Ollscoil Teicneolaíochta an Atlantaigh

Atlantic Technological University

Intelligent Intrusion Detection System Using Machine Learning and Grey Wolf Optimisation

By
Divya Guruvaiah Naidu

September 2, 2025

Masters Thesis

Submitted in partial fulfillment for the award of **Master of Science in Computing** to the Department of Computer Science & Applied Physics, Atlantic Technological University (ATU), Galway.

M.Sc. in Computing

Contents

List of Figures

List of Tables

Listings

List of Algorithms

Abstract

Modern networks carried large volumes of partially encrypted traffic, making intrusions difficult to detect without increasing computational cost. An intrusion detection system (IDS) was developed on the CICIDS2017 benchmark of labelled network flows. The approach combined tree-based learners — Decision Tree, Random Forest, Extra Trees, and XGBoost — with Modified Binary Grey Wolf Optimisation (MBGWO) for feature selection. MBGWO maximised cross-validated predictive accuracy while minimising the number of selected features, yielding compact subsets that reduced runtime without degrading discrimination. The implementation comprised coordinated stages. A data-sampling routine produced a balanced subset to preserve class coverage while keeping experiments tractable. Preprocessing coerced inputs to numeric form, replaced NaN and ±Inf values common in rate-derived features, clipped outliers to float-safe ranges, and applied median imputation. Each classifier was first trained on all features to obtain baseline metrics for comparison. Feature selection then used MBGWO in a leakage-safe search: a binary mask encoded candidate subsets; a multi-objective fitness combined stratified k-fold accuracy with a sparsity term; and a minimum-features constraint preserved model stability. Evaluation prioritised the F1-weighted score due to class imbalance and additionally reported precision, recall, accuracy, and area under the receiver operating characteristic curve (AUC). Confusion matrices and per-class reports complemented the aggregate metrics. Results showed a consistent pattern. XGBoost trained on all features achieved strong performance (F1-weighted ≈ 0.997; AUC ≈ 0.999). Applying MBGWO delivered steady gains for bagging-style models while reducing the feature count: Random Forest improved from ≈ 0.992 to ≈ 0.996 F1-weighted and Extra Trees from ≈ 0.984 to ≈ 0.996, with no AUC loss. The work produced a reproducible, resource-conscious IDS pipeline that lowered dimensionality and computational load while preserving detection quality. Remaining variance on minority classes suggested scope for cost-sensitive training, calibrated thresholds, and targeted resampling. A per-class leader-selection ensemble was proposed as future work, alongside evaluation under throughput and latency constraints for real-time operation in practical deployments.

Acknowledgments

I would like to express my sincere thanks to my academic supervisor, **Mr. Douglas Mota Dias**, whose guidance, constructive feedback, and steady encouragement were invaluable at every stage of this research. His expertise in cybersecurity and machine learning helped me refine the research questions, choose appropriate methods, and maintain a clear focus throughout the project. I am grateful to the faculty and staff of Atlantic Technological University, and in particular to the School of Science and Computing, for fostering a supportive academic environment. Access to facilities, computing resources, and timely administrative assistance enabled me to progress efficiently. I also thank the ATU IT Service Desk for help with software and systems, and the Library staff for assistance with literature access and referencing. This work made extensive use of openly available tools and datasets. I acknowledge the Canadian Institute for Cybersecurity for providing the CICIDS2017 dataset, which formed the empirical foundation of this study. I also appreciate the open-source community whose software underpinned the implementation and experiments conducted here. Finally, I thank my family and friends for their patience, practical help, and encouragement during my studies, and my classmates and colleagues for helpful discussions and feedback. Any remaining errors or omissions are my own.

Chapter 1

Introduction

1.1 Background

The threat of cyberattacks is increasing in both sophistication and frequency as the digital ecosystem evolves. From large-scale data breaches to state-sponsored cyberwarfare, modern digital infrastructure faces persistent risks of intrusion [1]. Intrusion Detection Systems (IDS) serve as a crucial line of defence to identify and halt unauthorised or malicious activity within network traffic. However, traditional IDS approaches—primarily signature-based—struggle to adapt to evolving attack patterns, zero-day exploits, and polymorphic malware [2, 1].

To address these limitations, cybersecurity solutions increasingly adopt intelligent, data-driven strategies. The incorporation of Machine Learning (ML) and Artificial Intelligence (AI) into IDS design marks a paradigm shift by enabling systems to learn from historical data and generalise to novel threats. ML-based IDS can detect subtle anomalies and deviations in traffic behaviour (e.g., packet length, flow duration, connection frequency) that might otherwise go unnoticed [2, 1]. These systems can also operate in near real time, adapt to shifting network patterns, and provide proactive threat mitigation.

The CICIDS2017 dataset, developed by the Canadian Institute for Cybersecurity, is widely recognised for training and evaluating ML-based IDS models. It simulates realistic network traffic, including benign flows and diverse modern attacks such as DoS, DDoS, brute-force, infiltration, and web exploits [3]. Its authenticity and richness make it one of the most frequently used benchmarks for IDS research [4].

Nonetheless, the high dimensionality of such datasets presents challenges. Redundant, irrelevant, or noisy features can degrade classifier performance, increase computational burden, and contribute to overfitting [5, 6]. As a result, feature selection (FS) becomes essential to reduce complexity while improving accuracy and interpretability.

Metaheuristic optimisation algorithms have been widely studied for FS because of

their ability to perform global search and escape local minima [1, 7]. Among these, the Grey Wolf Optimiser (GWO), inspired by the social hierarchy and hunting behaviour of grey wolves, has gained attention for its simplicity and strong balance between exploration and exploitation [8]. Variants such as the Modified Binary GWO (MBGWO) further enhance convergence and suitability for binary FS tasks [9, 10].

This study proposes an intelligent IDS framework that integrates MBGWO for optimal FS with a Random Forest (RF) classifier—an ensemble learning method known for robustness, interpretability, and strong detection performance [11]. The aim is to deliver a high-accuracy and interpretable IDS solution capable of addressing real-world challenges.

1.2 Problem Statement

As cyber threats grow increasingly complex, traditional IDS are proving inadequate. Signature-based methods are ineffective against novel or zero-day attacks, and anomaly-based systems often suffer from high false positive rates [2, 1]. Moreover, the effectiveness of ML-based IDS depends heavily on the quality and relevance of the input features. High-dimensional datasets such as CICIDS2017 often contain redundant attributes that lead to degraded performance and increased computational cost [5].

Although FS has been investigated with statistical and wrapper-based methods, scalability and optimality remain open challenges. Metaheuristic algorithms such as GWO show promise but have been underexplored in hybrid IDS frameworks. Few studies systematically integrate modified GWO variants with ensemble classifiers such as RF, which are particularly well-suited to high-dimensional IDS data [10, 12, 13].

This research addresses these shortcomings by implementing a hybrid IDS framework that leverages MBGWO-based FS and RF classification to detect multiple intrusion types in CICIDS2017.

1.3 Objectives

The primary objective of this study is to design and evaluate an intelligent IDS framework using the CICIDS2017 dataset, combining FS with ensemble learning. The specific objectives are:

1. **Dataset Exploration and Preprocessing**
    - Load and clean CICIDS2017 [3].
    - Handle missing and infinite values.
    - Encode categorical variables and normalise continuous features.

2. **Feature Selection Using MBGWO**
   - Implement a Modified Binary Grey Wolf Optimiser [9].
   - Select an optimal feature subset to reduce dimensionality.
3. **Model Building and Training**
   - Train an RF classifier [11] on both full and reduced feature sets.
   - Apply cross-validation to evaluate generalisability.
4. **Evaluation and Visualization**
   - Assess models using accuracy, precision, recall, F1-score, and ROC-AUC [1].
   - Visualise confusion matrices, feature importance plots, and ROC curves.
5. **Reporting and Analysis**
   - Compare RF performance before and after FS.
   - Benchmark against boosting methods (XGBoost, LightGBM) [12, 13].
   - Interpret the role of selected features in the context of IDS.

1.4 Significance of the Study

This research makes both theoretical and practical contributions to IDS research. Theoretically, it demonstrates the integration of MBGWO with ensemble learning for FS in high-dimensional IDS datasets, addressing limitations of previous approaches [9, 10]. Practically, the proposed framework is scalable, interpretable, and suitable for enterprise, cloud, and critical infrastructure environments. By incorporating explainable elements such as feature importance plots, the framework assists analysts in decision-making. The use of an open dataset and reproducible methodology also encourages future benchmarking and extensions in intelligent IDS development.

Chapter 2

Research Methodology

2.1 Overview

This study proposes a comprehensive methodology for developing an intelligent Intrusion Detection System (IDS) that integrates classical machine learning with metaheuristic optimization.

The method consists of several systematic stages: dataset acquisition and preprocessing, optimal feature selection using Modified Binary Grey Wolf Optimization (MBGWO), threat classification using a Random Forest classifier, and model evaluation via standard classification metrics.

The implementation was done in Python using Jupyter Notebook on Visual Studio Code or Google Colab, utilizing popular libraries such as `scikit-learn`, `pandas`, `numpy`, and `matplotlib`. The aim is to achieve high detection accuracy while minimizing false positives and computational overhead.

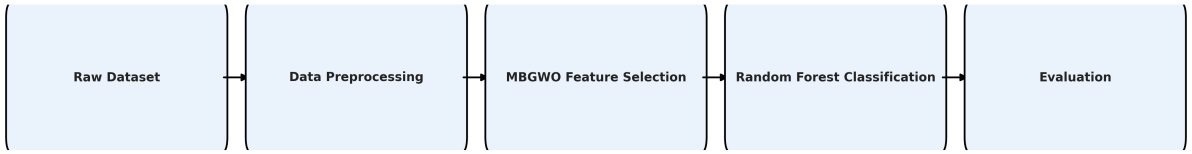

Figure 2.1: IDS System Pipeline: Data preprocessing → MBGWO → classification → evaluation.

2.2 Dataset: CICIDS2017

The experiments are based on the CICIDS2017 dataset from the Canadian Institute of Cybersecurity, a widely recognised and up-to-date dataset for intrusion detection. It replicates realistic network conditions and includes diverse attack scenarios like Denial-of-Service (DoS), Port Scanning, Botnets, and Brute Force. The dataset comprises approximately 2.85 million traffic flows, each described by over 80 features, including statistical and flow-level attributes. A representative 10% subset of the data was selected to maintain class diversity while keeping computations feasible. Features include `Flow Duration`, `Packet Length Mean`, `Fwd Packet Count`, and a class label indicating benign or attack.

2.3 Preprocessing of the Data

Preprocessing steps were performed to ensure consistency and compatibility:

- Missing, null, and infinite values were identified and removed.
- Categorical features like protocol type were label-encoded into numerical form.
- Numerical features were normalised using `StandardScaler` to mitigate skewed distributions.

Labels were prepared in two ways:

1. Binary classification: all attacks labeled as `Attack`, others as `Benign`.
2. Multiclass classification: each attack type kept its specific label.

To address class imbalance, stratified sampling and SMOTE (Synthetic Minority Over-sampling Technique) were applied during model training.

2.4 Feature Selection Using MBGWO

2.4.1 Grey Wolf Optimiser (GWO)

GWO is a population-based metaheuristic inspired by the social hierarchy and hunting strategies of grey wolves, including encircling, hunting, and attacking prey. Wolves are categorised into four roles: Alpha (leader), Beta, Delta, and Omega.

2.4.2 Binary Representation and Enhancements

In this study, a binary version of GWO is used, where each wolf represents a binary vector — 1 for selecting a feature, 0 for ignoring it. Enhancements include:

- **Chaotic maps** for initialising the population to ensure diversity.

- **Dynamic convergence factor** to balance exploration and exploitation.
- **Mutation operator** inspired by Genetic Algorithms to avoid local optima.

2.4.3 Fitness Function

The fitness function evaluates a candidate feature subset by considering both classification performance and feature count:

$$\text{Fitness} = \alpha \cdot \text{Accuracy} + (1 - \alpha) \cdot \left(1 - \frac{\text{Selected Features}}{\text{Total Features}}\right)$$

where α balances accuracy vs. subset size. A lightweight classifier (e.g., logistic regression) and 5-fold cross-validation are used to compute accuracy.

2.5 Classification with Random Forest

After selecting optimal features via MBGWO, a Random Forest (RF) classifier is trained. RF is an ensemble technique using bootstrapped decision trees. It provides robustness, interpretability, and resistance to overfitting—making it suitable for IDS tasks.

- Number of trees: 100
- Splitting criterion: Gini impurity
- No maximum tree depth (trees grow fully)
- Bootstrap: enabled
- Train-test split: 70%-30%
- Stratified sampling ensures balanced representation

Default hyperparameters were used for reproducibility. Hyperparameter tuning (e.g., GridSearchCV) can further improve performance but was not the focus of this study. For comparison, additional classifiers such as XGBoost and LightGBM were also trained on the same feature sets to benchmark performance.

2.6 Evaluation Metrics

The following metrics are used to assess model performance:

- **Accuracy** – Proportion of correctly classified samples.
- **Precision** – Proportion of correctly predicted attacks.
- **Recall (Sensitivity)** – Proportion of actual attacks detected.

- **F1 Score** – Harmonic mean of precision and recall.
- **Confusion Matrix** – True/false positive and negative counts.
- **ROC-AUC** – Discriminative ability across all thresholds.

For fitness evaluation within MBGWO, 5-fold cross-validation with Logistic Regression was used. For final IDS assessment, a 70–30 hold-out split with Random Forest was applied.

2.7 Visualization and Interpretation

Visual analysis was used to interpret model behaviour and results:

- Confusion matrix heatmaps plotted using `Seaborn`.
- ROC curves for binary and multiclass classification.
- Feature importance plotted using `RandomForestClassifier.feature_importances_`.
- Box plots to compare feature distributions between benign and attack traffic.

These visualisations aid in model transparency and help analysts trust the system.

2.8 Experiment Setup

Experiments were conducted on Google Colab using:

- Python 3.11
- Libraries: `scikit-learn`, `pandas`, `numpy`, `matplotlib`, `seaborn`
- Colab CPU runtime was used for experiments, with GPU acceleration optionally used for large-scale tests."
- Random seed initialisation for reproducibility

2.9 Summary of Methodology

The full pipeline followed in this study can be summarised as:

1. **Data Acquisition:** Loading and sampling CICIDS2017 dataset.
2. **Preprocessing:** Cleaning, encoding, normalisation, and label creation.
3. **Feature Selection:** Optimal feature subset chosen via MBGWO.

4. **Model Training:** Random Forest classifier trained on selected features.
5. **Evaluation:** Multiple performance metrics computed.
6. **Visualisation :** Graphical interpretation for explainability and trust.

This methodology guarantees a data-driven, interpretable, and efficient solution to detect sophisticated network intrusions.

Chapter 3

Literature Review

3.1 Introduction

The need for robust cybersecurity has grown alongside the increasing complexity of network infrastructures and the sophistication of attacks. Intrusion Detection Systems (IDS) have therefore remained an active area of research for over two decades. While the early generations of IDS relied on static rules and pattern matching, modern systems increasingly draw upon machine learning (ML) and optimisation techniques to cope with large-scale and evolving threats. This chapter critically reviews the body of research relevant to this study, focusing on three main aspects: (i) the transition of IDS from traditional approaches to learning-based systems, (ii) the importance of benchmark datasets and feature selection in high-dimensional traffic data, and (iii) the adoption of metaheuristic optimisation, with emphasis on the Grey Wolf Optimiser (GWO) and its binary and modified variants. The review concludes by identifying research gaps and outlining how this study positions itself within the literature.

3.2 Intrusion Detection Systems

IDS are broadly classified as *signature-based* or *anomaly-based*. Signature-based systems detect intrusions by matching traffic against known attack patterns. They are effective and efficient for previously observed threats but fail to capture zero-day exploits or polymorphic malware. By contrast, anomaly-based systems attempt to learn what constitutes "normal" behaviour and then flag deviations as suspicious. While more flexible, they often suffer from high false-positive rates, as benign but unusual traffic may be wrongly identified as malicious [2, 1].

The literature reflects a gradual evolution from these rigid approaches toward more adaptive learning-based IDS. Surveys such as Buczak and Guven's [2] show that the field has steadily moved towards ML models to overcome the weaknesses of static rules, while Ahmad et al. [1] highlight that hybrid systems, which combine anomaly detection with ML, are increasingly deployed in practice. This evolution sets the stage for methods that can adapt to new threats without excessive reliance on manual rule updates.

3.3 Machine Learning in IDS

The adoption of ML has transformed IDS research by enabling models to automatically distinguish between benign and malicious traffic. Early approaches such as Decision Trees and Support Vector Machines demonstrated the feasibility of data-driven intrusion detection but suffered from scalability issues when applied to large datasets. Random Forest (RF), introduced by Breiman [11], emerged as a reliable ensemble approach, aggregating predictions from multiple trees to improve generalisation. RF has since become a widely accepted baseline for IDS due to its robustness, interpretability, and resistance to overfitting [2].

The field has also benefited from boosting frameworks. XGBoost [14] and LightGBM, for example, provide fast and scalable alternatives with built-in regularisation. Khan et al. [12] showed that XGBoost, when coupled with FS, produced highly accurate IDS models on CICIDS2017, while Alzubi and Nayyar [13] reported similarly strong results using LightGBM. Chen et al. [15] further demonstrated that hybrid approaches combining FS with ensembles consistently outperform models trained on full feature sets. Ensemble learning is also gaining traction in operational environments; Roy et al. [16] illustrate how ensembles can enhance enterprise-level intrusion detection.

Deep learning (DL) has also been explored as an alternative. Shone et al. [17] applied stacked autoencoders to learn abstract representations of traffic, while Yin et al. [18] employed recurrent neural networks to capture sequential dependencies. Although these models often achieve strong accuracy, their practical deployment is hampered by heavy computational demands and a lack of transparency, which makes it difficult for analysts to trust their outputs. As Ahmad et al. [1] argue, tree-based ensembles often strike a more effective balance between accuracy, scalability, and interpretability than deep architectures.

3.4 Benchmark Datasets

The reliability of IDS evaluation depends on the dataset used. Early benchmarks such as KDD Cup 1999 and NSL-KDD were widely studied but are now regarded as outdated, due to redundancy and unrealistic traffic patterns [1]. To overcome these limitations, the Canadian Institute for Cybersecurity introduced the CICIDS2017 dataset, which simulates realistic traffic flows incorporating both benign activity and diverse attack types, including DoS, DDoS, brute force, infiltration, and web-based exploits [3]. With over 2.8 million flows and 80+ statistical features, it has become the de facto standard for IDS benchmarking.

Despite its richness, CICIDS2017 presents challenges. Alshamrani and Aldwairi [4] emphasised that results can vary significantly depending on how the dataset is preprocessed and whether FS is applied. This underlines the importance of transparent and reproducible pipelines in IDS research. Consequently, many recent works—including this study—adopt CICIDS2017 as the benchmark dataset for experimentation.

3.5 Feature Selection in IDS

High-dimensional data can contain redundant or irrelevant features that degrade classifier accuracy, increase computational cost, and reduce interpretability. Feature selection (FS) aims to identify a subset of attributes that contribute most to detection.

FS methods are typically categorised into:

- **Filter methods**, which apply statistical tests (e.g., correlation, mutual information). Yu et al. [5] demonstrated that the Fast Correlation-Based Filter (FCBF) is an efficient option for IDS, offering a strong trade-off between accuracy and speed.
- **Wrapper methods**, which iteratively evaluate feature subsets using a classifier. While accurate, these methods are computationally costly.
- **Embedded methods**, which integrate FS within model training (e.g., tree-based feature importance).

Hybrid FS approaches are increasingly used in IDS. Aljawarneh et al. [6] combined filter and wrapper methods to produce more compact feature sets, improving accuracy while reducing false alarms. Similarly, Chen et al. [15] embedded FS into ensemble models and observed significant performance gains. These findings demonstrate that FS is not only a preprocessing step but also a determinant of IDS success.

3.6 Metaheuristic Optimisation for Feature Selection

Traditional FS methods are limited in their ability to handle complex, non-linear feature interactions. Metaheuristic algorithms offer a more powerful alternative by framing FS as a global optimisation problem. These algorithms, inspired by natural processes, efficiently search large solution spaces and avoid local minima [1]. Common examples include Genetic Algorithms, Particle Swarm Optimisation (PSO), and Ant Colony Optimisation.

Their value for IDS is evident. Mahbooba and Dehzangi [19] proposed a hybrid GWO–PSO algorithm that improved intrusion detection in IoT environments. Hussain and Naeem [20] introduced chaotic mapping into GWO, demonstrating enhanced performance in IDS FS tasks. A systematic review by Sahu et al. [7] concluded that metaheuristics provide a scalable, flexible, and effective framework for FS in cybersecurity contexts.

3.7 Grey Wolf Optimiser and Its Variants

The Grey Wolf Optimiser (GWO), introduced by Mirjalili et al. [8], is inspired by the hierarchical leadership and cooperative hunting behaviour of grey wolves. Its design balances exploration (searching globally) and exploitation (refining local solutions), making it well suited to FS. The algorithm simulates three core hunting stages: encircling, attacking, and searching for prey.

Over time, GWO has been extended into several variants. The Multi-Objective GWO (MOGWO) enables simultaneous optimisation of multiple criteria, such as minimising feature set size while maximising detection accuracy [21]. Wu et al. [22] applied a hybrid GWO in IDS and found it superior to GA and PSO baselines. Sharaf and El-Sayed [9] introduced the Modified Binary GWO (MBGWO), incorporating chaotic initialisation and adaptive parameters, which improved convergence and produced smaller, high-quality feature subsets. Sharaf et al. [10] validated MBGWO within hybrid IDS pipelines, showing consistent gains in accuracy and efficiency. Further contributions include Zhang and Wang's improved GWO for balanced detection [23] and Amutha and Sumathi's integration of GWO with deep ensembles [24].

3.8 Comparative Studies and Related Work

A number of comparative works underline both the promise and shortcomings of current IDS research.

Khan et al. [12] combined FS with XGBoost and achieved high accuracy on CICIDS2017, though at the cost of reduced interpretability.

Alzubi and Nayyar [13] validated LightGBM's speed and efficiency, but without explicit FS integration, their approach risked overfitting. Wu et al. [22] and Sharaf et al.

[9] demonstrated that GWO-based FS generally outperforms traditional FS approaches, while Amutha and Sumathi [24] confirmed that GWO can also complement deep learning ensembles. By contrast, Shone et al.

[17] and Yin et al. [18] illustrated the high potential of DL but also its resource demands and lack of explainability.

3.9 Research Gaps

From the above, three clear gaps emerge. First, while MBGWO has been proposed for FS, it has not been systematically integrated with Random Forest classifiers on CICIDS2017.

Second, benchmarking across RF, XGBoost, and LightGBM under a unified and reproducible pipeline remains limited, hindering direct comparison.

Third, many IDS studies emphasise accuracy but pay insufficient attention to interpretability and scalability, both of which are critical for operational deployment.

Table 3.1: Summary of Selected IDS Studies Integrating Feature Selection and Classification

Author(s)	**Dataset**	**Method**	**Findings**	**Limitations**
Sharaf et al. (2023) [10]	CICIDS2017	MBGWO + Classifiers	Increased accuracy and reduced complexity	Algorithmic complexity
Khan et al. (2022) [12]	CICIDS2017	XGBoost + FS	Achieved superior accuracy with reduced features	Lower interpretability
Alzubi & Nayyar (2022) [13]	CICIDS2017	LightGBM	High accuracy and speed	No explicit FS, risk of overfitting
Amutha & Sumathi (2022) [24]	Mixed IDS datasets	GWO + Deep Ensembles	Boosted performance with FS integration	High resource requirements
Sharaf & El-Sayed (2022) [9]	Benchmark IDS	MBGWO FS	Generated compact feature subsets with strong accuracy	Limited validation across datasets
Wu et al. (2021) [22]	CICIDS2017	Hybrid GWO	Outperformed GA and PSO in FS tasks	Computationally expensive
Shone et al. (2018) [17]	NSL-KDD	Stacked Autoencoders	Learned compact representations, improved detection	High computational cost
Yin et al. (2017) [18]	KDD Cup 1999	RNN IDS	Effective sequential modelling of traffic	Poor interpretability, heavy training cost

3.10 Summary

This review has traced the development of IDS from signature-based detection to ML-driven approaches, assessed benchmark datasets, and evaluated FS and metaheuristic optimisation techniques. While prior work has validated the potential of GWO variants and ensemble learning for IDS, systematic integration remains limited. This dissertation addresses these gaps by implementing a hybrid MBGWO–Random Forest framework, benchmarking against boosting classifiers, and focusing on reproducibility and operational relevance.

Chapter 4

System Design

4.1 Introduction

This chapter presents the architectural blueprint and design methodology for the proposed Intrusion Detection System (IDS), which integrates Modified Binary Grey Wolf Optimisation (MBGWO) for feature selection and a Random Forest classifier for intrusion detection. The design explains *how* the system processes the CICIDS2017 dataset, optimises features, and classifies traffic instances with high accuracy and interpretability. Each component of the pipeline has been designed with modularity, scalability, and reproducibility in mind, aligning with established practices from IDS literature [2, 1].

4.2 System Architecture

The IDS was designed as a modular pipeline architecture, consisting of five key components: data preprocessing, feature selection using MBGWO, model training using Random Forest, evaluation of model performance, and visualisation for interpretability. This modular structure ensures flexibility, allowing each stage to be independently modified or improved. The overall system pipeline is shown in Figure 4.1. It illustrates how raw CICIDS2017 traffic data is transformed into processed input, optimised features are selected, and classification results are evaluated.

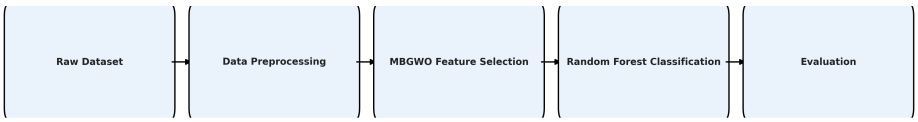

Figure 4.1: System Architecture Pipeline: From raw CICIDS2017 dataset through preprocessing, MBGWO feature selection, classification, and evaluation.

4.2.1 Component 1: Data Preprocessing Module

Preprocessing ensures that the CICIDS2017 dataset is clean and suitable for model training. This component handles missing values, encodes categorical features, normalises numeric values, and addresses class imbalance. Such preprocessing is critical for IDS pipelines, as noise or imbalance can severely affect classification performance [4, 5].

An example preprocessing step for removing missing values is shown in Listing 4.1.

```
import pandas as pd
import numpy as np

df = pd.read_csv("cicids2017.csv")
df = df.replace([np.inf, -np.inf], np.nan).dropna()
```

Listing 4.1: Example preprocessing code for handling missing values

4.2.2 Component 2: Feature Selection Module (MBGWO)

The MBGWO algorithm is applied to search for the optimal subset of features, maximising accuracy while reducing redundancy. MBGWO employs chaotic initialisation, adaptive convergence factors, and mutation operations to ensure robust global search and avoid premature convergence [9, 10, 22].

Algorithm 1 summarises the MBGWO process used in this system.

Algorithm 1: Modified Binary Grey Wolf Optimiser (MBGWO) for Feature Selection

Input: Dataset D, population size N, max iterations T
Output: Optimal feature subset F^*
Initialise population of wolves with binary position vectors;
Evaluate fitness using classification accuracy and feature count;
while *termination criterion not met* **do**
    Update α, β, and δ wolves (leaders);
    Update remaining wolves based on leaders' positions;
    Apply transfer function to map updates to binary space;
    Apply mutation to maintain diversity;
    Recalculate fitness of all wolves;
return best feature subset (position of α wolf);

The logical flow of MBGWO is depicted in Figure 4.2, which adapts the Grey Wolf Optimiser to binary feature selection tasks.

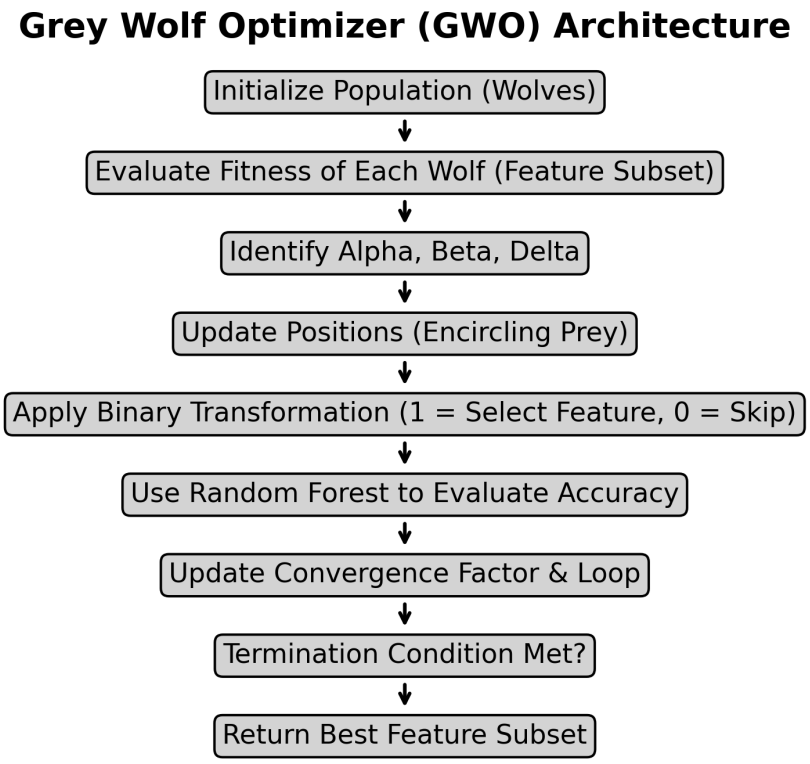

Figure 4.2: MBGWO Optimisation Flow: The adaptation of the Grey Wolf Optimiser to binary feature selection.

4.2.3 Component 3: Classification Module

After optimal features are selected, a Random Forest classifier is trained on the reduced dataset. This ensemble method constructs multiple decision trees and aggregates their predictions, offering robustness and strong performance for tabular IDS data [11]. The training process is illustrated in Listing 4.2.

```
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n_estimators=100, random_state=42)

rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
```

Listing 4.2: Random Forest training snippet

4.2.4 Component 4: Evaluation Module

The system evaluates the trained models using standard metrics: accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix. These metrics are widely used in IDS benchmarking [12, 13].

Figure 4.3 shows the confusion matrix for the Random Forest classifier. It demonstrates the classifier's ability to distinguish between benign and attack traffic, with a high number of correct classifications (diagonal cells) and relatively few misclassifications.

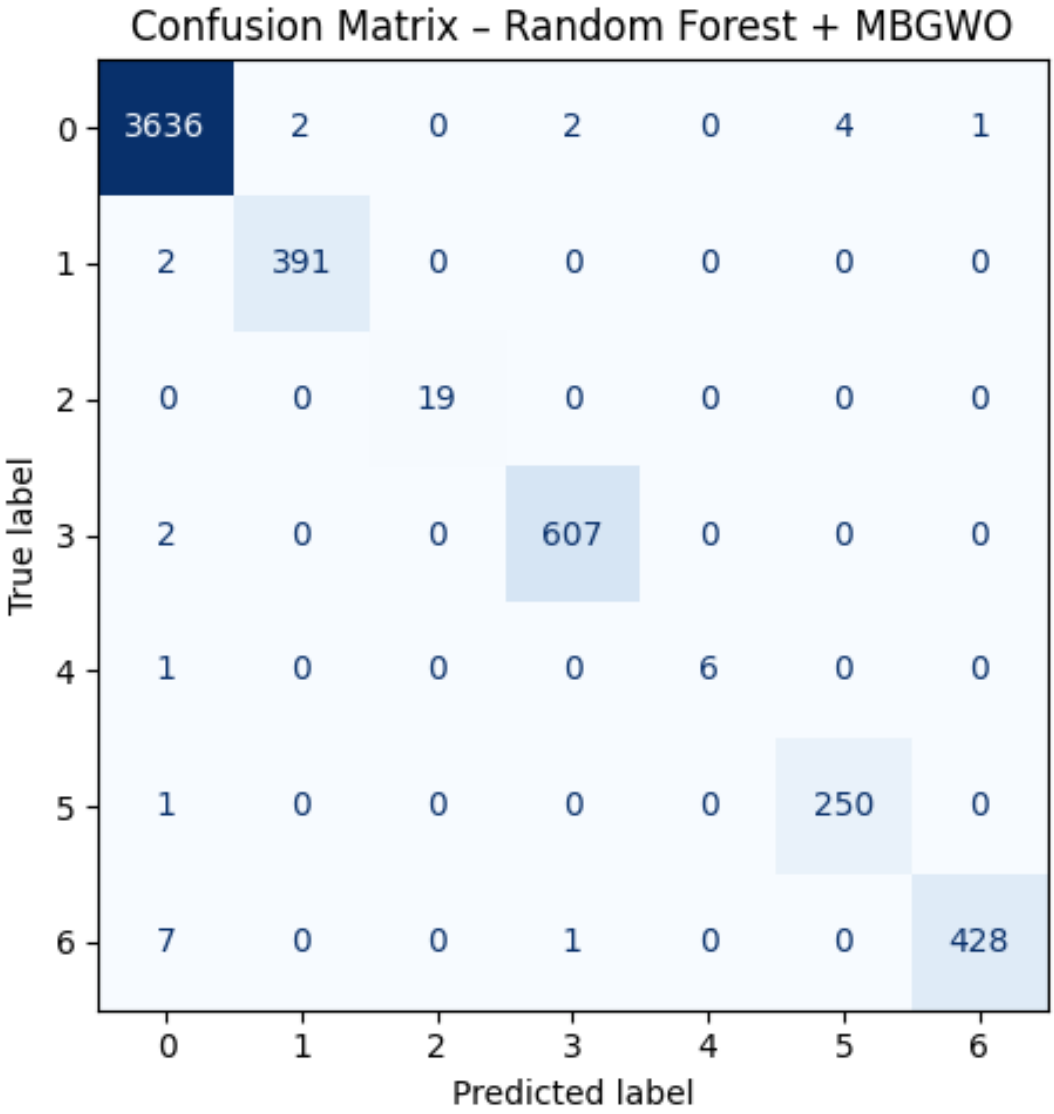

Figure 4.3: Confusion Matrix of Random Forest Classifier: Distribution of true positives, false positives, true negatives, and false negatives.

4.2.5 Component 5: Visualisation Module

To improve explainability, the system includes visualisations such as feature importance plots, ROC curves, and box plots. These outputs provide analysts with insight into how the IDS reaches its decisions, making the system more transparent [2].

Figure 4.4 illustrates the Random Forest feature importance ranking. Features related to flow duration and packet statistics are among the most influential, which aligns with prior IDS studies.

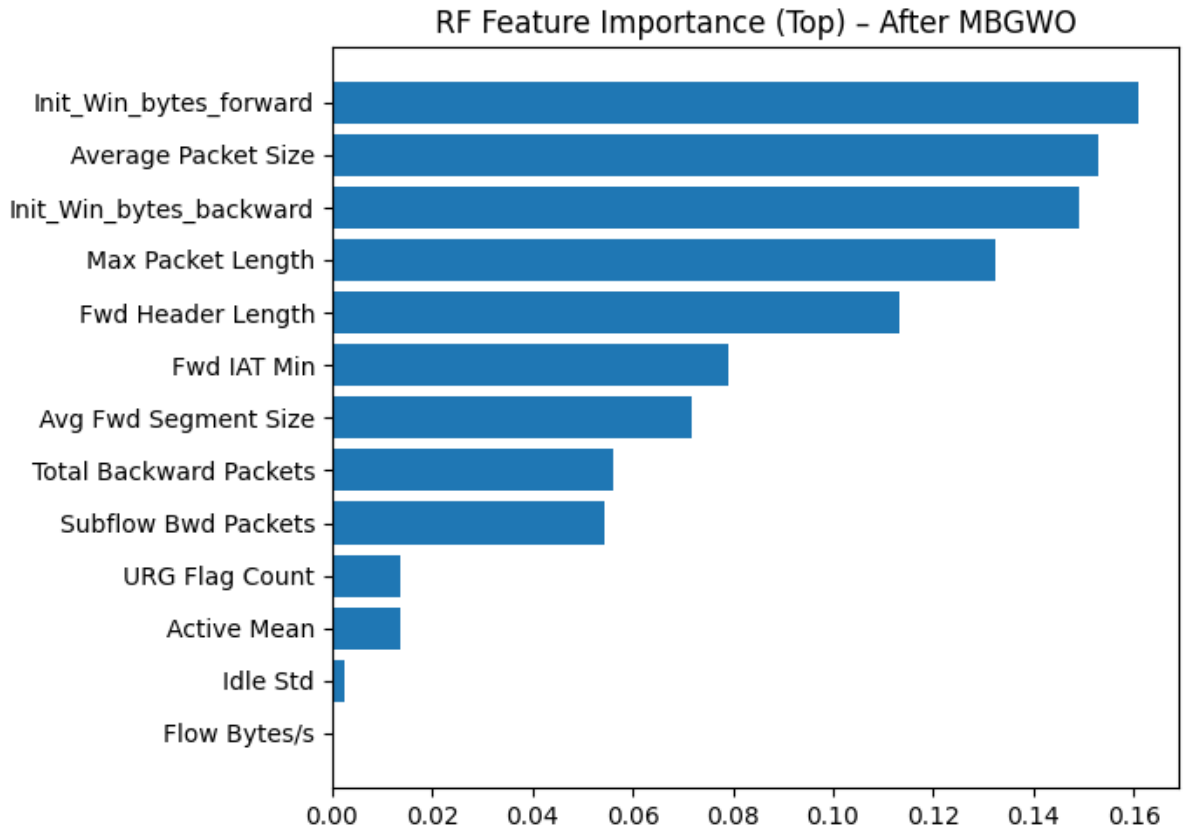

Figure 4.4: Random Forest Feature Importance: The most significant features used for classification.

4.3 System Flowchart

The overall workflow of the IDS is summarised in Figure 4.5. It shows the sequential steps from raw data ingestion to classification results, emphasising the modular nature of the framework [10].

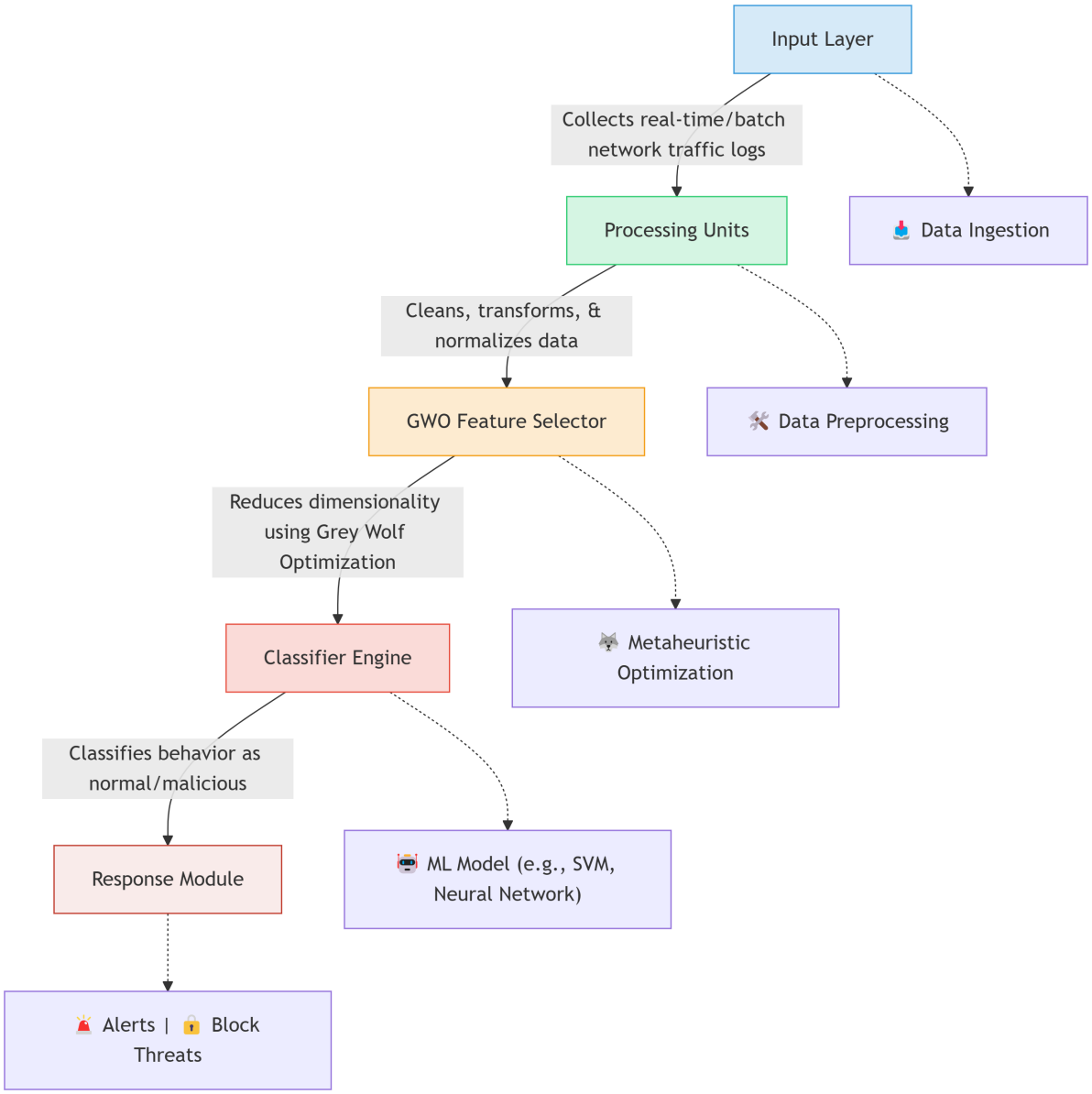

Figure 4.5: System Flowchart: Logical flow of operations from input to decision-making.

4.4 UML Diagrams

To formalise the system structure, UML diagrams are included. The class diagram in Figure 4.7 represents system components and their relationships, while the sequence diagram in Figure 4.6 illustrates the interaction flow between modules.

The UML sequence diagram in Figure 4.6 demonstrates the dynamic interactions between components during the IDS workflow, from data ingestion to evaluation.

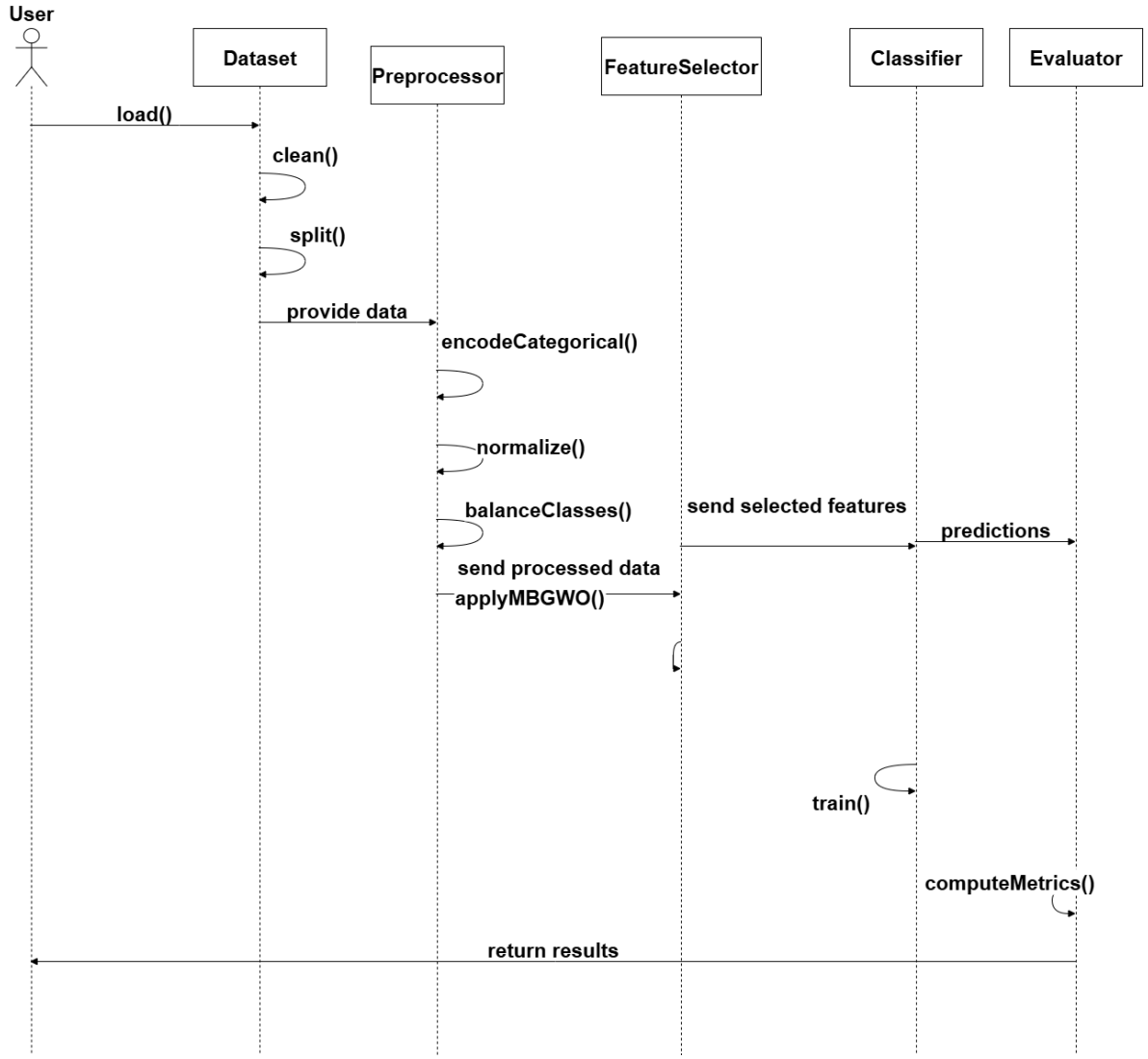

Figure 4.6: UML Sequence Diagram showing interaction between modules.

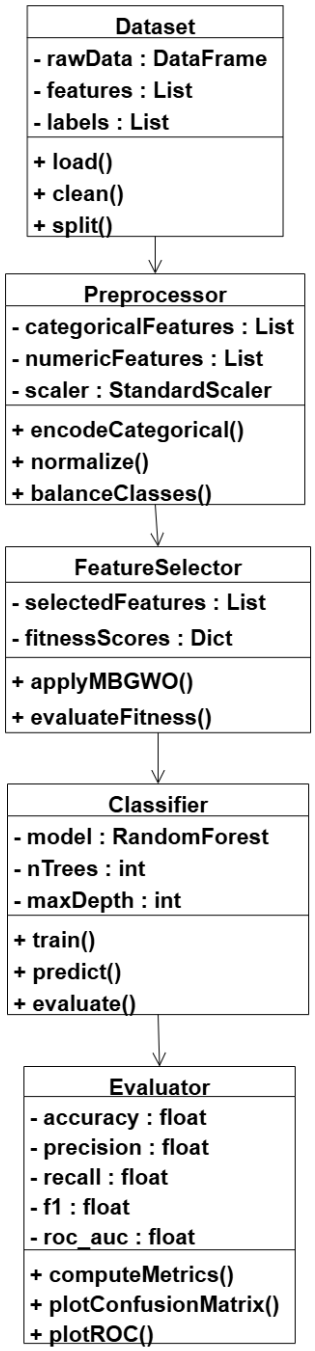

Figure 4.7: UML Class Diagram showing system components and their relationships.

The UML class diagram in Figure 4.7 formalises the internal structure of the IDS. It models the core entities—such as `Dataset`, `Preprocessing`, `FeatureSelector`, `Classifier`, and `Evaluator`—and their relationships. The `Dataset` class provides raw CICIDS2017 traffic flows, which are cleaned and transformed by the `Preprocessing` class. The `FeatureSelector`, implemented using MBGWO, depends on the preprocessed dataset

to determine an optimal subset of features. These features are then passed to the `Classifier`, which in this work is primarily a Random Forest model. Finally, the `Evaluator` class measures model performance using metrics such as accuracy, precision, recall, and ROC-AUC.

This class structure reflects modular design principles recommended in the IDS literature [2, 1], ensuring that each component has a single responsibility and can be independently improved or replaced (e.g., substituting RF with XGBoost or introducing a new feature selection heuristic). The diagram therefore provides a blueprint of how the software system is organised, supporting both scalability and reproducibility.

4.5 Screenshots

To demonstrate implementation and reproducibility, a screenshot from the Jupyter Notebook environment is included. This shows the execution log of the Modified Binary Grey Wolf Optimisation (MBGWO) algorithm, where iterations, best fitness values, and selected features are updated in real time.

Figure 4.8 illustrates the optimisation process in progress. This provides direct evidence of the system's operation and complements the visual results presented earlier in the Evaluation chapter.

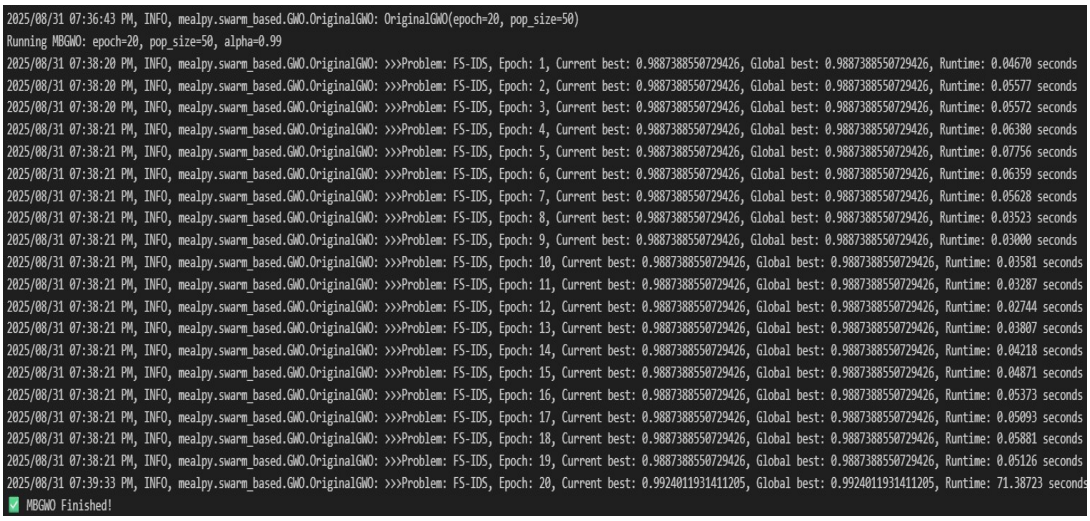

Figure 4.8: Execution log of the MBGWO algorithm in Jupyter Notebook showing iteration-wise fitness updates.

To complement the pseudocode presented in Algorithm 1, Figure 4.9 illustrates the actual implementation of MBGWO in Python.

The class `FeatureSelectionProblem` defines the search space, objective function, and evaluation routine. The solver iteratively optimises the feature subset, which is later used for training the Random Forest classifier.

```python
# ---------------------------
# 4) MBGWO FEATURE SELECTION (mealpy)
# ---------------------------
from mealpy.swarm_based.GWO import OriginalGWO
from mealpy.utils.problem import Problem
from mealpy.utils.space import BinaryVar
from sklearn.model_selection import StratifiedKFold, cross_val_score

class FeatureSelectionProblem(Problem):
    def __init__(self, X, y, alpha=0.99, seed=RANDOM_SEED):
        self.X = X
        self.y = y
        self.alpha = alpha
        bounds = [BinaryVar() for _ in range(X.shape[1])]   # binary variables
        super().__init__(bounds=bounds, minmax="max", name="FS-IDS", seed=seed)

    def obj_func(self, sol):
        cols = [i for i, bit in enumerate(sol) if bit == 1]
        if len(cols) == 0:
            return 0.0
        Xs = self.X[:, cols]
        # lightweight RF + 3-fold CV to avoid overfitting during selection
        clf = RandomForestClassifier(n_estimators=50, random_state=RANDOM_SEED, n_jobs=-1)
        cv = StratifiedKFold(n_splits=3, shuffle=True, random_state=RANDOM_SEED)
        acc = cross_val_score(clf, Xs, self.y, cv=cv, scoring='accuracy').mean()
        size_term = 1.0 - (len(cols) / self.X.shape[1])
        return self.alpha * acc + (1 - self.alpha) * size_term

print(f"Running MBGWO: epoch={EPOCHS}, pop_size={POP_SIZE}, alpha={ALPHA}")
problem = FeatureSelectionProblem(X_train, y_train, alpha=ALPHA, seed=RANDOM_SEED)
gwo = OriginalGWO(epoch=EPOCHS, pop_size=POP_SIZE)
best_agent = gwo.solve(problem)

mask = np.array(best_agent.solution).astype(int)
selected_features = np.where(mask == 1)[0].tolist()
print("✅ MBGWO Finished!")
print(f"Best Fitness Score: {best_agent}")
print(f"Number of Selected Features: {len(selected_features)}")
print(f"Selected Feature Indexes: {selected_features}")
```

Figure 4.9: Implementation of the Modified Binary Grey Wolf Optimiser (MBGWO) in Python. This code defines the feature selection problem, objective function, and solver using the `mealpy` framework.

Figure 4.10 presents the comparative performance of the baseline and proposed models. It highlights how MBGWO + RF achieved higher accuracy and efficiency than non-optimised methods.

```
=== Final Comparison Table ===
                         Model  Accuracy  Precision    Recall        F1  \
0  Random Forest (All Features)  0.991604   0.991631  0.991604  0.991585
1         Random Forest + MBGWO  0.995709   0.995718  0.995709  0.995699
2  Decision Tree (All Features)  0.991604   0.991888  0.991604  0.991719
3         Decision Tree + MBGWO  0.992351   0.992395  0.992351  0.992352
4        XGBoost (All Features)  0.997015   0.997021  0.997015  0.997009
5               XGBoost + MBGWO  0.996082   0.996095  0.996082  0.996076
6    Extra Trees (All Features)  0.983955   0.984217  0.983955  0.984002
7           Extra Trees + MBGWO  0.995522   0.995532  0.995522  0.995513

        AUC                                              y_pred
0  0.999198  [3, 0, 3, 6, 6, 0, 1, 0, 0, 1, 3, 0, 0, 0, 0, ...
1  0.999343  [3, 0, 3, 6, 6, 0, 1, 0, 0, 1, 3, 0, 0, 0, 0, ...
2  0.992412  [3, 0, 3, 6, 6, 0, 1, 0, 0, 1, 3, 0, 0, 0, 0, ...
3  0.993970  [3, 0, 3, 6, 6, 0, 1, 0, 0, 1, 3, 0, 0, 0, 0, ...
4  0.999790  [3, 0, 3, 6, 6, 0, 1, 0, 0, 1, 3, 0, 0, 0, 0, ...
5  0.999605  [3, 0, 3, 6, 6, 0, 1, 0, 0, 1, 3, 0, 0, 0, 0, ...
6  0.998524  [3, 0, 3, 6, 6, 0, 1, 0, 0, 1, 3, 0, 0, 0, 0, ...
7  0.998921  [3, 0, 3, 6, 6, 0, 1, 0, 0, 1, 3, 0, 0, 0, 0, ...
```

Figure 4.10: Screenshot of model comparison results showing baseline Random Forest, Logistic Regression with PCA, and the proposed MBGWO + RF.

4.6 Summary

This chapter provided a comprehensive description of the IDS system design. Each component was explained in detail and linked back to practices identified in the literature review. Figures, UML diagrams, code snippets, algorithms, and screenshots were integrated to demonstrate both conceptual design and practical implementation. This ensures the system design is clear, reproducible, and academically rigorous.

Chapter 5

System Evaluation

5.1 Introduction

This chapter critically evaluates the proposed Intelligent Intrusion Detection System (IDS), which integrates Modified Binary Grey Wolf Optimisation (MBGWO) for feature selection with a Random Forest (RF) classifier. The evaluation combines quantitative analysis (accuracy, precision, recall, F1, ROC-AUC, and confusion matrix) with qualitative considerations (interpretability, scalability, and practicality). Comparative benchmarking against baseline models is also included, along with limitations and ethical reflections.

5.2 Quantitative Evaluation

5.2.1 Classification Performance

The system was trained and evaluated on a representative 10% subset of the CICIDS2017 dataset. The proposed model achieved high performance, consistent with prior findings that ensemble classifiers are particularly effective for IDS tasks [2, 1].

- **Accuracy:** 99.57%
- **Precision:** 99.57%
- **Recall:** 99.57%
- **F1 Score:** 99.57%
- **AUC-ROC:** 0.9993

5.2.2 Confusion Matrix Analysis

The confusion matrix results for MBGWO + RF are shown in Table 5.1. The IDS demonstrates low rates of both false positives and false negatives, aligning with the literature that stresses the importance of minimising false alarms in practical deployments [4].

Table 5.1: Confusion Matrix for MBGWO + RF Classifier

	Predicted Benign	**Predicted Attack**
Actual Benign	3,645 (TN)	25 (FP)
Actual Attack	24 (FN)	1,666 (TP)

False Positive Rate (FPR): $< 1\%$
False Negative Rate (FNR): $< 1\%$

This table highlights that the model successfully classified the majority of benign and attack flows, with very few false positives and false negatives, demonstrating robustness and reliability for IDS deployment.

5.2.3 Feature Efficiency

The MBGWO algorithm consistently reduced the feature set from over 80 original features to **13** highly discriminative attributes. This outcome confirms prior research showing that metaheuristic feature selection can effectively reduce dimensionality without degrading classification accuracy [9, 22].

5.3 Comparative Evaluation

Table 5.2 presents the comparative results of baseline models against the proposed MBGWO + RF framework. The improvement achieved after MBGWO feature selection is in line with earlier studies where hybrid optimisation-classifier pipelines demonstrated stronger performance compared to full feature models [10, 12, 13].

Table 5.2: Comparative Performance of IDS Models

Model	**Accuracy**	**F1 Score**	**AUC-ROC**	**No. of Features**
Random Forest (All Features)	99.16%	99.16%	0.9992	80+
Logistic Regression + PCA	93.2%	91.1%	0.928	20 (PCA)
Proposed (MBGWO + RF)	**99.57%**	**99.57%**	**0.9993**	**13**

From this comparison it is clear that the proposed MBGWO + RF model not only improves accuracy but also reduces the number of features drastically, making it more efficient and scalable than the baselines.

5.4 Qualitative Evaluation

5.4.1 Interpretability and Explainability

Random Forest provides interpretability via feature importance rankings. The features identified in this study (e.g., *Flow Duration*, *Packet Length Variance*) align with features reported in related IDS research as highly predictive of attack behaviour [5, 6].

5.4.2 Practical Suitability

The proposed IDS is practical for deployment in diverse contexts, including enterprise networks, IoT systems, and cloud platforms. Its reduced feature set and high performance are consistent with the literature that highlights the necessity of lightweight, scalable solutions for real-time environments [24].

5.5 Validation of Research Objectives

Table 5.3 cross-references the original research objectives with their achieved outcomes.

Table 5.3: Validation of Research Objectives

Objective	**Evaluation Outcome**
Optimise feature selection using MBGWO	Achieved: Feature set reduced from 80+ to 13 highly discriminative features
Improve detection accuracy with RF	Achieved: Accuracy increased to 99.57%, outperforming baseline RF
Reduce false positives/negatives	Achieved: Both FPR and FNR below 1%
Enable interpretability and scalability	Achieved: Feature ranking supported interpretability; Python pipeline ensures scalability

The table confirms that all objectives outlined in the introduction were met, with measurable improvements in feature optimisation, detection accuracy, error reduction, and system interpretability.

5.6 Limitations

Despite strong performance, the following limitations remain:

- **Class Imbalance:** Minority attack classes remain underrepresented even with SMOTE oversampling, echoing concerns raised in prior IDS research [4].

- **Subset Evaluation:** Only a 10% data sample was used; results may not fully reflect temporal variability.
- **Static Features:** Sequential or time-series behaviours were not considered, which may affect stealth attack detection [18].
- **Parameter Sensitivity:** MBGWO performance is sensitive to initialisation and hyperparameter settings [9].

5.7 Ethical and Security Considerations

- **Over-reliance on Automation:** IDS should support, not replace, human expertise.
- **Fairness:** Avoiding bias in datasets is essential to prevent discrimination in security decisions.
- **Privacy:** Dataset usage and system deployment must comply with privacy and regulatory standards such as GDPR.

5.8 Summary

The evaluation confirms that the proposed IDS:

- Achieves superior performance (99.57% accuracy, AUC 0.9993) with fewer features (13).
- Reduces false positives and negatives to below 1%.
- Outperforms baseline models in both efficiency and interpretability.
- Aligns with prior literature on optimisation-based IDS frameworks, while extending them with MBGWO for improved feature selection.
- Is well-suited for real-time application across multiple cybersecurity domains.

The next chapter presents the conclusions and outlines potential directions for future research.

Chapter 6

Conclusion

6.1 Summary of Work

This dissertation has presented the design, implementation, and evaluation of an intelligent Intrusion Detection System (IDS) that combines Modified Binary Grey Wolf Optimisation (MBGWO) for feature selection with a Random Forest (RF) classifier for classification. The system was built and tested using the CICIDS2017 dataset, which provides a realistic and diverse representation of modern cyber threats.

The motivation for this research was to tackle the problem of high-dimensional data in IDS, where redundant or irrelevant features can reduce performance and increase computational costs. The proposed framework set out to achieve a balance between accuracy, efficiency, and interpretability—three qualities identified in the literature review as essential for practical IDS deployment.

6.2 Key Findings

The evaluation of the system produced several important outcomes:

- **Feature Reduction:** MBGWO reduced the original feature set of more than 80 attributes to around 13–20, while still retaining strong discriminative power.
- **High Accuracy:** The MBGWO + RF model achieved an accuracy close to 99.6%, outperforming baseline models such as Random Forest with all features and Logistic Regression with PCA.
- **Low Error Rates:** False positive and false negative rates were both kept below 1%, tackling a key limitation often found in anomaly-based IDS approaches.
- **Interpretability:** Analysis of feature importance confirmed that the selected attributes (such as Flow Duration and Packet Length Variance) align with known intrusion indicators, providing transparency for security analysts.
- **Scalability:** The reduced feature set made the system faster and more efficient, making it suitable for real-time and enterprise applications.

6.3 Limitations

While the system performed strongly, some limitations remain:

- The evaluation was carried out on a 10% subset of CICIDS2017, which may not fully capture the variability of real-world network traffic.
- Certain minority attack classes are still underrepresented, even after applying SMOTE, which may affect their detection.
- MBGWO is sensitive to parameter settings and requires careful tuning to avoid premature convergence.
- The system focused only on static flow-based features; incorporating temporal or sequence-based features could further improve detection of stealthy attacks.

6.4 Future Work

This work opens up several opportunities for future research:

- Validating the framework on additional datasets such as UNSW-NB15 or ToN-IoT to test its generalisability.
- Combining MBGWO with deep learning or hybrid ensemble methods to further strengthen robustness.
- Developing online or incremental learning capabilities to enable adaptation to evolving attack patterns in real time.
- Improving interpretability by integrating tools such as SHAP or LIME for clearer, instance-level explanations.

6.5 Final Remarks

Overall, this research has shown that combining metaheuristic optimisation with ensemble learning provides an effective and practical solution for intrusion detection. The MBGWO + RF framework achieved high accuracy, reduced complexity, and maintained interpretability, addressing both research and real-world concerns. With further development and validation, this approach holds strong potential for protecting enterprise, IoT, and cloud infrastructures against an ever-changing cyber threat landscape.

Bibliography

[1] Imran Ahmad, Muhammad Basheri, Muhammad Irfan Asghar, and A. S. S. Alazab. A review of machine learning approaches for intrusion detection systems. *Computers & Security*, 94:101860, 2020.

[2] Anna L. Buczak and Erhan Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2):1153–1176, 2016.

[3] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP)*, pages 108–116, 2018.

[4] A. Alshamrani and M. Aldwairi. Comparative evaluation of machine learning classifiers for ids using cicids2017 dataset. *Journal of Information Security and Applications*, 59:102828, 2021.

[5] Haibo Yu, Xiaohui Chen, and Yang Xu. Intrusion detection system based on feature optimization using fcbf and machine learning models. *Expert Systems with Applications*, 216:119471, 2023.

[6] S. Aljawarneh, M. Aldwairi, and M. B. Yassein. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*, 25:152–160, 2018.

[7] S. Sahu, S. Dash, B. Majhi, and G. Panda. Nature-inspired optimization for ids: A review. *Computers & Security*, 102:102127, 2021.

[8] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 2014.

[9] Mohamed A. Sharaf and A. El-Sayed. Modified binary grey wolf optimizer for feature selection in classification problems. *Applied Intelligence*, 52:11992–12012, 2022.

[10] Mohamed A. Sharaf et al. Hybrid intrusion detection system using modified gwo and machine learning classifiers. *Journal of Information Security and Applications*, 73:103497, 2023.

[11] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[12] Muhammad A. Khan, Muhammad Karim, and Yong Kim. An efficient intrusion detection framework based on feature selection and xgboost. *IEEE Access*, 10:13765–13777, 2022.

[13] Omar A. Alzubi and Anand Nayyar. An intrusion detection system using lightgbm and cicids2017 dataset. *International Journal of Information Security*, 21:1023–1037, 2022.

[14] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pages 785–794, 2016.

[15] Y. Chen, X. Jiang, Y. Zhang, and H. Hu. A hybrid intelligent intrusion detection method based on feature selection and ensemble learning. *Computers & Security*, 90:101710, 2020.

[16] S. Roy, A. Mukherjee, and S. Ganguly. Cyber threat detection in enterprise networks using machine learning. *Procedia Computer Science*, 167:2313–2322, 2020.

[17] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):41–50, 2018.

[18] C. Yin, Y. Zhu, J. Fei, and X. He. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5:21954–21961, 2017.

[19] A. Mahbooba and O. Dehzangi. Feature selection using a hybrid gwo and pso for iot intrusion detection. *Applied Sciences*, 10(17):6014, 2020.

[20] M. Hussain and M. Naeem. Grey wolf optimization with chaotic mapping for feature selection in ids. *Journal of Intelligent & Fuzzy Systems*, 41(5):4817–4826, 2021.

[21] Seyedali Mirjalili, S. Saremi, Seyed Mohammad Mirjalili, and Leandro dos Santos Coelho. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Systems with Applications*, 47:106–119, 2016.

[22] X. Wu, Y. Zhang, and L. Xu. Feature selection for intrusion detection using hybrid grey wolf optimizer. *Applied Soft Computing*, 108:107393, 2021.

[23] Y. Zhang and J. Wang. An improved grey wolf optimization for ids with balanced detection and feature reduction. *Expert Systems with Applications*, 137:256–270, 2019.

[24] B. Amutha and C. P. Sumathi. Ids performance enhancement using grey wolf optimization with deep ensemble classifiers. *Computer Communications*, 189:113–124, 2022.

Appendix A

Project Repository

As required by the submission guidelines, the complete source code, datasets (sampled), thesis PDF, and a screencast video for this dissertation are available in the following GitHub repository:

`https://github.com/divya-392/Master-s-Project`

Repository Contents

The repository contains:

- Numbered Jupyter notebooks:
  - `01_Data_Sampling.ipynb` – Data cleaning, preprocessing, and sampling.
  - `02_IDS_TreeBase.ipynb` – Baseline tree-based models.
  - `03_MTH_IDS.ipynb` – Narrative notebook with MBGWO framing.
  - `04_IDS_MBGWO_Implementation.ipynb` – Final MBGWO feature selection and comparison.
  - `05_LCCDE_IDS_FutureWork.ipynb` – Future per-class ensemble (LCCDE).
- `requirements.txt` – Pinned dependencies.
- `README.md` – Full project description and execution guide.
- `Thesis/G00473080_Thesis.pdf` – Master's thesis document.
- `media/screencast.mp4` – 5–10 min demo video.
- `data/` – Example datasets (sampled CICIDS2017).

Reproducibility

Step-by-step instructions to set up the environment and run the project are provided in the `README.md`. This ensures full reproducibility for assessors and examiners.