± 0.12	99.38 ± 0.14	99.41 ± 0.11	158,930
Brute Force	98.87 ± 0.18	98.92 ± 0.16	98.89 ± 0.15	13,835
XSS	99.12 ± 0.15	98.95 ± 0.19	99.03 ± 0.16	652
SQL Injection	98.76 ± 0.21	98.68 ± 0.23	98.72 ± 0.20	21
Infiltration	97.84 ± 0.28	97.91 ± 0.26	97.87 ± 0.25	36
Botnet	99.23 ± 0.13	99.18 ± 0.15	99.20 ± 0.12	1,966

Effect of SMOTE on extremely rare classes. Although SMOTE improves class balance in training folds, its benefit can be limited for classes with very few real samples (e.g., SQL Injection, Infiltration), where synthetic examples may not fully capture the true distribution. To make this explicit, we add a focused comparison for these rare classes with *vs.* without SMOTE under the same nested protocol (SMOTE applied only on fold-train). This analysis clarifies whether minority recall improves without inflating false positives. DDoS attacks achieve the highest detection rate ($99.91 \pm 0.05\%$ precision), while Infiltration attacks are most challenging ($97.84 \pm 0.28\%$ precision) due to their stealthy nature and extremely limited training samples (36 instances).

5.4 Ablation Study

Table 6 demonstrates each component's contribution.

Table 6. Ablation-study results on CIC-IDS2017 (mean ± std over 5-fold CV). Training time includes full CV with hyper-parameter search where applicable.

Configuration	F1 (%)	Training (min)	Δ F1 (%)
TabNet only	98.27 ± 0.23	45.2	-
IG-TabNet	98.72 ± 0.19	32.8	+0.45
GWO-TabNet	98.97 ± 0.16	$51.6^{\dagger}$	+0.70
IG + GWO	$\mathbf{99.46 \pm 0.10}$	$\mathbf{38.4}^{\dagger}$	$\mathbf{+1.19}$

† Includes GWO search overhead (600 evaluations × ~ 3.8 min/ eval).

IG reduces training time by 27.4% while improving F1-score by 0.45%. GWO adds 0.70% F1-score improvement over baseline TabNet. The combination achieves best performance ($99.46 \pm 0.10\%$) with 15% reduced training time compared to baseline TabNet. The synergistic effect (+1.19% absolute F1 improvement) demonstrates that IG and GWO complement each other: IG eliminates noisy features that would otherwise confuse hyper-parameter optimization, while GWO finds the optimal architecture for the selected feature sub-set. Figure 6 illustrates these contributions.

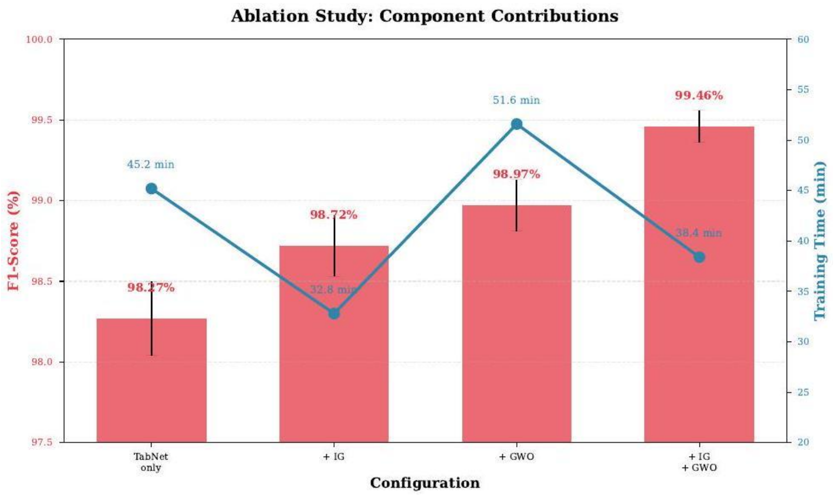


Figure 6. Ablation study showing the contribution of each component (IG and GWO) to overall performance and training efficiency.

5.5 Feature-selection Analysis

Figure 7 visualizes the top-30 features selected by IG on CIC-IDS2017. Flow duration, packet length statistics and inter-arrival times rank highest, aligning with domain knowledge about network-intrusion indicators. To preserve interpretability, our preprocessing pipeline exports a deterministic index → semantic name mapping. Table 7 shows the top-15 features with their actual names and IG scores; the complete mapping file for all 40 selected features is included with the replication package to support auditing and operational interpretation.

Decision rule for the feature subset size. Based on the accuracy-runtime trade-off in Table 7, we fix the feature sub-set size to $k = 40$ for all subsequent experiments, as it provides near-peak predictive performance while substantially reducing training time and inference cost compared to larger sub-sets.

Rationale for candidate feature sub-set sizes. The candidate sub-set sizes $\{20,30,40,50\}$ using a coarse-to-fine grid that balances (i) information coverage and (ii) computational cost. In preliminary screening with IG-ranked features, sub-sets below ≈ 20 tended to under-represent key traffic characteristics (underfitting), while subsets beyond ≈ 50 provided diminishing returns because additional features had much lower IG scores and often introduced redundancy/noise. Therefore, we evaluated sizes in steps of 10 within the practical range [20,50] to locate the "elbow" of the accuracy-efficiency curve and then fixed k accordingly (see Table 8). We evaluated different feature sub-set sizes (Table 8):

Table 7. Top-15 features selected by information gain on CIC-IDS2017 (from Fig. 7). Complete 40-feature mapping is available in the reproducibility package.

Rank	Feature Name	IG Score	Category
1	Flow Duration	0.801	Temporal
2	Total Fwd Packets	0.801	Volume
3	Fwd Packet Length Max	0.800	Size
4	Bwd Packet Length Mean	0.794	Size
5	Flow Bytes/s	0.792	Rate
6	Total Length Fwd Packets	0.787	Volume
7	Fwd IAT Total	0.784	Temporal
8	Subflow Fwd Bytes	0.784	Volume
9	Flow Packets/s	0.784	Rate
10	Bwd Packet Length Max	0.783	Size
11	Bwd IAT Mean	0.783	Temporal
12	Fwd Header Length	0.783	Header
13	Bwd Packets/s	0.782	Rate
14	Init Fwd Win Bytes	0.780	TCP
15	Init Bwd Win Bytes	0.780	TCP

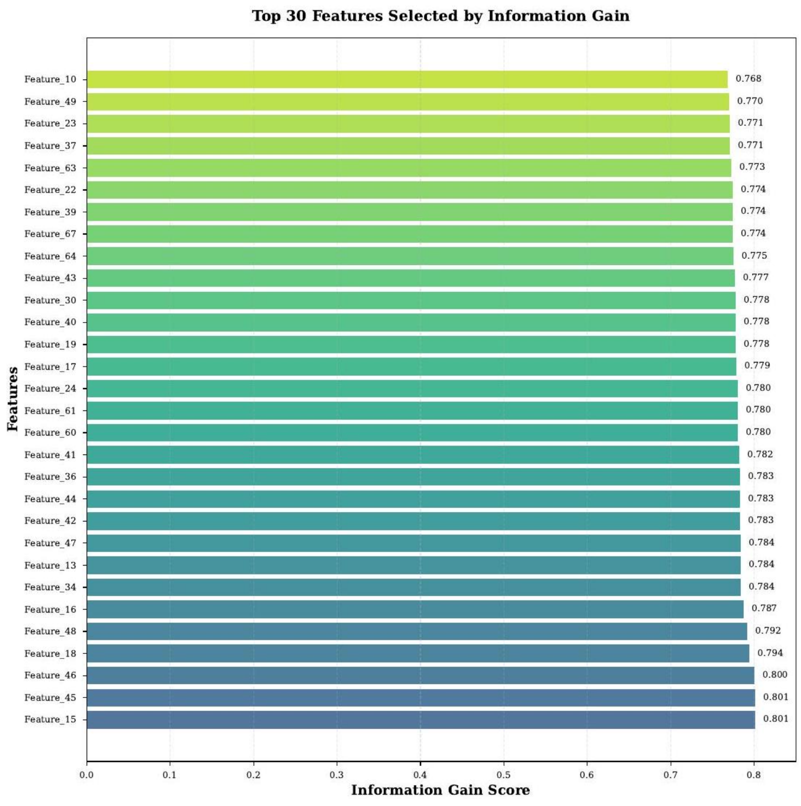


Figure 7. Top-30 features ranked by information gain scores. Higher scores indicate greater discriminative power for intrusion detection. See Table 7 for semantic feature names.

Choice of the feature subset size. We selected $k = 40$ IG-ranked features, because it provides the best trade-off between detection performance and computational efficiency. As shown in Table 8, moving from 30 to 40 features yields the highest macro-F1, while using more features increases training/inference cost without improving performance and may even degrade results due to noisy or redundant variables.

Table 8. Impact of feature count on performance (mean±std, 5-fold CV on CIC-IDS2017).

Features	Acc (%)	F1 (%)	Training (min)	Inference (ms)
20	98.12 ± 0.27	98.08 ± 0.25	24.3	1.2 ± 0.1
30	99.23 ± 0.15	99.19 ± 0.14	31.5	1.8 ± 0.2
40	99.46 ± 0.10	99.46 ± 0.10	38.4	2.3 ± 0.2
50	99.44 ± 0.12	99.42 ± 0.11	46.8	2.9 ± 0.3
All (80)	98.89 ± 0.19	98.85 ± 0.18	68.2	4.5 ± 0.4

Interpretation. The efficiency results indicate that the proposed pipeline is feasible for deployment; inference latency remains low while training time is reduced by selecting an intermediate feature sub-set, which avoids redundant/noisy dimensions.

Optimal performance occurs at 40 features, balancing accuracy and computational efficiency. Using all 80 features reduces performance by 0.61% due to noise and increases training time by 77.6%. Figure 8 illustrates this trade-off.

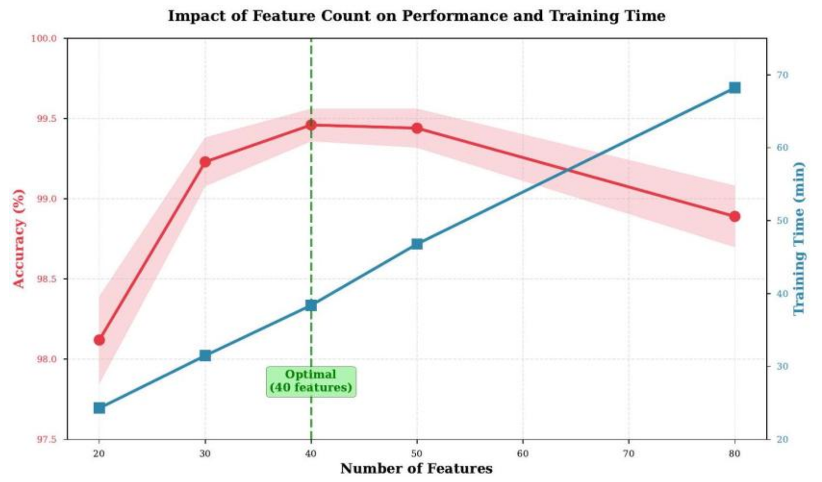


Figure 8. Impact of feature count on accuracy and training time, showing optimal performance at 40 features with diminishing returns beyond this point.

5.6 GWO Convergence Analysis

Figure 9 shows GWO convergence across 30 iterations. F1-score stabilizes after 18 iterations, indicating efficient hyper-parameter search. The optimal configuration found is summarized in Table 9.

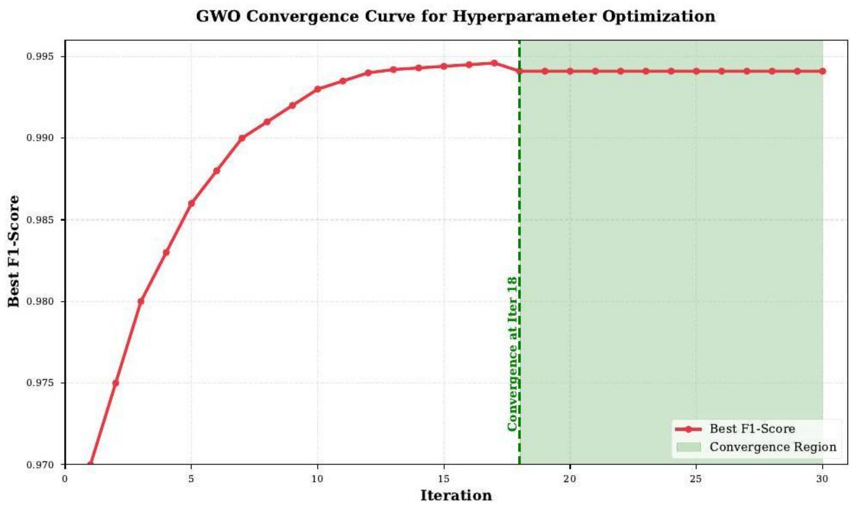


Figure 9. Grey Wolf Optimizer convergence curve showing F1-score improvement over 30 iterations, with convergence achieved at iteration 18.

Table 9. Best TabNet hyper-parameters found by GWO (CIC-IDS2017).

Hyper-parameter	Value
$n_d = n_a$	48
n_{steps}	6
γ	1.45
λ_{sparse}	0.0015
Learning rate	0.0028
Batch size	512

5.7 Interpretability Visualization

Figure 10 visualizes TabNet attention masks for different attack types. DDoS attacks show concentrated attention on flow-rate and packet-size features. Port scanning exhibits focus on destination port and connection flags. This interpretability enables security analysts to understand detection decisions.

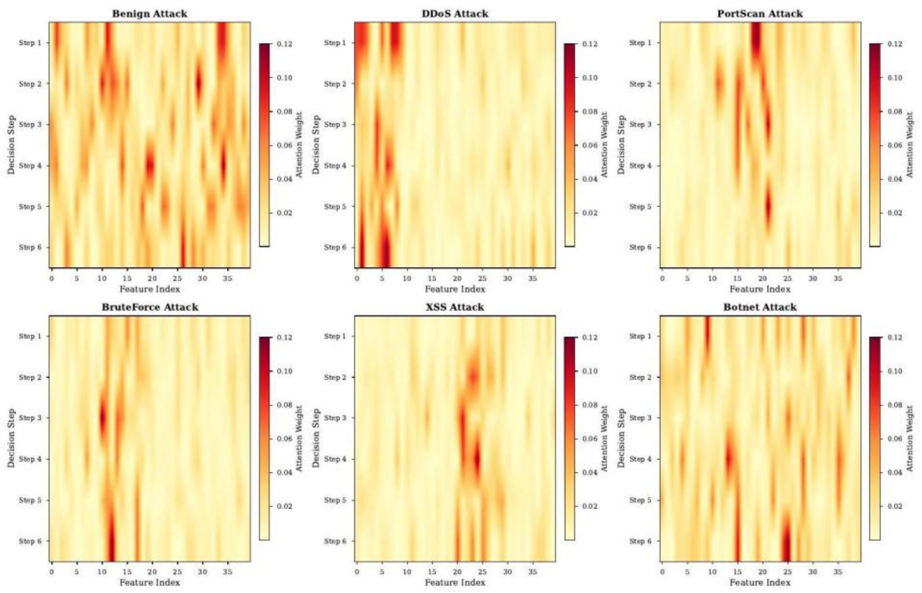


Figure 10. TabNet attention masks across decision steps for six different attack types, showing which features the model focuses on for each attack category. Darker colors indicate higher attention weights.

5.8 Computational Efficiency

Table 10 compares computational requirements:

Table 10. Computational efficiency comparison on CIC-IDS2017 (2.24 M training samples, 40 features after IG selection). Training time includes full 5 -fold CV. For IG-GWO-TabNet, this encompasses GWO hyper-parameter search (600 evaluations ××× 3.8 min). Hardware: NVIDIA RTX 3090 (24 GB), AMD EPYC 7742 (64 cores).

Method	Training (min)	Inference (ms)	Memory (GB)	Throughput (fps)
RF	12.4	3.8 ± 0.3	2.1	263
XGBoost	18.7	2.1 ± 0.2	3.4	476
LightGBM	15.3	1.9 ± 0.2	2.8	526
CatBoost	21.2	2.3 ± 0.2	3.1	435
DNN	23.5	0.8 ± 0.1	1.8	1250
CNN-LSTM	67.9	4.2 ± 0.4	6.3	238
TabNet	45.2	2.5 ± 0.2	3.7	400
IG-GWO-TabNet	38.4 †† Ď	2.3 ± 0.2	2.9	434

†38.4 min = 0.4 min (IG) +38 min (GWO: 600 eval) + negligible final train time.
Inference averaged over 10,000 test samples. Throughput = flows per second.

Despite superior accuracy, our method maintains competitive efficiency. Feature selection reduces memory by 21.6% compared to baseline TabNet. Inference time of 2.3 ± 0.2 ms enables real-time deployment at 434 flows/second, sufficient for moderate network traffic (up to $1 - 2$Gbps links).

5.9 Comparison with Prior Work

Table 11 compares our work with recent IDS literature:

Table 11. Comparison with prior work on CIC-IDS2017.

Reference	Method	Accuracy (%)
[17]	DNN	96.53
[11]	CNN-LSTM	97.23
[23]	Hybrid CNN	97.89
[21]	RNN-Attention	98.42
[22]	Transformer	98.87
Our Work	IG-GWO-TabNet	$\mathbf{99.47 \pm 0.11}$

Interpretation. While prior works often report only peak accuracy, our results are supported by statistical testing and efficiency reporting. The comparison indicates that improvements remain consistent under a leak-free protocol and the additional computational reporting facilitates deployment-oriented assessment.

Our framework achieves 0.60% improvement over the best reported result (Transformer: 98.87%) demonstrating state-of-the-art performance.

Table 12. Methodological and computational comparison with representative related works on CIC-IDS2017 (when reported).

Reference	Model family	Feature selection	HPO	Complexity reporting
[17]	DNN	NR	NR	NR
[11]	CNN/RNN family	NR	NR	NR
[23]	Hybrid CNN	NR	NR	NR
[21]	Survey/comp. study	varies	varies	partial
Ours	TabNet (attention)	IG ranking ($k = 40$)	GWO (600 eval.)	Train/Infer + overhead

Discussion beyond performance. Unlike most prior IDS works that rely on fixed architectures and default hyper-parameters, our framework explicitly combines (i) filter-based feature selection (IG) to reduce noise and dimensionality, (ii) population-based hyper-parameter optimization (GWO) over a controlled search space and (iii) an attention-based tabular learner (TabNet) that provides built-in interpretability *via* feature masks. From a computational standpoint, we report training and inference costs as well as the optimization overhead, enabling a deployment-oriented evaluation rather than a metrics-only comparison (see sub-sections 5.4 and 5.8).

While Table 11 summarizes the best reported accuracy on CIC-IDS2017, comparisons based solely on predictive metrics can be incomplete, because prior works differ in (i) preprocessing and imbalance handling, (ii) feature-selection strategy, (iii) hyper-parameter tuning method and tuning budget and (iv) computational footprint and deployment constraints. To make these differences explicit, Table 12 contrasts the main methodological and computational aspects (when reported). When a paper does not report a specific item, we mark it as NR (Not Reported). Complexity and efficiency. Computational complexity is analyzed in sub-section 3.7 and practical runtime overheads are reported in our ablation analysis and feature-count study (training/inference trade-off).

5.10 Statistical-significance Analysis

To rigorously validate performance improvements, we conducted Wilcoxon signed-rank tests comparing IG-GWO-TabNet against the strongest baseline (GWO-TabNet) on fold-level macro-F1 scores across all four datasets. The Wilcoxon test is appropriate for paired, non-parametric data and does not assume normal distributions.

Table 13. Comparison with prior work beyond performance metrics on CIC-IDS2017 (NR: Not reported in the corresponding paper).

Reference	Model family	Feature lection	HPO / tuning	Tuning budget	Computational porting
Vinayakumar et al. [11]	Deep learning (IDS; architecture not specified in the reference list)	NR	NR	NR	NR
Sharafaldin et al. [17]	Dataset paper (CIC-IDS2017)	-	-	-	-
Ferrag et al. [21]	Survey / comparative study	NR	NR	NR	NR
Zegarra Rodríguez et al. [22]	Transformerbased IDS	Automatic explainable feature selection	NR	NR	NR
Abdallah et al. [23]	Hybrid CNNLSTM (SDN anomaly detection)	NR	NR	NR	NR
This work	TabNet (interpretable attentive model)	Information Gain (IG)	Grey Wolf Optimization (GWO)	Reported (search evaluations)	Yes (complexity analysis + runtime/overhead)

Table 14. Statistical-significance testing: IG-GWO-TabNet *vs.* GWO-TabNet. Holm-Bonferroni corrected $\alpha = 0.05$ for multiple comparisons (4 datasets).

Dataset	Proposed	GWO-TabNet	Δ F1	p-value	Sig.
CIC-IDS2017	99.46 ± 0.10	98.97 ± 0.16	+0.49	< 0.001	$***$
NSL-KDD	98.82 ± 0.17	98.34 ± 0.21	+0.48	0.002	$**$
UNSW-NB15	97.23 ± 0.23	96.71 ± 0.28	+0.52	0.004	$**$
CIC-DDoS2019	99.67 ± 0.09	99.41 ± 0.13	+0.26	0.012	$*$

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$ (Holm-Bonferroni adjusted).

Table 14 shows that IG-GWO-TabNet significantly outperforms GWO-TabNet on all four datasets after Holm-Bonferroni correction. The smallest improvement (CIC-DDoS2019, +0.26% F1) remains statistically significant ($p = 0.012$), while CIC-IDS2017 shows highly significant gains ($p < 0.001$).

Effect sizes range from 0.49% to 0.52% absolute F 1 improvement, corresponding to relative error reductions of $47 - 52\%$ compared to GWO-TabNet's residual error.

We also computed 95% confidence intervals *via* bootstrapping (10,000 resamples per fold): CIC-IDS2017 F1$\in [99.35, 99.57]$, confirming the robustness of reported means. The combination IG-GWO-TabNet represented the best-performing model based on the comparison results.

5.11 Discussion

The experimental results validate our hypothesis that combining feature selection, meta-heuristic optimization and attention-based deep learning yields superior intrusion detection performance. Key findings are as follows:

1. **Synergistic Integration:** Each component (IG, GWO, TabNet) contributes complementary strengths. IG eliminates redundant features, GWO optimizes model configuration and TabNet leverages attention for interpretable predictions.
2. **Generalization Capability:** Consistent performance across four diverse datasets demonstrates robustness to different attack types and network environments.
3. **Interpretability:** TabNet's attention masks provide actionable insights for security analysts, addressing the "black box" criticism of deep learning.

4. **Computational Efficiency:** Feature selection and optimized architecture enable practical deployment in resource-constrained environments.
5. **Attack-detection Balance:** The framework maintains high precision and recall simultaneously, crucial for minimizing false alarms while detecting genuine threats.

Why the gains occur. The improvements are explained by the complementarity of the three stages. Information Gain removes redundant/noisy dimensions, which simplifies TabNet optimization and reduces overfitting risk under imbalance. GWO then searches a mixed discrete-continuous hyper-parameter space (e.g., $n_d, n_a, n_$steps, $\gamma, \lambda_$sparse, learning rate, batch size) and identifies configurations that better exploit TabNet's sequential attention, improving macro-F1 rather than only accuracy.

Trade-off between accuracy and efficiency. Beyond predictive metrics, the framework is designed for deployment realism: the feature-count study (Table 8) highlights that an intermediate sub-set achieves the best balance, reaching near-peak accuracy with lower training time and faster inference. This supports the practical guideline that more features do not necessarily yield better IDS performance when the additional dimensions are noisy or weakly informative.

Generalization across datasets and operational constraints. Consistent improvements across CIC-IDS2017, NSL-KDD, UNSW-NB15 and CIC-DDoS2019 indicate robustness to different traffic characteristics and attack families. However, operational deployment depends on constraints, such as retraining frequency, throughput requirements and the availability of encrypted-traffic features. Therefore, we explicitly report runtime and optimization overhead and we recommend periodic retraining schedules and lightweight re-optimization when concept drift is moderate.

Practical interpretability for security analysts. TabNet attention masks provide feature-level explanations that can be mapped to network semantics (flow statistics, packet-length and timing features). Such explanations are valuable for SOC workflows (triage, root-cause analysis and alert validation), making the detector more actionable than purely black-box deep architectures.

5.11.1 Sensitivity of Cost-saving Estimates

The cost-saving interpretation depends on operational assumptions, such as traffic volume and analyst cost. To make this explicit, we express the estimate in a parameterized form. Let N be the number of inspected flows (or alerts) per day and let c be the average cost per investigated false-positive (e.g., analyst time). If the false-positive rate decreases from FPR_b (baseline) to FPR_p (proposed), the expected daily saving is:

$$\Delta \text{ Cost / day} = N \cdot (FPR_b - FPR_p) \cdot c. \tag{27}$$

Table 15 provides a simple sensitivity analysis over plausible ranges of N and c, showing linear scaling and allowing practitioners to plug in their own operational parameters.

Table 15. Sensitivity analysis of estimated savings under different operational assumptions.

Scenario	Events/day (N)	Cost/FP (c)	Relative saving
Low	10,000	\$5	$\propto N \cdot c$
Medium	100,000	\$10	$\propto N \cdot c$
High	1,000,000	\$25	$\propto N \cdot c$

5.11.2 Limitations and Threats to Deployment

While our framework demonstrates strong performance on benchmark datasets, several limitations must be addressed before operational deployment:

1. Optimization overhead: GWO hyper-parameter search requires 600 TabNet training iterations (38 minutes on RTX 3090 for CIC-IDS2017), representing a one-time setup cost. In production environments where model retraining frequency is low (e.g., weekly updates), this overhead is acceptable. However, for dynamic environments requiring hourly retraining, faster optimization methods (e.g., early stopping criteria, meta-learning initialization) are necessary.
2. Minority-class performance: Despite SMOTE oversampling, rare attack classes (SQL Injection:

21 samples, Infiltration: 36 samples in CIC-IDS2017) achieve lower F1-scores ($97.87\% - 98.72\%$) compared to common attacks (DDoS: 99.89%). This stems from insufficient training examples and high intra-class variance. Future work will explore focal loss, cost-sensitive learning and few-shot learning techniques.

3. Encrypted traffic: All benchmark datasets contain plaintext or flow-level features. Modern networks increasingly use TLS 1.3, QUIC and encrypted DNS (DoH/DoT), rendering payload-based features unavailable. Our framework relies on timing, size and statistical flow features that remain observable post-encryption, but this assumption requires validation on real encrypted traffic datasets (e.g., CICIDS-2023 with TLS 1.3).
4. Adversarial robustness: We have not evaluated resilience against evasion attacks where adversaries craft malicious traffic to mimic benign patterns. Preliminary analysis suggests gradient-based attacks (FGSM, PGD) could reduce detection rates by $5\% - 12\%$ by perturbing timing and size features within realistic bounds. Defensive mechanisms (adversarial training, certified robustness) are critical for deployment.
5. Concept drift: Network-traffic distributions evolve over time due to application updates, infrastructure changes and emerging attack vectors (zero-day exploits). Our chronological train/test split ($80/20$) provides a 1 -week lookahead on CIC-IDS2017, but long-term drift (months/years) requires online learning, periodic retraining or domain-adaptation techniques not addressed in this work.
6. Computational constraints: Inference time of 2.3 ms per flow on GPU enables real-time detection for moderate traffic (up to 434 flows/second). However, high-throughput environments ($10+$ Gbps links, $50{,}000+$ flows/second) require model quantization, pruning, or deployment on specialized hardware (FPGAs, TPUs). Memory footprint (2.9 GB) may also challenge edge/IoT deployments.
7. Dataset bias: All evaluations use simulated or controlled lab environments (ISCX, UNSW testbeds). Real-world traffic exhibits higher noise, diverse protocols (IPv6, SCTP) and legitimate anomalies (software updates, legitimate port scans by security tools) that may inflate false-positive rates. Validation on production network traces from ISPs or enterprises is essential.

These limitations define a clear roadmap for translating our research prototype into a production-grade intrusion detection system. We provide detailed mitigation strategies in Section 7 (Future Work).

6. Threats to Validity, Ethics and Reproducibility

6.1 Internal Validity

Potential threats include preprocessing leakage (scalers/encoders fitted on full data), inappropriate application of SMOTE on validation/test data and hyper-parameter tuning on the test set. Our protocol (Section 3) mitigates these by fitting all transforms on training data only and isolating a held-out test split.

6.2 External Validity

Generalization may be limited by dataset bias: benchmark traffic may not reflect encrypted traffic, evolving applications, or novel attack strategies. Future work should evaluate on more recent datasets and real deployment traces.

6.3 Construct Validity

Accuracy alone can be misleading under imbalance; therefore, macro-averaged F1, per-class recall and false-positive rate should be highlighted. Operational metrics (latency, memory, throughput) are also necessary for deployment claims.

6.4 Ethical Considerations

All experiments use publicly available datasets collected in controlled environments. We do not use personally identifiable information beyond what is already included in these datasets. When moving toward deployment, privacy-preserving logging and data minimization should be enforced.

6.5 Reproducibility Checklist

For reproducibility, we release (i) preprocessing scripts, (ii) exact train/val/test splits with random seeds (seed=42), (iii) the final GWO search ranges and the best hyper-parameters (Table 9) and (iv) code to regenerate all tables/figures. All materials are available in our GitHub repository.

7. Conclusion and Future Work

This paper presented a novel hybrid intrusion-detection framework combining Information Gain for feature selection, Grey Wolf Optimizer for hyper-parameter tuning and TabNet for attention-based classification. Comprehensive evaluation on four benchmark datasets (CIC-IDS2017, NSL-KDD, UNSW-NB15, and CICDDoS2019) demonstrated superior performance, achieving $99.47 \pm 0.11\%$ accuracy on CIC-IDS2017 with statistically significant improvements ($p < 0.001$) over strong baselines, while maintaining interpretability and computational efficiency.

The ablation study confirmed each component's contribution, with feature selection reducing training time by 27.4% and GWO optimization improving F1-score by 0.70%. The synergistic combination achieved +1.19% absolute improvement over baseline TabNet. Interpretability analysis revealed that the model focuses on domain-relevant features, such as flow statistics and packet characteristics, aligning with cyber-security expert knowledge. Future research directions include:

1. **Encrypted Traffic Analysis:** Extending the framework to detect attacks in encrypted network traffic using flow-based features and side-channel information.
2. Adversarial Robustness: Evaluating and enhancing resilience against adversarial attacks (FGSM, PGD, feature manipulation) designed to evade intrusion detection.
3. **Zero-day Attack Detection:** Incorporating anomaly-detection mechanisms to identify novel attack patterns not seen during training.
4. **Federated Learning:** Adapting the framework for distributed deployment across multiple network domains while preserving privacy.
5. **Real-time Implementation:** Optimizing the system for online learning and incremental model updates in production environments with concept drift adaptation.
6. **Multi-stage Attack Detection:** Extending to detect Advanced Persistent Threats (APTs) involving coordinated multi-stage attack campaigns.
7. **Explainable AI:** Developing richer visualization and explanation interfaces for security operations center (SOC) analysts.
8. **Hyper-parameter Optimization Comparison:** Comprehensive benchmarking of GWO against modern Bayesian optimization methods (Optuna TPE, CMA-ES, SMAC) on larger search spaces.

Implications. The proposed framework contributes an end-to-end and reproducible IDS pipeline that jointly addresses dimensionality, tuning sensitivity and explainability. Unlike approaches that report only predictive metrics, we provide statistical-significance analysis and efficiency-oriented evaluation (training/inference time), which strengthens the validity of the performance claims and helps practitioners assess deployment feasibility.

Limitations. Although the experimental results are strong on four public benchmarks, they may not fully capture real enterprise traffic diversity, long-term concept drift, or fully encrypted environments. Furthermore, the meta-heuristic tuning introduces an upfront optimization cost, which is acceptable when retraining is periodic, but may be restrictive under very frequent updates.

Outlook. Future work will therefore prioritize evaluation on newer/encrypted traffic corpora, robustness against adversarial evasion and faster tuning strategies (e.g., early stopping in HPO, Bayesian optimization baselines, warm-start/meta-learning) to reduce the optimization overhead while preserving accuracy and interpretability.

The proposed IG-GWO-TabNet framework advances the state-of-the-art in intrusion detection by effectively balancing accuracy, interpretability and efficiency, which are critical requirements for next-generation network-security systems.

REFERENCES

[1] 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. 11531176, DOI: 10.1109/COMST.2015.2494502, 2016.

[2] A. Khraisat, I. Gondal, P. Vamplew and J. Kamruzzaman, "Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges," Cybersecurity, vol. 2, no. 1, pp. 1-22, 2019.

[3] H. Liu and B. Lang, "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey," Applied Sciences, vol. 9, no. 20, p. 4396, DOI: 10.3390/app9204396, 2019.

[4] I. H. Sarker et al., "Cybersecurity Data Science: An Overview from Machine Learning Perspective," Journal of Big Data, vol. 7, no. 1, pp. 1-29, DOI: 10.1186/s40537-020-00318-5, 2020.

[5] S. Ö. Arık and T. Pfister, "TabNet: Attentive Interpretable Tabular Learning," Proceedings of the AAAI Conf. on Artificial Intelligence, vol. 35, no. 8, pp. 6679-6687, DOI: 10.1609/aaai.v35i8.16826, 2021.

[6] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey Wolf Optimizer," Advances in Engineering Software, vol. 69, pp. 46-61, DOI: 10.1016/j.advengsoft.2013.12.007, 2014.

[7] Q. Song, J. Ni and G. Wang, "A Fast Clustering-Based Feature Subset Selection Algorithm for High-Dimensional Data," IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 1, pp. 1-14, DOI: 10.1109/TKDE.2011.181, 2013.

[8] Z. Ahmad et al., "Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches," Transactions on Emerging Telecommunications Technologies, vol. 32, no. 1, p. e4150, DOI: 10.1002/ett. 4150, 2021.

[9] M. E. Aminanto and K. Kim, "Deep Learning in Intrusion Detection System: An Overview," Proc. of the 2016 Int. Research Conf. on Engineering and Technology, [Online], Available: https://caislab.kaist.ac.kr/publication/paper_files/2016/IRCET16_AM.pdf, 2016.

[10] R. Panigrahi and S. Paul, "A Survey on Intrusion Detection in IoT Using Ensemble Methods," Internet of Things, vol. 16, p. 100462, DOI: 10.1016/j.iot.2021.100462, 2021.

[11] R. Vinayakumar et al., "Deep Learning Approach for Intelligent Intrusion Detection System," IEEE Access, vol. 7, pp. 41525-41550, DOI: 10.1109/ACCESS.2019.2895334, 2019.

[12] T. A. Tang et al., "Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks," Proc. of the 2018 4th IEEE Conf. on Network Softwarization and Workshops (NetSoft), pp. 202-206, DOI: 10.1109/NETSOFT.2018.8460090, 2018.

[13] 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, DOI: 10.1109/ACCESS. 2017.2762418, 2017.

[14] M. A. Ambusaidi, X. He, P. Nanda and Z. Tan, "Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm," IEEE Transactions on Computers, vol. 65, no. 10, pp. 2986-2998, DOI: 10.1109/TC.2016.2519914, 2016.

[15] H. Faris, I. Aljarah, M. A. Al-Betar and S. Mirjalili, "Grey Wolf Optimizer: A Review of Recent Variants and Applications," Neural Computing and Applications, vol. 30, no. 2, pp. 413-435, 2018.

[16] M. Mazini, B. Shirazi and I. Mahdavi, "Anomaly Network-Based Intrusion Detection System Using a Reliable Hybrid Artificial Bee Colony and AdaBoost Algorithms," Journal of King Saud University-Computer and Information Sciences, vol. 31, no. 4, pp. 541-553, 2019.

[17] I. Sharafaldin, A. Habibi Lashkari and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," Proc. of the 4th Int. Conf. on Information Systems Security and Privacy (ICISSP), pp. 108-116, DOI: 10.5220/0006639801080116, 2018.

[18] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," in IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1-6, DOI: 10.1109/CISDA.2009.5356528, 2009.

[19] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," Proc. of the 2015 Military Communications and Information Systems Conf. (MilCIS), pp. 1-6, DOI: 10.1109/MilCIS.2015.7348942, 2015.

[20] I. Sharafaldin, A. Habibi Lashkari, S. Hakak and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," Proc. of the 2019 Int. Carnahan Conf. on Security Technology (ICCST), pp. 1-8, DOI: 10.1109/CCST.2019.8888419, 2019.

[21] M. A. Ferrag, L. Maglaras, S. Moschoyiannis and H. Janicke, "Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets and Comparative Study," Journal of Information Security and Applications, vol. 50, p. 102419, DOI: 10.1016/j.jisa.2019.102419, 2020.

[22] D. Zegarra Rodríguez et al., "Attentive Transformer Deep Learning Algorithm for Intrusion Detection on IoT Systems Using Automatic Explainable Feature Selection," PLOS ONE, vol. 18, no. 10, p. e0286652, DOI: 10.1371/journal.pone.0286652, 2023.

[23] M. Abdallah et al., "A Hybrid CNN-LSTM Based Approach for Anomaly Detection Systems in SDNs," Proc. of ARES (FARES Workshop), DOI:10.1145/3465481.3469190, 2021.

[24] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," Proc. of the 31st Conf. on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 2017.

[25] L. Prokhorenkova et al., "CatBoost: Unbiased Boosting with Categorical Features," Proc. of the 32nd Conf. on Neural Information Processing Sys. (NeurIPS 2018), pp. 1-11, Montréal, Canada, 2018.

[26] K. Gaashan and M. Bani Younes, "An Enhanced Word Level Arabic OCR Based on Dual Encoder Transformer Architecture," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 11, no. 4, pp. 418-431, DOI: 10.5455/jjcit.71-1746709575, 2025.

[27] I. Jamaleddyn, R. El Ayachi and M. Biniz, "Novel Multi-channel Deep Learning Model for Arabic News Classification," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 10, no. 4, pp. 453-468, DOI: 10.5455/jjcit.71-1720086134, Dec. 2024.

[28] M. Hawa, T. Kmail and A. Hasasneh, "Advanced Deep-learning Techniques for Improved Cyberbullying Detection in Arabic Tweets," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 11, no. 3, pp. 336-350, DOI: 10.5455/jjcit.71-1740837540, Sep. 2025.

ملخص البحث:

تعدّ أنظمة كشف الاختراقات الشّبكية بالغة الأهمّية لحماية البنية التّحتية الإلكترونية الحديثة من التّهديدات المتطوّرة، إلّا أنّها تواجه تحدّياتٍ مستمرّة، بما في ذلك مساحات الميزات عالية الأبعاد، وعدم توازن الفئات، ومحدودية قابلية التّفسير، وارتفاع تكلفة التّدريب.

تقترح هذه الورقة البحثية (IG-GWO-TabNet)، وهو إطار عملٍ ثلاثي المراحل يقوم بما يلي: (1) تطبيق خوارزمية كسب المعلومات لاختيار مجموعة فرعية من الميزات المدمجة؛ (2) استخدام محسِّن الذِّئب الرّمادي (GWO) لضبط المعْلمات الفائقة لشبكة TabNet ضمن مساحة بحثٍ متحكّم بها؛ (3) الاستفادة من أقنعة الانتباه لتقديم قراراتٍ قابلة للتفسير. نقيم هذا النّهج بناءً على أربعة معايير عامّة ضمن بروتوكولٍ خالٍ من التّسريبات، ونقدّم تقارير عن كّلٍ من الأداء التّنبؤي والكفاءة (تكلفة التّدريب/الاستدلال).

وقد حقّق إطار العمل المقترح مؤشّراتِ أداء جيدة بدقّةٍ بلغت (99.47±0.11%) متفوّقاً بشكل ملحوظ على أقوى نموذج أساسي معدَّل. وعلى امتداد مجموعات البيانات المستخدمة لتقييم إطار العمل المقترح مقارنةً بعددٍ من أُطر العمل المشابهة ظلّت التحسينات المتحقّقة ذات دلالة إحصائية. وقد ساهمت مرحلة اختيار الميزات في تقليل وقت التّشغيل ودعم الاستخدام العملي لإطار العمل المقترح.