2025/2026

Report Case Study

Artificial Intelligence for Security

Alessandro Boffolo Aldo - 841260

INDEX

1 INTRODUCTION

This work aims to develop, evaluate and compare three different machine learning models for the identification and classification of DDoS attacks based on network traffic characteristics. The models considered are:

- **Random Forest Classifier:** chosen for its robustness with respect to outliers and non-linear distributions, as well as for its ability to handle features with very different scales. The model is optimised through grid search of hyperparameters as a criterion for node subdivision, maximum number of features and fraction of samples used for each tree.
- **Multi-Layer Perceptron (MLP) with Grid Search:** a classic fully connected neural network, optimised through a grid search on learning rate, number of epochs and batch size, with MinMax preprocessing of the data. This model allows the optimal configuration to be identified in order to minimise validation loss and maximise predictive accuracy.
- **Ensemble of fully connected neural networks (Bagging Neural Network):** an approach based on multiple identical neural networks trained on bootstrap samples from the dataset, with prediction aggregation via majority voting. This model aims to reduce variance and improve the overall stability of predictions by learning complex patterns present in DDoS traffic.

The ultimate goal is to identify the most effective and reliable model for multi-class classification of DDoS attacks, highlighting the strengths and limitations of each approach in relation to the nature and complexity of the dataset.

Link GitHub: GitHub

Canva: Canva

2 RANDOM FOREST CLASSIFIER

2.1 DATASETS

The training set, contained in the trainDdosLabelNumeric.csv file, consists of 10,000 network traffic samples and has an initial structure of 10,000 rows and 79 columns. Of these, 78 columns represent descriptive features, while one column is dedicated to the class variable (label).

The test set, provided in the testDdosLabelNumeric.csv file, consists of 1,000 samples used for the final evaluation of the model and has a structure of 1,000 rows and 79 columns.

2.2 EXPLORATORY DATA ANALYSIS

The first step in the data pre-processing process involved a thorough descriptive statistical analysis of the entire dataset. For each variable, the main summary statistics were calculated, including mean, standard deviation, minimum and maximum values, and quartiles, in order to understand the distribution of the data and identify any anomalies. This phase also made it possible to identify columns characterised by zero variance or constant values, which were subsequently removed as they lacked informative value.

Analysis of the output of the describe() function highlighted several critical characteristics of the dataset that require particular attention, especially in relation to the presence of extreme values, differences in scale between features, and the strong asymmetry of distributions.

2.2.1 Presence of outliers

Numerous features show statistical evidence of extreme outliers, which can be identified by comparing the maximum value with the 75th percentile. Among the most significant cases are:

- Flow_Bytes, with an average of 278,480,700 and a standard deviation of 458,824,500, has a maximum value of 2,944,000,000 compared to a 75th percentile of 458,000,000. The Max/Q3 ratio of 6.4 indicates the presence of exceptionally high values, close to 3 billion bytes.
- Flow_Packets shows an average of 876,205 packets and a maximum value of 3,000,000, significantly higher than the 75th percentile of 2,000,000, highlighting the presence of flows with an extremely high number of packets.
- Fwd IAT Mean has an average of 1,488,125 and a standard deviation of 60,237,320, about 40 times higher than the average. The maximum value reaches 3,604,371,000, suggesting a strongly skewed distribution and the possible presence of anomalies attributable to specific DDoS attacks.
- Bwd IAT Mean shows similar behaviour, with an average of 398,170, a standard deviation of 14,401,150 (36 times the average) and a maximum value of 1,276,447,000.
- Packet Length Mean has an average of 314.81 and a standard deviation of 4.009.22, with a maximum value of 312,375, almost 1,000 times higher than the average, constituting an extreme outlier.
- Avg Bwd Segment Size shows an average of 1,052, a standard deviation of 29,569 (28 times the average) and a maximum value of 1,560,784, indicating a highly skewed distribution.

The presence of these outliers is consistent with the nature of DDoS traffic, characterised by sudden bursts of packets and flooding phenomena. These characteristics make it necessary to use models that are robust to outliers, such as Random Forest, while other algorithms may require normalisation or logarithmic transformations.

2.2.2 Significant differences in attribute scales

Another notable feature of the dataset concerns the marked differences in scale between features. Variables can be divided into very different size classes: they range from binary features, such as Fwd PSH Flags and SYN Flag Count with values between 0 and 1, to variables with values exceeding billions, such as Flow_Bytes, Flow_Packets and Fwd IAT Mean.

For example, a comparison between flag-type features (range [0,1]) and Flow_Bytes (range [0, 2,944,000,000]) shows a scale ratio of approximately 3,000,000,000:1. Similarly, packet lengths, which reach a maximum of 3,617 bytes, are much smaller than packet inter-arrival times, such as Flow IAT Max, which can reach 82,012,200 microseconds, with a scale ratio of approximately 22,700:1.

Particularly critical is the presence of anomalous negative values in some features related to header lengths. Specifically:

- Fwd Header Length has a minimum value of -17,003,500,000 and a maximum of 28,784;
- Bwd Header Length varies from -2,125,430,000 to 49,392;
- min_seg_size_forward shows a minimum of -1,062,719,000 and a maximum of 1,480.

These extreme negative values have no direct physical meaning for header lengths and can be attributed to overflow phenomena or errors in the feature extraction phase. However, their systematic presence and negative average suggest that they may still serve as discriminating indicators for the model, representing traffic or logging anomalies.

From a modelling point of view, these differences in scale do not pose a problem for Random Forest, which is invariant with respect to feature scale.

2.2.3 Highly skewed distributions (skewness)

The distribution analysis shows that many features are characterised by a strong right skew, with long tails and a significant distance between the mean and median. For example, Flow_Bytes has a median of 9,541,667 compared to a mean of 278,480,700, a difference of approximately 29 times. Similarly, Flow_Packets has a median of 46,875 and a mean of 876,205, while Fwd IAT Mean has a median of 1 and a mean of 1,488,125, with a difference of more than a million times.

In many features, the first two quartiles are zero, indicating that at least 50% of the values are null. For example, 50% of the Bwd Packet Length Max values are zero, while for the Bwd IAT and Idle statistics, 75% of the values are less than or equal to 1 or even zero. This behaviour suggests that many connections are highly unidirectional, with minimal or no backward traffic.

The coefficients of variation are extremely high: Fwd IAT Mean has a CV of 40.5 (4.050%), Bwd IAT Mean has a CV of 36.2 (3.620%) and Packet Length Mean has a CV of 12.7 (1.270%). Considering that CV values above 10 indicate extreme variability, these results confirm the high heterogeneity of the analysed traffic.

2.2.4 Zero-variance attributes

Statistical analysis also identified 12 features with zero variance, i.e. constants equal to zero for all samples. These include several protocol flags (Bwd PSH, Fwd/Bwd URG, FIN, PSH, ECE) and all statistics relating to bulk forward and backward traffic. These features are not present or relevant in the type of DDoS traffic captured and have therefore been removed.

2.2.5 Class distribution

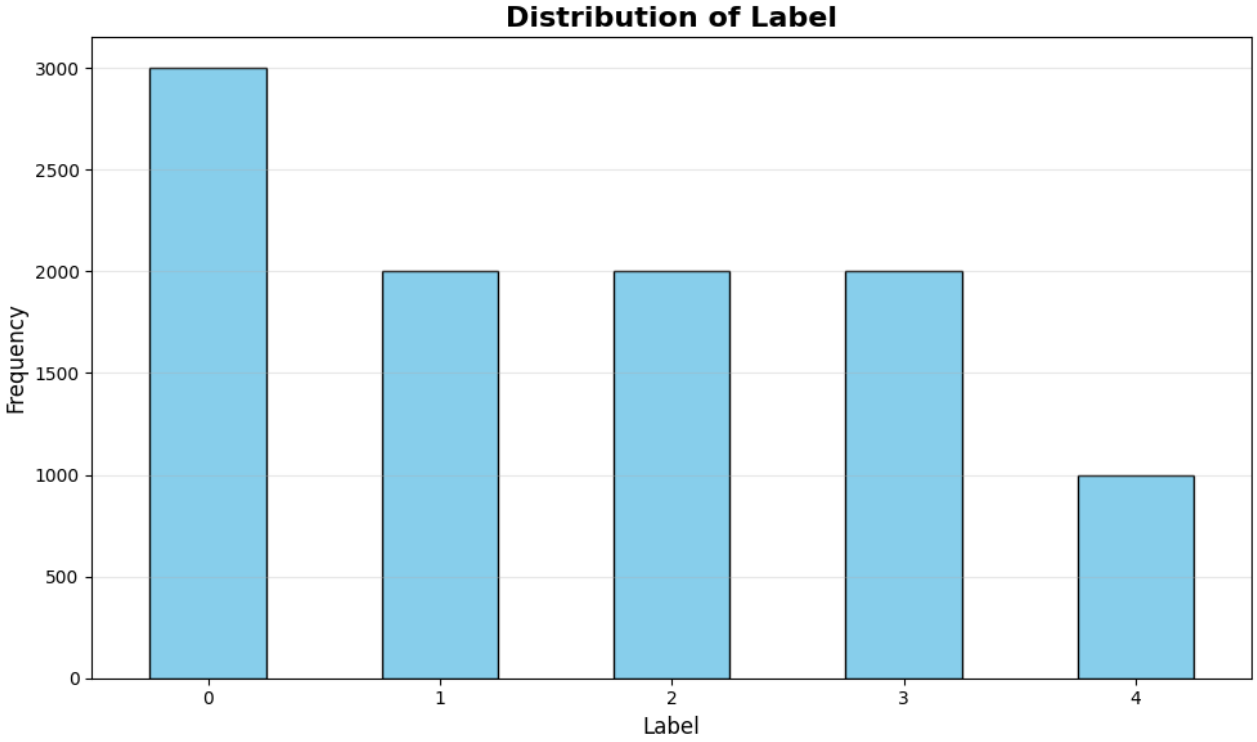


Figure 1 - Distribution of Label

Analysis of the target variable (Label) shows that the dataset is divided into five distinct classes, with a moderately unbalanced distribution overall. In the training set, class 0 is the most represented, with 3,000 samples, equal to 30% of the total, while classes 1, 2 and 3 each have 2,000 samples, corresponding to 20% each. Class 4 is the least frequent, with 1,000 samples, equal to 10% of the training set.

A similar distribution is maintained in the test set, which comprises a total of 1,000 samples, divided proportionally among the classes (300 for class 0, 200 for classes 1, 2 and 3, and 100 for class 4). Overall, although not perfectly balanced, the dataset has a limited imbalance, which allows the model to be trained and evaluated without necessarily resorting to aggressive rebalancing techniques, while requiring attention in evaluating performance on the less represented classes.

Class	Training Samples	Test Samples	Percentage Training
0	3000	300	30.0%
1	2000	200	20.0%
2	2000	200	20.0%
3	2000	200	20.0%
4	1000	100	10.0%

Table 1 - Class Distribution in Training and Test Sets

2.3 STRATIFIED K-FOLD CROSS VALIDATION

To ensure a reliable and robust evaluation of the model's performance, Stratified K-Fold Cross Validation was adopted. Specifically, the dataset was divided into 5 folds, using a random seed of 42 to ensure the reproducibility of the experiments.

The choice of a stratified strategy allows the same proportion of classes present in the original dataset to be preserved within each fold. This aspect is particularly relevant in the presence of class imbalance, as it prevents some folds from being poor or lacking certain labels. In this way, each training and validation iteration provides a more stable and representative estimate of the model's actual generalisation capabilities.

2.4 RANDOM FOREST CLASSIFIER

The model adopted for the classification phase is a Random Forest Classifier, chosen for its robustness with respect to outliers, non-linear distributions and the presence of features on very different scales. In order to identify the most effective configuration, a systematic exploration of different hyperparameters was conducted.

In particular, two main alternatives were considered for the node splitting criterion:

- **gini**, based on the Gini impurity measure
- **entropy**, which uses information gain

As for the number of features considered in each split, three different strategies were tested:

- **sqrt**, i.e. the square root of the total number of features
- **log2**, corresponding to the logarithm to base 2 of the number of features
- **None**, which involves the use of all available features

Finally, the effect of the max_samples parameter, which controls the fraction of samples used to train each individual tree in the forest, was analysed. Values between 0.5 and 1.0 were tested, specifically **[0.5, 0.6, 0.7, 0.8, 0.9, 1.0]**, allowing us to evaluate the trade-off between tree diversity, model robustness and generalisation ability.

2.5 GRID SEARCH CON CROSS VALIDATION

An exhaustive search (grid search) was conducted on all possible combinations of the hyperparameters defined above. A total of 36 configurations were evaluated, obtained by combining 2 criteria, 3 settings for the number of features, and 6 values for max_samples (2 × 3 × 6 = 36).

The evaluation metric chosen is the Macro F1 Score, which is particularly suitable for multi-class and moderately imbalanced datasets, as it assigns the same weight to each class regardless of its frequency. For each configuration, the model's performance was estimated as the average of the Macro F1 Score calculated on the 5 folds of the Stratified K-Fold Cross Validation, thus ensuring a stable and representative evaluation of the model's generalisation capacity.

3 MULTI-LAYER PERCEPTRON

3.1 METHODOLOGY

For the classification of DDoS attacks, an approach based on feedforward artificial neural networks was adopted, specifically a Multi Layer Perceptron (MLP). This methodological choice was motivated by the ability of MLP networks to learn complex and non-linear patterns present in network traffic, combining multiple pieces of information relating to temporal, volumetric and protocol characteristics.

3.1.1 Model architecture

The architecture of the MLP used consists of fully connected (densely connected) layers designed to capture the relationships between the 78 features of the dataset. The structure of the model is as follows:

- **Input Layer:** size equal to the number of features in the dataset (78).
- **Hidden Layer 1:** 128 neurons with ReLU activation function.
- **Hidden Layer 2:** 64 neurons with ReLU activation function.
- **Hidden Layer 3:** 32 neurons with ReLU activation function.
- **Output Layer:** 5 neurons, one for each class, with Softmax activation function, in order to produce a probability distribution on the output classes.

ReLU (Rectified Linear Unit) activation in the hidden layers allows the model to learn complex non-linear patterns, while the Softmax function in the output allows for multi-class classification.

3.2 DATASET AND PREPROCESSING

The dataset was normalised using a MinMaxScaler, transforming all features into a range [0,1]. This operation is necessary to ensure stable convergence during training and to reduce the impact of features with very different scales. The test set was transformed using the same scaler applied to the training set, avoiding contamination between training and testing.

The dataset has five classes, with a moderately unbalanced distribution. The division between training and testing was carried out in a stratified manner, preserving the proportion of classes in both sets.

3.3 GRID SEARCH ON HYPERPARAMETERS

To identify the optimal model configuration, a grid search was performed on combinations of key hyperparameters, evaluating the model based on validation loss:

- **Adam optimiser learning rate:** [0.0001, 0.001, 0.01, 0.1]
- **Number of epochs:** [30, 60, 90, 120, 150]
- **Batch size:** [16, 32, 64, 128]

Training was performed with Early Stopping on validation loss, with a patience of 10 epochs, to prevent overfitting.

3.4 TRAINING AND VALIDATION

The training set was divided into a training set (80%) and a validation set (20%), maintaining the stratification of the classes. For each combination of hyperparameters, the model was trained on the training set and validated on the validation set. The configuration with minimum validation loss was selected as the best and saved for final evaluation on the test set.

4 BAGGING NEURAL NETWORK

4.1 METHODOLOGY

An Ensemble Learning approach was adopted for the classification of DDoS attacks, specifically the Bagging (Bootstrap Aggregating) technique applied to artificial neural networks. This methodological choice was motivated by bagging's ability to reduce variance, improve prediction stability and increase the overall robustness of the model by combining the decisions of multiple classifiers trained on different subsets of the dataset.

The ensemble consists of multiple fully connected neural networks, all characterised by the same architecture, designed to learn complex patterns present in network traffic. The architecture of each model is as follows:

- **Input Layer:** size equal to the number of features in the dataset (78).
- **Hidden Layer 1:** 128 neurons with ReLU activation function.
- **Hidden Layer 2:** 64 neurons with ReLU activation function.
- **Hidden Layer 3:** 32 neurons with ReLU activation function.
- **Output Layer:** 5 neurons, one for each class, with Softmax activation function, in order to produce a probability distribution on the output classes.

4.2 GRID SEARCH ON BOOTSTRAP SAMPLES

The ensemble training process was structured through the generation of bootstrap samples and the optimisation of hyperparameters via grid search.

Specifically, 10 independent bootstrap samples were generated, each obtained by extracting 80% of the original training set with replacement, corresponding to 8,000 samples. Each bootstrap sample was used to train a single model of the ensemble.

For each bootstrap, an exhaustive search of the hyperparameters was performed, dividing the sample into:

- **Training set (80%)**
- **Validation set (20%)**

The hyperparameters considered in the grid search were:

- **Patience for Early Stopping:** {5, 10, 15}
- **Learning Rate of the Adam optimiser:** {0.0001, 0.001, 0.01, 0.1}
- **Number of training epochs:** {10, 20, 50, 100}

A total of 48 configurations were evaluated for each ensemble model. The optimal configuration was selected by minimising the Validation Loss, in order to ensure a good compromise between generalisation capacity and performance.

4.3 Aggregation of Predictions through Majority Voting

Once the 10 ensemble models with the best hyperparameter configuration identified for each bootstrap were trained, the predictions on the test set were combined using a **Majority Voting** mechanism.

In this approach, each ensemble model expresses a class prediction for each test sample; the final class assigned is the one that receives the most votes among the models. This strategy reduces the impact of any errors made by individual classifiers, improving the accuracy and reliability of the final classification.

5 RESULTS

5.1 RANDOM FOREST CLASSIFIER

After training and evaluating the model, the results show excellent and near-perfect performance in classifying DDoS attacks.

5.1.1 Best Configuration

The optimal configuration identified through grid search is:

- Criterion: **gini**
- Max Features: **log2**
- Max Samples: **0.5**
- Average Macro F1 (CV): 0.9938

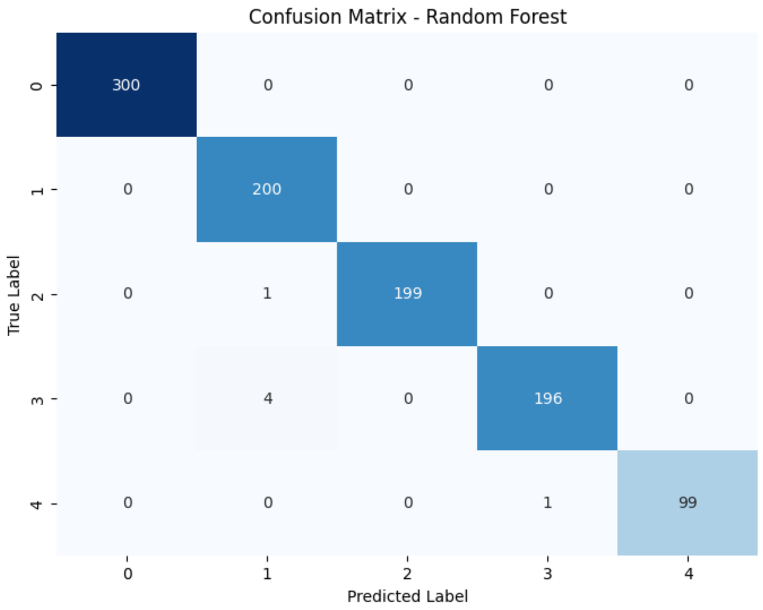
This combination guaranteed the best average performance calculated on the 5 folds of Stratified K-Fold Cross Validation.

5.1.2 Test Set Performance

When evaluating the model trained on the independent test set (1,000 samples), the main metrics are:

- Test Macro F1 Score: 0.9935
- Test Accuracy: 0.99 (99%)
- Training-Test Difference: 0.0003 (0.03%)
- Total errors: 6 out of 1,000 samples

The confusion matrix highlights the details of the errors:



	0	1	2	3	4
0	300	0	0	0	0
1	0	200	0	0	0
2	0	1	199	0	0
3	0	4	0	196	0
4	0	0	0	1	99

Figure 2 - Confusion Matrix Random Forest

5.1.2.1 *Error analysis*

- Class 2: 1 sample classified as class 1
- Class 3: 4 samples classified as class 1
- Class 4: 1 sample classified as class 3
- Classes 0 and 1: no errors, perfect classification

5.1.2.2 *Classification Report*

Class	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	300
1	0.98	1.00	0.99	200
2	1.00	0.99	1.00	200
3	0.99	0.98	0.99	200
4	1.00	0.99	0.99	100
Macro Avg	**0.99**	**0.99**	**0.99**	**1000**
Weighted Avg	**0.99**	**0.99**	**0.99**	**1000**
Accuracy			**0.99**	**1000**

Table 2 - Classification Report Random Forest

5.1.2.3 *Key observations*

- The majority class (0.30%) was classified perfectly (100% precision and recall).
- The minority classes also performed very well (class 4: F1=0.99).
- Only a few marginal errors in classes 2, 3 and 4, confirming the robustness of the model.

5.2 Multi-Layer Perceptron

After training and evaluating the Multi-Layer Perceptron (MLP) model, the results demonstrate excellent performance in the multi-class classification of DDoS traffic. The neural network shows strong generalization capabilities and a very limited gap between training and test performance, confirming the effectiveness of the selected architecture and hyperparameter tuning strategy.

5.2.1 Best Configuration

The optimal configuration was identified through an exhaustive grid search based on the minimization of the validation loss. The best-performing model uses the following parameters:

- Learning Rate: **0.01**
- Batch Size: **128**
- Maximum Epochs: **150**

This configuration achieved the lowest validation loss among all tested combinations, ensuring optimal convergence and reduced overfitting.

5.2.2 Test Set Performance

The selected MLP model was evaluated on an independent test set composed of 1,000 samples, maintaining the same class distribution as the training set. The main evaluation metrics are reported below:

- Test Accuracy: 0.99 (99%)
- Test Macro F1 Score: 0.99
- Training Accuracy: 0.98
- Training–Test Difference: ≈ 0.01
- Total Errors: 8 out of 1,000 samples

The confusion matrix provides a detailed overview of the classification errors.

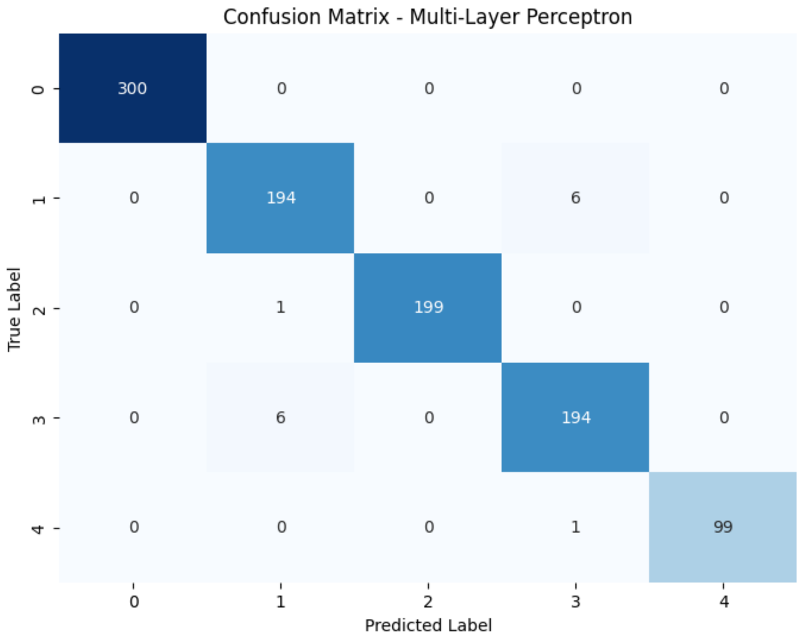


Figure 3 - Confusion Matrix Multi-Layer Perceptron

5.2.2.1 Errror Analysis

The error distribution across classes is extremely limited and mainly affects neighboring classes:

- Class 1: 6 samples misclassified as class 3
- Class 2: 1 sample misclassified as class 1
- Class 3: 6 samples misclassified as class 1
- Class 4: 1 sample misclassified as class 3
- Class 0: No errors, perfect classification

The misclassifications are marginal and mostly occur between classes with similar traffic patterns, which is common in network intrusion datasets.

5.2.2.2 Classification Report

Class	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	300
1	0.97	0.97	0.97	200
2	1.00	0.99	1.00	200
3	0.97	0.97	0.97	200
4	1.00	0.99	0.99	100
Macro Avg	**0.99**	**0.98**	**0.99**	**1000**
Weighted Avg	**0.99**	**0.99**	**0.99**	**1000**
Accuracy			**0.99**	**1000**

Table 3 - Classification Report Multi-Layer Perceptron

5.2.2.3 Key Observations

- The majority class (class 0) is classified perfectly, with 100% precision and recall.
- The minority class (class 4) achieves excellent performance, with an F1-score of 0.99, indicating strong robustness even in less represented classes.
- The overall error rate is very low and concentrated in a few samples, mainly between classes with similar behavioral characteristics.
- The small gap between training and test performance confirms the good generalization capability of the MLP model and the effectiveness of early stopping and hyperparameter tuning.

5.3 Bagging Neural Network

The training and evaluation of the neural network ensemble produced extremely high performance, demonstrating the effectiveness of the approach.

5.3.1 Best Configuration

Analysis of the grid search results on the 10 bootstraps showed a clear trend towards hyperparameters that favour stable and in-depth learning:

- Learning Rate: Most models performed best with a learning rate of **0.001**.
- Patience: The best-performing patience values were **10 and 15**, indicating that the models benefited from longer training before early termination.
- Epochs: The maximum number of epochs, **100**, was almost always selected, confirming the need for extended training to achieve optimal convergence.

5.3.2 Test Set Performance

The performance of the individual models, evaluated on the test set prior to aggregation, showed consistently high accuracy, as reported in the following table:

Model	Accuracy
1	0.9870
2	0.9850
3	0.9880
4	0.9830
5	0.9880
6	0.9850
7	0.9850
8	0.9840
9	0.9830
10	0.9860
Average	**0.9854**

Table 4 - Classification Report Random Forest

Evaluating the final ensemble on the independent test set (1,000 samples), the aggregate metrics are:

- Test Macro F1 Score: 0.99
- Test Accuracy: 0.986 (98.6%)
- Improvement over individual average: +0.0006 (0.06%)
- Total errors: 14 out of 1,000 samples

The minimal difference between the individual average performances and those of the ensemble, combined with the slight improvement, confirms the stability and reliability of the final model.

The confusion matrix highlights the details of the errors:

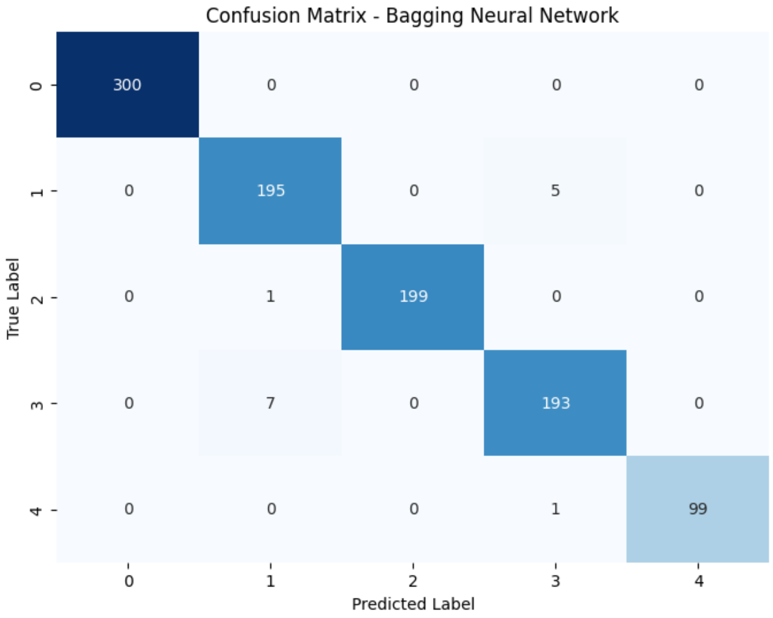


Figure 4 - Confusion Matrix Bagging Neural Network

5.3.2.1 Error analysis

- Class 1: 5 samples classified as class 3.
- Class 2: 1 sample classified as class 1.
- Class 3: 7 samples classified as class 1.
- Class 4: 1 sample classified as class 3.
- Class 0: No errors, perfect classification.

5.3.2.2 Classification Report

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	1.00	1.00	1.00	300
1	0.96	0.97	0.97	200
2	0.99	0.99	0.99	200
3	0.96	0.96	0.96	200
4	0.99	0.99	0.99	100
Macro Avg	**0.98**	**0.98**	**0.98**	**1000**
Weighted Avg	**0.98**	**0.98**	**0.98**	**1000**
Accuracy			**0.98**	**1000**

Table 5 - Classification Report Bagging Neural Network

5.3.2.3 Key observations

The results obtained show extremely high performance across all classes considered. In particular, the majority class (class 0), which represents approximately 30% of the dataset, was classified without any errors, achieving precision and recall values of 100%. Even minority classes, such as class 4, which constitutes approximately 10% of the samples, showed excellent performance, with an F1-score of 0.99. This result suggests that the model is not significantly affected by a moderate imbalance between classes. Overall, the number of classification errors is extremely low and is mainly attributable to a slight confusion between classes 1 and 3, further confirming the high robustness and accuracy of the ensemble approach adopted.

6 DISCUSSION OF RESULTS AND CONCLUSION

The aim of this study was to develop, evaluate and compare three machine learning models for the multi-class classification of DDoS attacks, based on network traffic characteristics. All the models analysed demonstrated exceptionally high performance. This highlights the good quality and separability of the classes present in the provided dataset. However, a detailed analysis of the quantitative performance and qualitative characteristics of each approach identified a clear winner.

6.1 COMPARATIVE PERFORMANCE SUMMARY

To enable direct comparison, we summarise the key performance metrics obtained by each model on an independent test set comprising 1,000 samples.

Metric	**Random Forest Classifier**	**Multi-Layer Perceptron**	**Bagging Neural Network**
Accuracy	0.99	0.99	0.986
Macro F1-Score	0.9935	0.99	0.98
Total Number of Errors	6	8	14

Table 6 - Comparative Performance

The following observations emerge from the analysis of the table:

- Peak performance: The Random Forest Classifier and the Multi-Layer Perceptron achieve an identical accuracy of 99.0%. However, the Random Forest Classifier achieves a marginally higher macro F1 score (0.9935 vs 0.99), indicating a slightly better balance in the classification of all classes, including minority ones.
- Number of errors: Random Forest is the model that makes the fewest errors overall (only six out of 1,000). The MLP model follows closely behind with eight errors, while the neural network ensemble (Bagging NN) makes more than twice as many errors (14).
- Bagging stability: Interestingly, the Bagging approach, which is designed to reduce variance and improve robustness, did not lead to improved performance over the single MLP in this case. In fact, the aggregate accuracy (98.6%) was lower than that of the best-configured single MLP (99.0%) and lower than the average of the individual models in the ensemble (98.54%). This suggests that the single MLP models were already very stable for this dataset and that variance was not the main problem. Aggregation via majority voting failed to correct the base models' systematic errors.

6.2 QUALITATIVE ANALYSIS AND MODEL COMPLEXITY

In addition to quantitative metrics, qualitative and practical aspects must also be considered.

- Random Forest Classifier: This model has several key advantages. Firstly, as the exploratory analysis showed, no data normalisation or scaling is required. Its decision tree-based nature makes it robust to outliers and extreme differences in scale between features, both of which are prominent characteristics of this dataset. Implementing it and tuning its hyperparameters is relatively simple and computationally efficient.

- Multi-Layer Perceptron (MLP): In order to function properly, MLP requires a mandatory pre-processing step (MinMax normalisation). Although it achieved excellent performance, its 'black box' nature makes it less interpretable than a Random Forest. Furthermore, training a neural network is generally more sensitive to the choice of hyperparameters (learning rate, batch size and number of epochs) and requires more computational resources than an RFC.
- Bagging Neural Network: This was by far the most complex and computationally expensive approach. It required the training of ten separate neural networks, each with its own grid search phase for hyperparameter optimisation. Despite this additional complexity, it failed to offer any performance advantage and produced inferior results. This violates the principle of parsimony (Occam's razor), which states that, all things being equal, the simplest solution should be preferred.

6.3 Best Model

In light of the comparative analysis, the Random Forest Classifier model is the optimal choice for solving this classification problem. The reasons are as follows:

- Quantitative superiority: Although the differences are minimal, the Random Forest model achieved the best performance metrics across the board, with the highest macro F1 score and the lowest absolute number of classification errors. This makes it the most accurate and reliable model among those tested.
- Intrinsic robustness and simplicity of pre-processing: The model's ability to handle non-normalised data with outliers and asymmetric distributions natively is a huge practical advantage. It simplifies the machine learning pipeline, reduces the risk of errors introduced during pre-processing (e.g. data leakage) and facilitates implementation in a production environment.
- Computational efficiency: Compared to neural network-based approaches, particularly ensembles, Random Forest is significantly faster to train. In a cybersecurity context, where models may need to be retrained frequently on new data, efficiency is critical.
- Application of the principle of parsimony: Random Forest achieved the best results with the structurally simplest and least data-intensive model. The increase in complexity introduced by neural networks did not lead to a corresponding increase in performance, making them less justifiable.

6.4 Conclusion

In conclusion, this study successfully demonstrated that standard machine learning approaches can be used to classify different types of DDoS attack with extremely high accuracy. All three models achieved an accuracy rate of over 98.5%, confirming the effectiveness of these techniques for identifying anomalous traffic.

However, the Random Forest Classifier was the clear winner, not only due to its slightly better quantitative performance, but also thanks to its robustness, operational simplicity and computational efficiency. Its ability to perform well without the need for complex pre-processing steps makes it the ideal choice for real-world network security applications, where reliability, speed, and ease of maintenance are as important as predictive accuracy.