UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

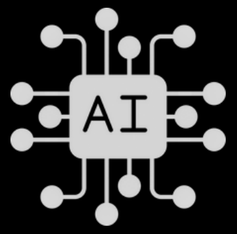
KDDE

Case Study

Comparison of Models for Traffic Classification

Artificial Intelligence For Security

Prof:
Annalisa Appice
Giuseppina Andresini



Students:
Nicola Balzano

Introduction

Scope of the study

The **primary goal** of this study is the evaluation and comparison of three different machine learning architectures to determine the most effective approach for Network Traffic Classification.

The study systematically analyzes **three distinct models** to evaluate their performance in identifying network protocols and behaviors:

- **Random Forest**: An ensemble learning method evaluated for its efficiency and classification accuracy.
- **Baseline Multilayer Perceptron** (MLP): A standard deep learning approach based on the original 66 network features.
- **K-Means Augmented Neural Network**: A hybrid model utilizing unsupervised spatial feature extraction to increase the input dimension to 89 features.

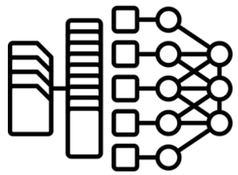
Dataset Info

Training and Test Dataset information & Pre-processing

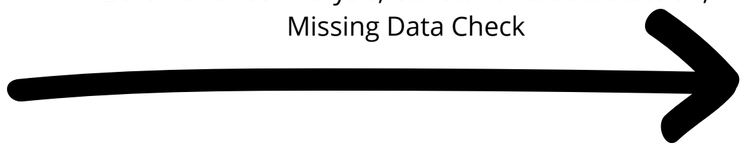
The analysis utilized two main data sources to validate generalization:

- **Training & Validation Set**: 80% for training and 20% for internal validation and hyperparameter selection.
- **External Test Set**: A separate dataset consisting of 1,000 unseen samples used for the final model evaluation.
- **Target Classes**: 5 distinct network traffic categories.

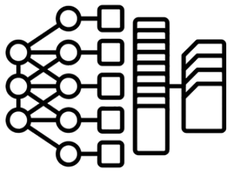
88 original features

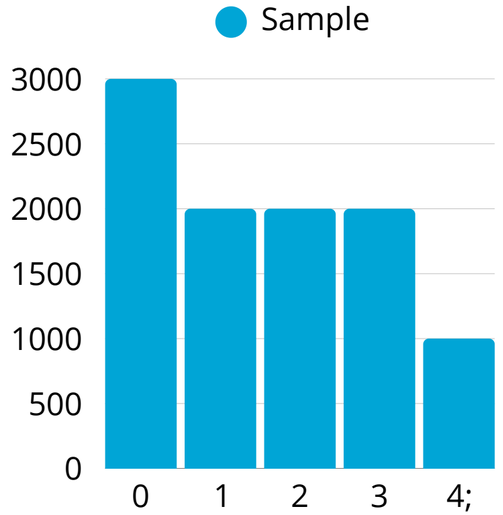


Zero Variance Analysis, Constant Value Detection, Missing Data Check



66 features





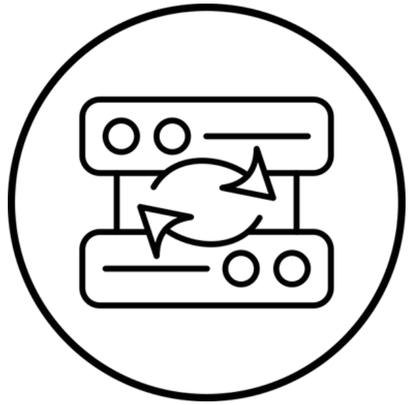
Random Forest

Methods

No Feature Scaling Required: the Random Forest algorithm was trained directly on the raw feature set.

Stratified 5-Fold Cross-Validation: The training dataset was partitioned into 5 distinct folds using StratifiedKFold.

Class Balance Preservation: Stratification ensured that each fold maintained the exact proportion of the 5 traffic classes (0 through 4), which is vital for correctly training the model on the minority class.

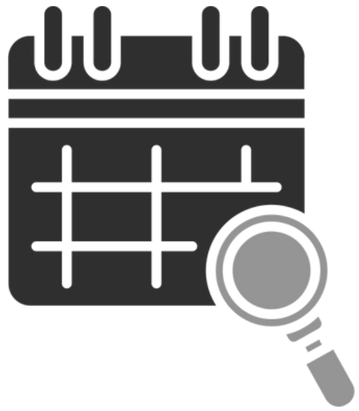


Random Forest

Parameters

To identify the optimal configuration, a systematic **Grid Search** was performed across the following hyperparameter space:

- **Criterion**: Evaluated ***Gini*** impurity vs. ***Entropy*** to determine the best split quality.
- **Max Features**: Tested ***sqrt***, ***log2***, and ***None*** (all features) to optimize the random feature subset size at each node.
- **Max Samples**: Explored bootstrapping fractions ***from 0.5 to 1.0*** to control the trade-off between bias and variance.

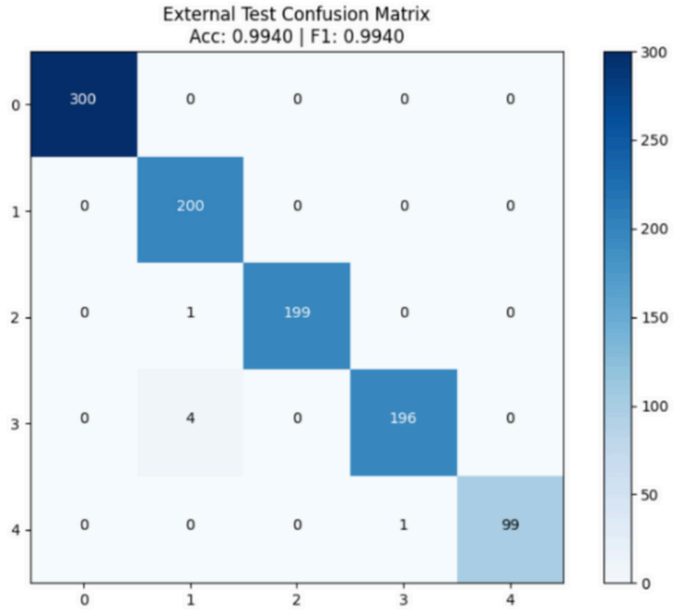


Random Forest

Result & Conclusion

The best combination is with Criterion=**gini**, Max Features=**log2**, Max Samples=**0.5**
The Random Forest classifier demonstrated slightly superior performance and higher stability. Its ability to achieve a **0.9940 accuracy** without the need for intensive feature scaling makes it a highly efficient solution for this specific network traffic classification task.

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

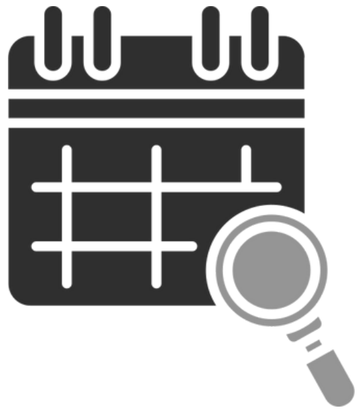


Multilayer Perceptron

Parameters

The tuning process explored the following hyperparameters to balance model complexity and generalization, with a total of 27 configuration:

- **Batch Size**: [32, 64, 128] — tested to optimize gradient stability and memory efficiency.
- **Learning Rate**: [0.0001, 0.001, 0.01] — evaluated to find the most precise convergence speed.
- **Dropout Rate**: [0.2, 0.3, 0.4] — utilized as a regularization technique to prevent overfitting.



Multilayer Perceptron

Training & Validation Result

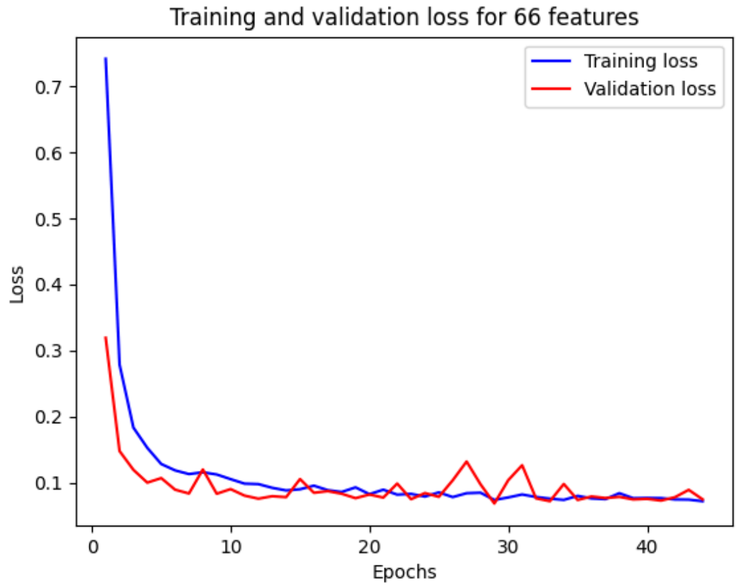
TOP 10 by Validation F1-score				
Batch size	**Learning Rate**	**Dropout**	**Validation Loss Min**	**Validation F1-score**
64	0.01	0.3	0.0977	0.988008
32	0.01	0.3	0.0919	0.985091
32	0.01	0.4	0.1233	0.983942
128	0.01	0.4	0.0926	0.983590
128	0.001	0.2	0.0646	0.982583
32	0.0001	0.3	0.0690	0.982086
128	0.001	0.4	0.0780	0.982084
64	0.001	0.2	0.0642	0.982082
64	0.001	0.3	0.0727	0.981084
64	0.0001	0.3	0.0739	0.981084

TOP 10 by Validation Loss				
Batch size	**Learning Rate**	**Dropout**	**Validation Loss Min**	**Validation F1-score**
64	0.001	0.2	0.0642	0.982082
128	0.001	0.2	0.0646	0.982583
32	0.0001	0.2	0.0649	0.981084
64	0.0001	0.2	0.0682	0.980583
32	0.001	0.2	0.0682	0.979074
128	0.001	0.3	0.0686	0.979077
32	0.0001	0.3	0.0690	0.982086
64	0.001	0.4	0.0724	0.981082
64	0.001	0.3	0.0727	0.981084
64	0.0001	0.3	0.0739	0.981084

Multilayer Perceptron

Training & Validation Result

The training process was halted at **epoch 55** after the validation loss failed to improve significantly for 15 consecutive epochs (patience).
The system restored the weights from the point of minimum **validation loss (0.0642)**, ensuring that the final model utilized its most generalized state before any onset of overfitting.

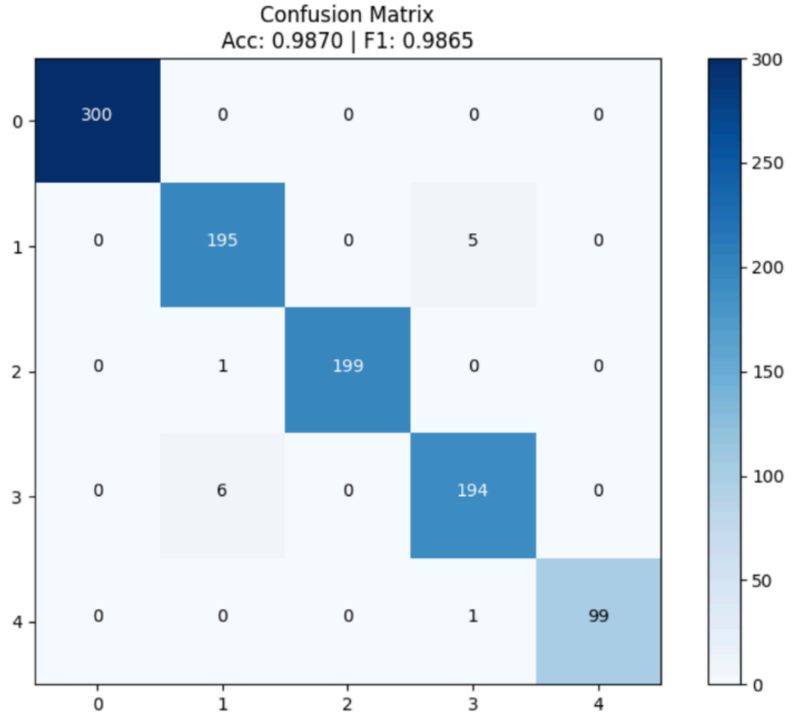


Multilayer Perceptron

Result & Conclusion

The high **98.7% Accuracy** and F1-Score on the external test set confirm that the three-layer dense architecture was sufficient to capture the non-linear relationships present in the 66 network features.

Class	Precision	Recall	F1-score	Analysis
0	1.00	1.00	1.00	No errors recorded for this class.
1	0.97	0.97	0.97	6 samples were misclassified as Class 3.
2	1.00	0.99	1.00	Only 1 sample misclassified.
3	0.97	0.97	0.97	6 samples were misclassified as Class 1.
4	1.00	0.99	0.99	99% recall on the smallest class.

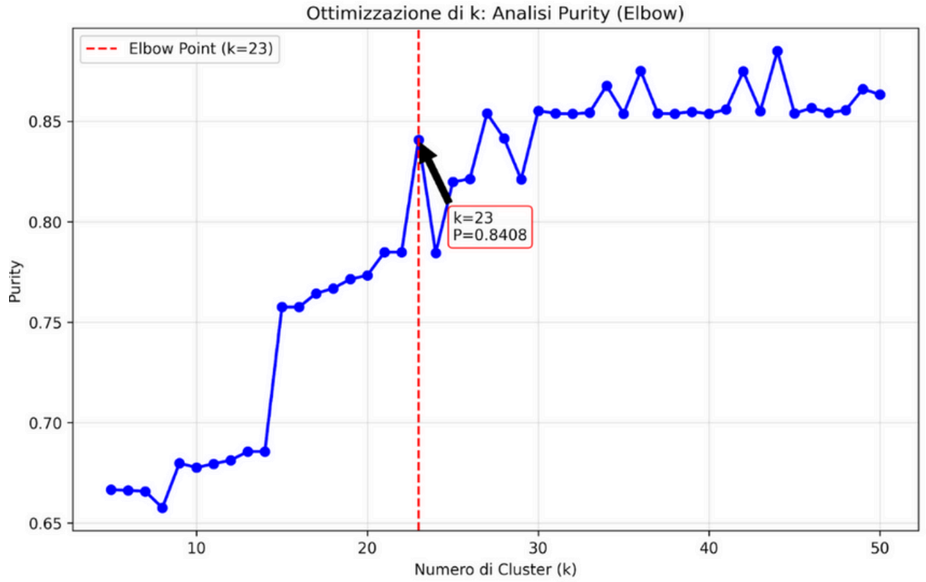


K-Means augmentation

K-Means

Applied **MinMaxScaler** to scale all features into a common [0, 1] range, ensuring stable gradient updates during training.

To determine the **optimal number of clusters** (k), we performed an analysis ranging from the number of target classes (k=5) up to k=50 (to not add a lot of features that could lead to the "curse of dimensionality" or make the model overly complex and prone to overfitting)

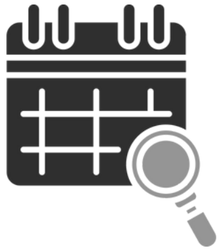


K-Means augmentation

Parameters

The **tuning process** explored the following hyperparameters to balance model complexity and generalization, with a total of 162 configurations:

- **Batch Size**: [32, 64, 128] — tested to optimize gradient stability and memory efficiency.
- **Learning Rate**: [0.0001, 0.001, 0.01] — evaluated to find the most precise convergence speed.
- **Dropout Rate**: [0.2, 0.3, 0.4] — utilized as a regularization technique to prevent overfitting.
- **Hidden Layers**: [2, 3] — evaluated to determine the necessary depth for modeling spatial distances.
- **First Layer Nodes**: [64, 128, 256] — tested to find the ideal initial capacity of the network.



K-Means augmentation

Training & Validation Result

TOP 10 by Validation F1-score						
Batch size	**Learning Rate**	**Dropout**	**1° layer neuron**	**Hidden layer**	**Validation Loss Min**	**Validation F1-score**
32	0.001	0.2	64	2	0.0575	0.9918
128	0.001	0.4	64	2	0.0564	0.9908
32	0.001	0.4	128	3	0.0623	0.9903
32	0.0001	0.3	256	2	0.0556	0.9898
32	0.01	0.2	128	2	0.0638	0.9896
64	0.0001	0.3	128	3	0.0561	0.9893
64	0.001	0.2	128	3	0.0541	0.9888
128	0.001	0.2	128	2	0.0628	0.9883
64	0.001	0.3	64	3	0.0559	0.9881
32	0.0001	0.2	128	3	0.0522	0.9878

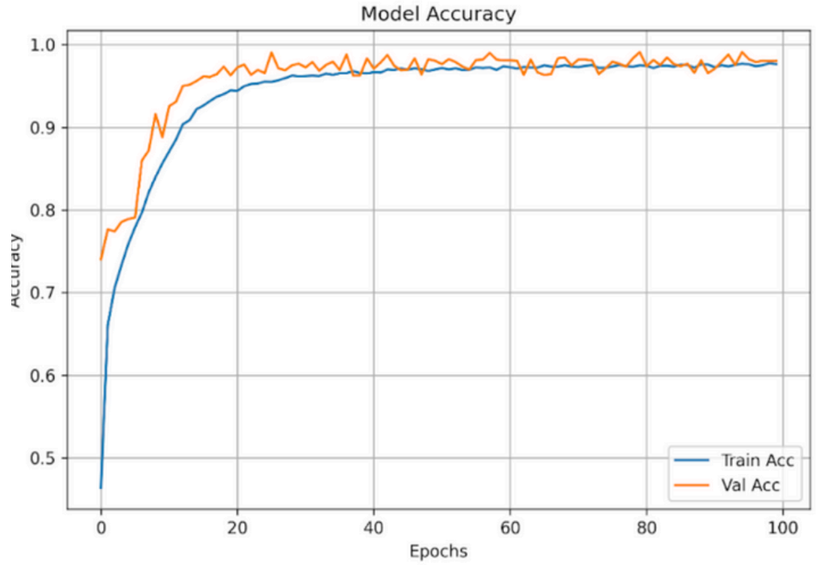
TOP 10 by Validation Loss						
Batch size	**Learning Rate**	**Dropout**	**1° layer neuron**	**Hidden layer**	**Validation Loss Min**	**Validation F1-score**
32	0.0001	0.2	256	3	0.0500	0.9818
32	0.0001	0.3	256	3	0.0506	0.9848
64	0.001	0.2	64	2	0.0510	0.9843
64	0.001	0.2	256	2	0.0510	0.9828
64	0.001	0.3	256	2	0.0513	0.9834
64	0.0001	0.2	256	3	0.0514	0.9833
32	0.0001	0.2	128	3	0.0522	0.9878
32	0.0001	0.4	256	3	0.0523	0.9828
32	0.0001	0.3	128	3	0.0525	0.9838
64	0.001	0.3	64	2	0.0526	0.9828

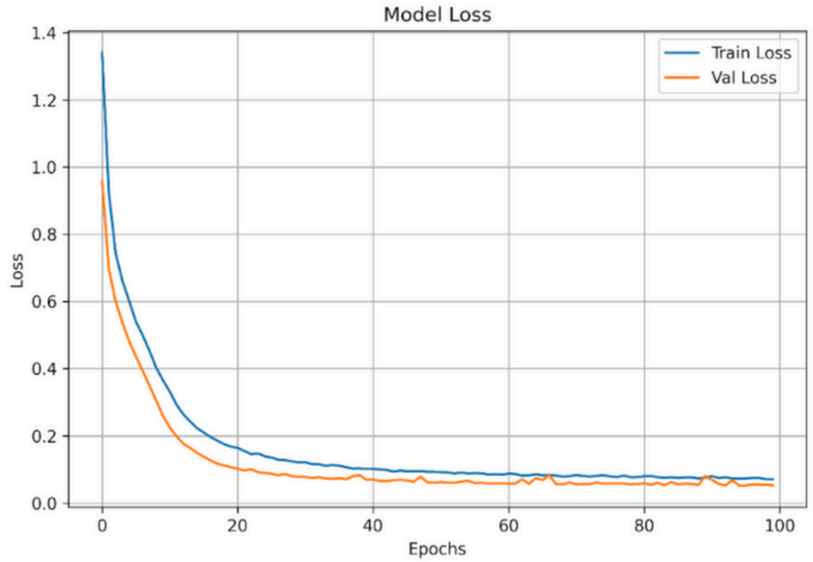
K-Means augmentation

Training & Validation Result

The model demonstrated a highly stable learning curve, as evidenced by the convergence of training and validation metrics. Unlike previous iterations, this configuration ran for the full duration of **100 epochs**.

The training process reached a minimum **Validation Loss** of **0.0522**.



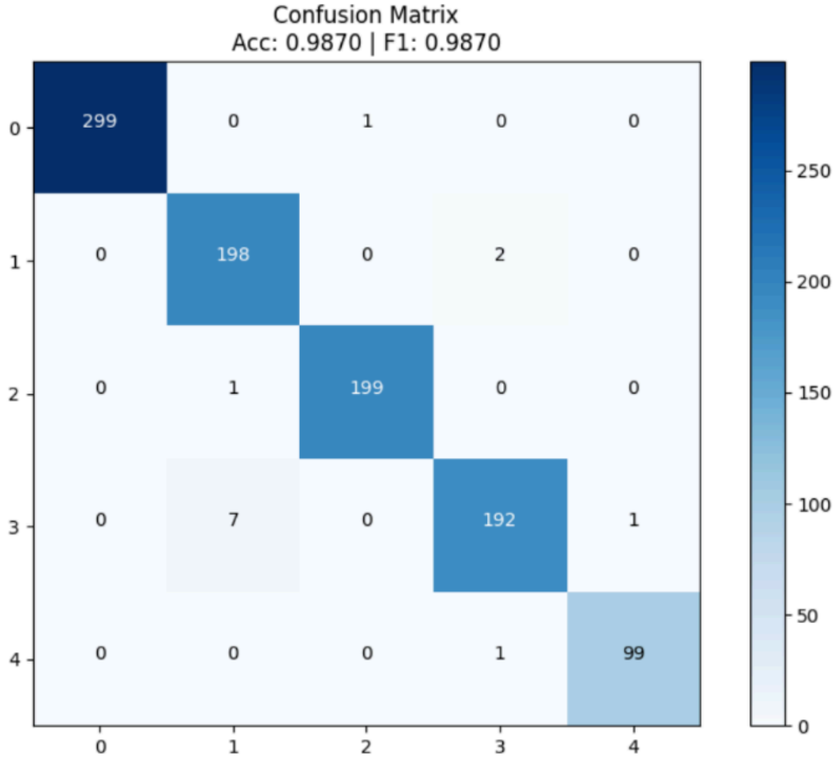


K-Means augmentation

Result & Conclusion

The high **98.7% Accuracy** and F1-Score on the external test set confirm that the three-layer dense architecture was sufficient to capture the non-linear relationships present in the 89 network features augmented with the k-means clustering.

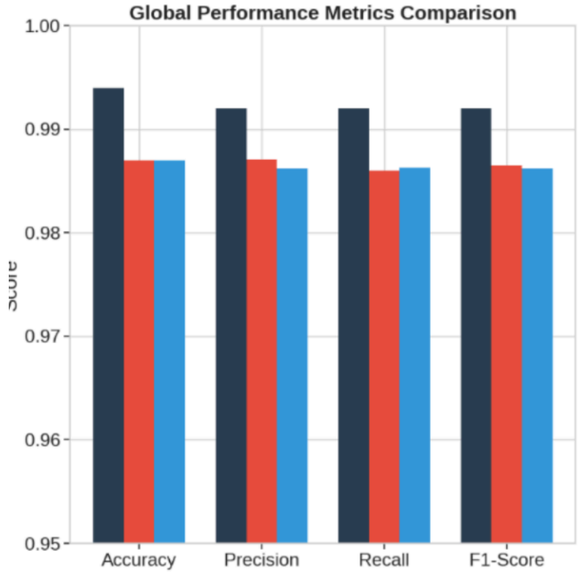
Class	Precision	Recall	F1-score	Analysis
0	1.00	1.00	1.00	Only one misclassification
1	0.96	0.99	0.98	Effectively captures almost all instances of Class 1.
2	0.99	0.99	0.99	Near-perfect identification.
3	0.98	0.96	0.97	Minimal confusion with Class 1.
4	0.99	0.99	0.99	Excellent performance on smaller datasets.



Model Comparison

Best Model Evaluetion

The **Random Forest** is selected as the final project model because it provides the best balance between maximum accuracy and operational simplicity. While the K-Means augmentation was a successful experiment in feature engineering, achieving a low validation loss of 0.0522 and proving stable over 100 epochs, but the introduction of 23 features doesn't help the model to improve the performance.



Metric	Random Forest	MLP	K-Means Augmented
Accuracy	99.40	98.70	98.70
F1-Score	0.9920	0.9865	0.9862
Precision	0.9920	0.9871	0.9862
Recall	0.9920	0.9860	0.9863
Training Epochs	-	55	100

Random Forest
MLP Baseline
K-Means Augmented

UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

KDDE

Comparison of Models for Traffic Classification

Thanks for your attention

Artificial Intelligence For Security

Prof:
Annalisa Appice
Giuseppina Andresini

Students:
Nicola Balzano