

**Phase 6**

**Machine Learning procedures**

**Mestrado de Engenharia Informática (MEI)**

**Metodologias Ofensivas e Defensivas de CiberSegurança**

**Prof. Dr. Ibéria Medeiros**

**Grupo 11**

David Martins (62725)

Mariana Bento (53676)

Miguel Mota (62702)

**2023/2024**

# Introdução

O objetivo deste trabalho é detetar ameaças num conjunto de dados de rede utilizando métodos de aprendizagem de máquina (ML). Diversos tipos de algoritmos foram considerados, incluindo modelos supervisionados e não supervisionados, mas optámos por utilizar modelos supervisionados devido à disponibilidade de dados rotulados.

# Tratamento dos Dados

A primeira etapa do processo foi o tratamento dos dados, uma fase crucial para garantir a qualidade dos resultados dos modelos de ML. Os dados foram lidos do dataset escolhido em formato CSV utilizando a biblioteca `pandas`, com especificações detalhadas dos tipos de dados para otimizar a memória e a eficiência do processamento.

## Passos de Tratamento dos Dados

O tratamento dos dados envolveu várias etapas importantes. Primeiramente, as colunas que continham apenas um valor foram removidas, pois não contribuíam para a discriminação entre classes. Em seguida, tratamos os valores faltantes, ou seja, as colunas com mais de 40% de valores faltantes foram eliminadas, enquanto para as colunas com valores faltantes em menos de 5% dos casos, as linhas correspondentes foram removidas. Além disso, valores infinitos em algumas colunas foram substituídos por NaN e posteriormente tratados. Por fim, foram escolhidas as colunas mais relevantes para a modelagem, baseando-se na análise de correlação e na importância das características.

Essas etapas garantiram que os dados utilizados para treinar os modelos fossem de alta qualidade e livres de inconsistências que poderiam comprometer a precisão dos modelos.

# Modelagem e Avaliação

Quatro algoritmos de aprendizagem de máquina foram escolhidos para comparação: Random Forest, Regressão Logística, Gaussian Naive Bayes e Multi-Layer Perceptron (MLP). Cada modelo foi treinado e avaliado utilizando um conjunto de dados tratado, com uma divisão de 70% para treino e 30% para teste.

## Algoritmos Utilizados

1. **Random Forest**: Escolhido pela sua capacidade de lidar com grandes volumes de dados e pela robustez contra overfitting.
2. **Regressão Logística**: Um modelo linear simples, útil como base de comparação.
3. **Gaussian Naive Bayes**: Adequado para problemas de classificação com suposições de independência entre as características.
4. **MLP**: Uma rede neural multicamadas que pode capturar relações não lineares nos dados.

## Procedimentos

Para cada modelo, seguimos uma série de procedimentos específicos com um enfoque técnico detalhado.

Primeiramente, realizámos a divisão dos dados em conjuntos de treino e teste, utilizando a função `train\_test\_split` da biblioteca `sklearn`. Especificamente, os dados foram divididos com uma proporção de 70% para treino e 30% para teste, assegurando que o conjunto de treino fosse suficientemente grande para a aprendizagem do modelo, enquanto o conjunto de teste permitisse uma avaliação robusta do desempenho.

Após a divisão, cada modelo foi treinado com os dados de treino. Este processo envolveu a utilização dos métodos `fit` disponíveis em cada algoritmo selecionado. Para o Random Forest e Gaussian Naive Bayes, foi utilizada a implementação padrão do `sklearn`, enquanto a Regressão Logística foi configurada para um máximo de 10.000 iterações para garantir a convergência. O MLP, por ser uma rede neural, exigiu um número maior de iterações (1000) para alcançar um bom nível de ajuste.

Após o treinamento, os modelos foram avaliados com os dados de teste. Para isso, utilizámos as previsões feitas pelo modelo nos dados de teste (método `predict` do `sklearn`). Avaliámos o desempenho dos modelos utilizando métricas padrão da classificação, incluindo a precisão (`accuracy\_score`), o recall (`recall\_score`), o f1-score (`f1\_score`) e a matriz de confusão (`confusion\_matrix`). Estas métricas foram escolhidas para fornecer uma visão abrangente da performance dos modelos, capturando tanto a capacidade de prever corretamente as classes quanto a relação entre falsos positivos e falsos negativos.

Especificamente para o MLP, houve a necessidade de padronizar os dados. A padronização foi realizada com o `StandardScaler`, também da biblioteca `sklearn`, que transforma as características para que tenham média zero e desvio padrão unitário. Este passo é crucial para redes neurais, pois características em diferentes escalas podem afetar negativamente a convergência do modelo e o desempenho geral. O `StandardScaler` foi ajustado nos dados de treino (`fit`) e aplicado tanto nos dados de treino quanto nos dados de teste (`transform`), assegurando que a mesma transformação fosse aplicada consistentemente.

Estas etapas garantiram que cada modelo fosse treinado e avaliado de maneira rigorosa e consistente, permitindo uma comparação justa do desempenho entre os diferentes algoritmos de aprendizagem utilizados.

## Resultados

A avaliação dos modelos revelou que o Random Forest apresentou o melhor desempenho geral, com um f1-score superior aos demais. A tabela a seguir resume as principais métricas de desempenho dos modelos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Modelo | Precisão (%) | Recall (%) | F1-Score (%) | Acurácia (%) |
| Random Forest |  |  | 99.969% |  |
| MLP |  |  | 98.842% |  |
| Gaussian Naive Bayes |  |  | 73.891% |  |
| Regressão Logística |  |  | 73.381% |  |

A matriz de confusão do modelo Random Forest mostrou uma excelente capacidade de distinguir entre classes benignas e maliciosas, minimizando tanto falsos positivos quanto falsos negativos.

# Conclusão

Através deste trabalho, demonstramos que a aplicação de algoritmos de aprendizagem de máquina pode efetivamente detectar ameaças em um ambiente de rede. O Random Forest foi identificado como o melhor modelo para este conjunto de dados específico, oferecendo uma combinação robusta de precisão, recall e f1-score.

O sucesso deste estudo se deve em grande parte ao cuidadoso tratamento dos dados e à seleção criteriosa das características mais relevantes. Este relatório documenta o processo e justifica as escolhas metodológicas, contribuindo para futuras implementações de sistemas de detecção de ameaças baseados em aprendizagem de máquina.