Universidade Federal do Paraná - UFPR
Departamento de Informática - DINF
Ciência de Dados para Segurança

Felipe Ribeiro Quiles

Relatório Técnico
Treinamento de Modelos para reconhecimento de ataques DDoS

Maio-2022

SUMÁRIO

1. INTRODUÇÃO

Este trabalho tem como finalidade o estudo e prática das técnicas de modelagem aprendidas na disciplina de “Ciência de Dados para Segurança” para geração de modelos preditores para o reconhecimento de ataques DDoS que possam acontecer na rede.

Desta forma, foi escolhido um dataset de ataques DDoS disponibilizado na plataforma Kaggle, o qual tem uma alta variedade de ataques, mas o problema está binarizado, sendo as amostras classificadas em “ddos” e “benigno”.

Para a utilização do dataset foi necessário o processamento dos dados. Sendo assim, decidiu-se a retirada de 6 características do vetor, as quais têm pouco impacto ao analisarmos uma situação real de ataque DDoS. Além disso, o dataset foi dividido em 20% para predição tardia e 80% para o treino e teste.

Após, foram modelados 3 tipos de preditores: o K-Neighbors, o RandomForest e o MLP. Foi utilizado a ferramenta GridSearchCV para a obtenção da melhor parametragem para cada um dos modelos de acordo com a métrica escolhida, no caso a precisão.

Para cada modelo é gerado a curva ROC e matrizes de confusão utilizando KFolding com valor de 5 para auxiliar Desta forma, é possível verificar se a mudança do dataset de treino faz com que os resultados obtidos se alterem drasticamente.

Por fim, após o treinamento o modelo é testado com os 20% do dataset que estava armazenado, e também com arquivos pcap de ataques reais de DDoS, sendo datasets diferentes. Assim, podemos verificar se o modelo tem bom desempenho para diferentes datasets.

2. APRESENTAÇÃO DO DATASET

O dataset é composto por 12794627 datapoints. Cada um deles corresponde a um fluxo, sendo de ida e volta. Os datapoints são compostos de 82 características. Eles podem estar classificados como “DDoS” e “Benigno”. O dataset utilizado tem 51% DDoS e 49% Benigno. Na Figura 1 é apresentado uma amostra do dataset em questão.

FIGURA 1: Amostra retirada do dataset de DDoS.

['624', '192.168.4.118-203.73.24.75-4504-80-6', '192.168.4.118', '4504', '203.73.24.75', '80', '6', '12/06/2010 08:34:32 AM', '3974862', '29', '44', '86.0', '59811.0', '86.0', '0.0', '2.9655172413793096', '15.969799083226464', '1460.0', '0.0', '1359.3409090909086', '372.02718975289076', '15068.950821437324', '18.365417466065487', '55206.41666666666', '195478.31665363663', '1566821.0', '167.0', '3735347.0', '133405.25', '341775.6887123293', '1805015.0', '167.0', '3974862.0', '92438.65116279072', '248174.8205743075', '1566821.0', '3997.0', '0', '0', '0', '0', '768', '896', '7.295850774190399', '11.069566691875089', '0.0', '1460.0', '809.4189189189186', '728.8624277195806', '531240.438541281', '0', '1', '0', '0', '0', '0', '0', '0', '1.0', '820.5068493150685', '2.9655172413793105', '1359.340909090909', '0', '0', '0', '0', '0', '0', '29', '86', '44', '59811', '-1', '5840', '1', '0', '0.0', '0.0', '0.0', '0.0', '0.0', '0.0', '0.0', '0.0', 'ddos']

3. PROCESSAMENTO DOS DADOS

Para facilitar o uso do dataset foi realizada uma diminuição no número de amostras, tornando-se 500 mil amostras com 50% “ddos” e 50% “benigno”. Esse dataset ainda foi dividido entre 20% e 80%. Sendo o 20% guardado para uso posterior ao treinamento do modelo, sendo uma validação do mesmo.

Além disso, foram retiradas 6 características dos vetores, sendo elas:

1. FlowID
2. IP Destino
3. IP Origem
4. Porta Destino
5. Porta Origem
6. Timestamp

Essas características foram removidas por se mostrarem muito específicas para esse dataset, mas que podem trazer problemas ao generalizar o modelo para outros valores e entradas.

Para facilitar o manejo da informação, a classe “ddos” passou a ser 1 e a classe “benigno” passou a ser identificada como 0.

4. **Treinamento e Resultados**

Foi utilizado para auxiliar no treinamento a ferramenta GridSearchCV que é uma ferramenta utilizada para automatizar o processo de ajuste dos parâmetros de um algoritmo. Desta forma, é possível encontrar os parâmetros que melhor se adaptam para que o modelo atinja a meta escolhida, nesse caso a maior precisão.

Os treinamentos foram realizados utilizando a proporção de 80/20 e 50/50 para o treino e teste do modelo. A classe positiva escolhida é a “ddos” já que se trata de um problema binário. Além disso, utilizou-se o KFolding igual a 5 para verificar as diferenças ao mudar as amostras de treino e teste, impactando no resultado que é obtido ao final.

O treinamento foi realizado usando um PC com as seguintes configurações: Processador Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz, Memória de 16 GB e Sistema Operacional Linux Mint 20.04.

4.1. KNN

Os parâmetros escolhidos para serem testados pelo GridSearchCV foram 1, 3 e 5 vizinhos. Na Figura 2 e 3, é possível verificar o tempo de treinamento e a obtenção dos melhores parâmetros para se obter uma maior precisão. Também é mostrado a relação entre falso positivos e recall dos 5-folds gerados, sendo que para o primeiro fold, são as amostras usadas para que se gere a versão final do modelo.

FIGURA 2: KNN proporção 50/50

```
Melhores parametros do KNeighbors para o precision_score
{'n_neighbors': 1}

Tempo de treinamento para o KNeighbors: 75.34206819534302

Auc Fold 0: 0.9946326840597697
Auc Fold 1: 0.9973693028338546
Auc Fold 2: 0.9974340483774409
Auc Fold 3: 0.9972134869567371
Auc Fold 4: 0.9972990658683852
```

Já nas figuras 4 e 5, são mostradas as curvas ROC geradas para o KNN com proporção 50/50 e 80/20, respectivamente. Ao analisar é possível perceber que as 5 curvas estão muito próximas umas das outras, visto que em todos os folds as relações falso positivos e recall estão em 99%.

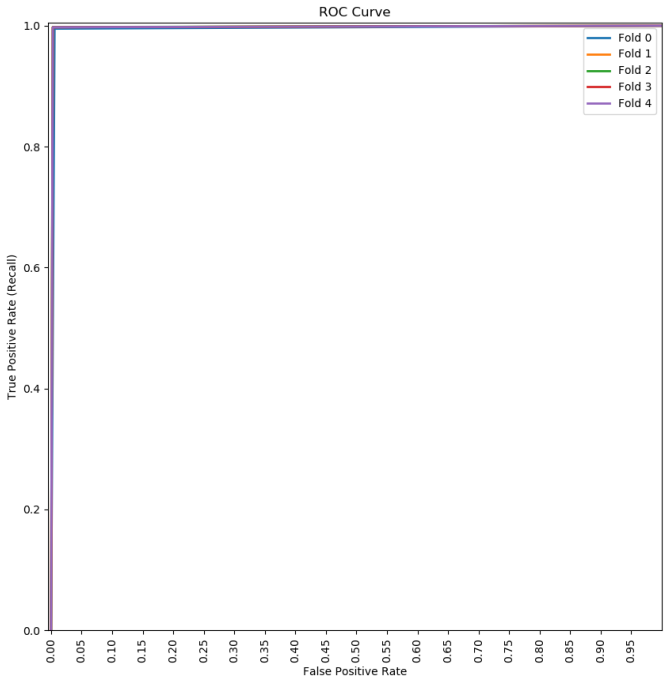
FIGURA 3: KNN proporção 80/20

```
Melhores parametros do KNeighbors para o precision_score
{'n_neighbors': 1}

Tempo de treinamento para o KNeighbors: 159.31919622421265

Auc Fold 0: 0.9955827660583146
Auc Fold 1: 0.9990088675790347
Auc Fold 2: 0.9992974465756712
Auc Fold 3: 0.999046279035318
Auc Fold 4: 0.9991473299027351
```

FIGURA 4: Curva ROC - 50/50



As Figuras 6 e 7, apresentam as matrizes de confusão para a predição da classe realizada pelo modelo KNN.

Ao comparar os resultados obtidos pelos autores e essa réplica de experimento vê-se que estão bem próximos, tendo diferenças de menos de 1%, mesmo que não seja a mesma quantidade, nem a mesma escolha de amostras para o treinamento.

FIGURA 5: Curva ROC - 80/20

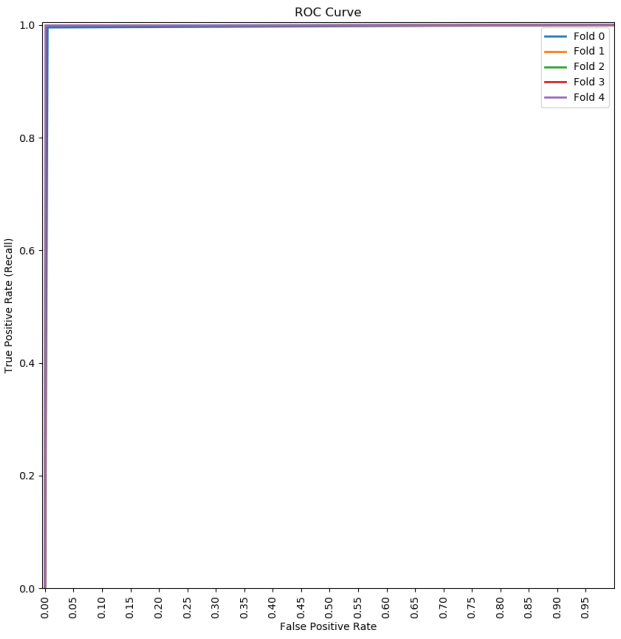


FIGURA 6: Matriz de Confusão 80/20

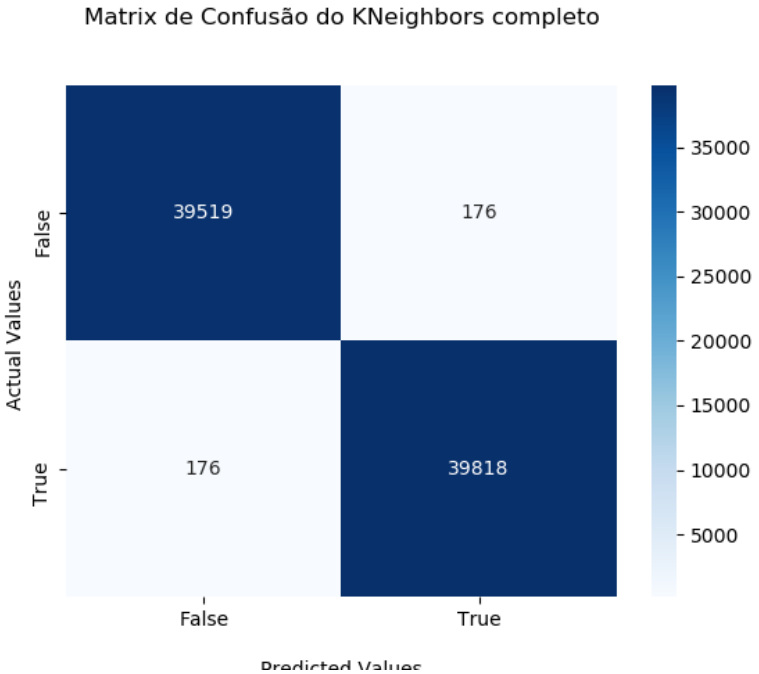
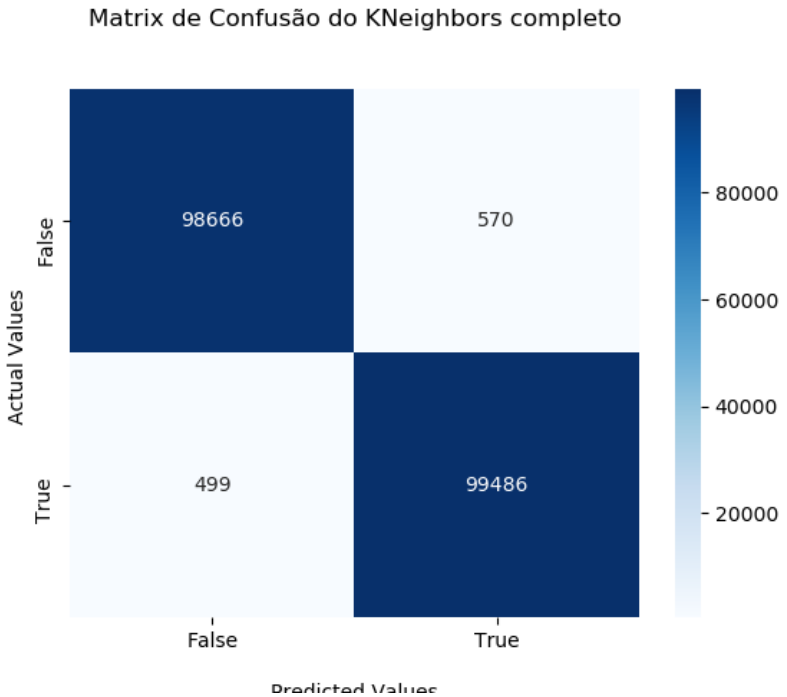


FIGURA 7: Matriz de Confusão 50/50



4.2 RandomForest

Os parâmetros escolhidos para serem testados pelo GridSearchCV foram 50 e 100 árvores. Na Figura 8 e 9, é possível verificar o tempo de treinamento e a obtenção dos melhores parâmetros para se obter uma maior precisão. Também é mostrado a relação entre falso positivos e recall dos 5-folds gerados, sendo que para o primeiro fold, são as amostras usadas para que se gere a versão final do modelo.

FIGURA 8: RF proporção 50/50

```
Melhores parametros do RandomForest para o precision_score
{'n_estimators': 100}

Tempo de treinamento para o RandomForest: 80.37305212020874

Auc Fold 0: 0.9999229980429992
Auc Fold 1: 0.9999608262816269
Auc Fold 2: 0.9999556606976475
Auc Fold 3: 0.9999709526544666
Auc Fold 4: 0.9999701640118444
```

Já nas figuras 10 e 11, são mostradas as curvas ROC geradas para o RF com proporção 50/50 e 80/20, respectivamente. Ao analisar é possível perceber que as 5 curvas estão muito próximas umas das outras, visto que em todos os folds as relações falso positivos e recall estão em 99%.

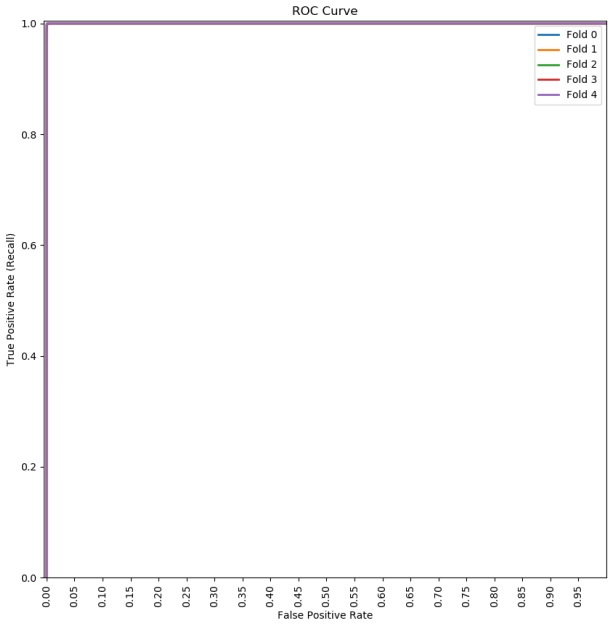
FIGURA 9: RF proporção 80/20

```
Melhores parametros do RandomForest para o precision_score
{'n_estimators': 100}

Tempo de treinamento para o RandomForest: 221.26284837722778

Auc Fold 0: 0.9999029121278382
Auc Fold 1: 0.9999728870402483
Auc Fold 2: 0.999986604616212
Auc Fold 3: 0.999973619295193
Auc Fold 4: 0.9999600695867069
```

FIGURA 10: Curva ROC - 50/50



As Figuras 12 e 13, apresentam as matrizes de confusão para a predição da classe realizada pelo modelo RandomForest.

Ao comparar os resultados obtidos pelos autores e a réplica de experimento vê-se que estão bem próximos, tendo diferenças menores que 1%, atingindo 4 casas decimais com 9, mesmo que não seja a mesma quantidade, nem a mesma escolha de amostras para o treinamento.

FIGURA 11: Curva ROC - 80/20

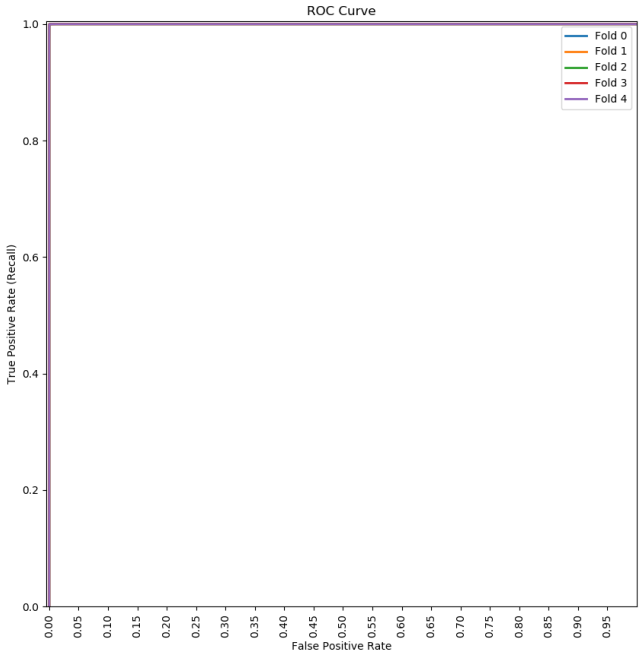
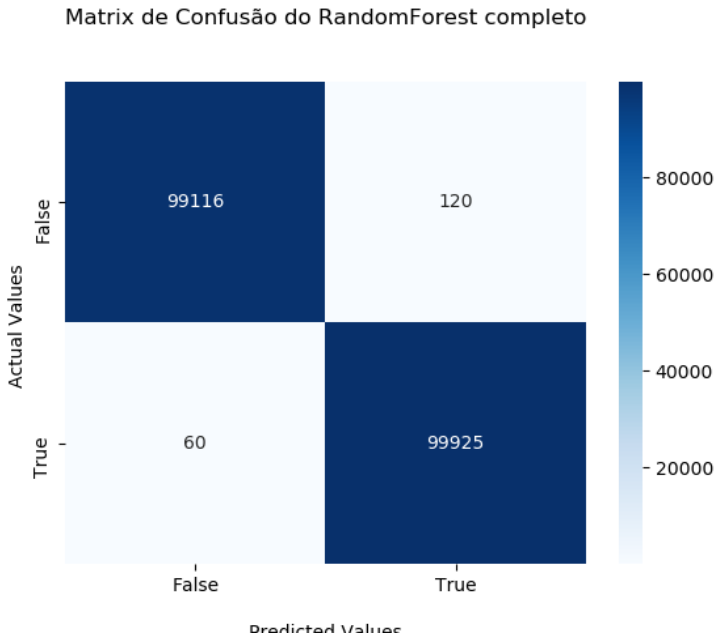


FIGURA 12: Matriz de Confusão 50/50



4.3 MLP

Os parâmetros escolhidos para serem testados pelo GridSearchCV foram 1000 E 5000 épocas com erro de 0.01 e 7 camadas ocultas. Na Figura 14 e 15, é possível verificar o tempo de treinamento e a obtenção dos melhores parâmetros para se obter uma maior precisão. Também é mostrado a relação entre falso positivos e recall dos 5-folds gerados, sendo que para o primeiro fold, são as amostras usadas para que se gere a versão final do modelo.

FIGURA 13: Matriz de Confusão 80/20

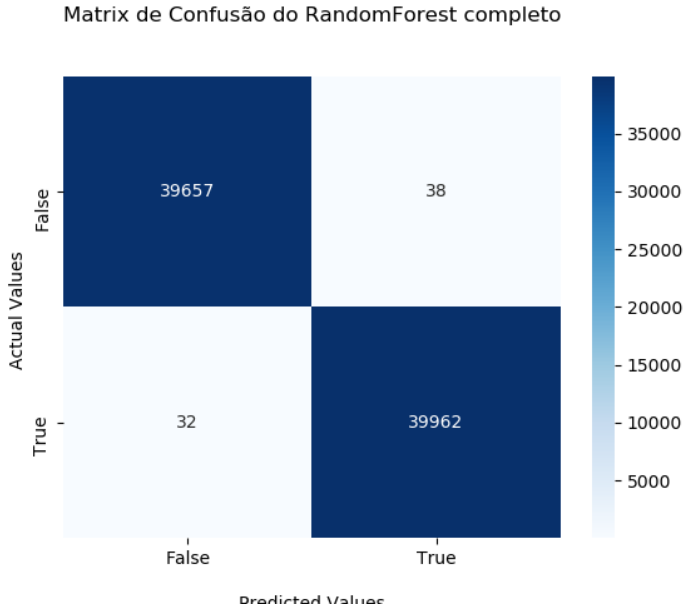


FIGURA 14: MLP proporção 50/50

```
Melhores parametros do MLP para o precision_score
{'hidden_layer_sizes': 7, 'max_iter': 1000, 'tol': 0.01}

Tempo de treinamento para o MLP: 90.26089715957642

Auc Fold 0: 0.8064113343471754
Auc Fold 1: 0.8061260448690828
Auc Fold 2: 0.8049639165210506
Auc Fold 3: 0.805409465537288
Auc Fold 4: 0.8038654728033059
```

Já nas figuras 16 e 17, são mostradas as curvas ROC geradas para o MLP com proporção 50/50 e 80/20, respectivamente. Ao analisar é possível perceber que as 5 curvas estão muito próximas umas das outras, visto que em todos os folds as relações falso positivos e recall estão em 80%. Fato que ocorre pois o modelo se especializou em encontrar os DDoS, fazendo com o mesmo tenha sua taxa de falso positivos aumentada, visto que seus os pesos nas camadas que designam um DDoS tende a aumentar, fazendo com que amostras benignas que pudessem estar perto do limiar de decisão sejam interpretadas como DDoS.

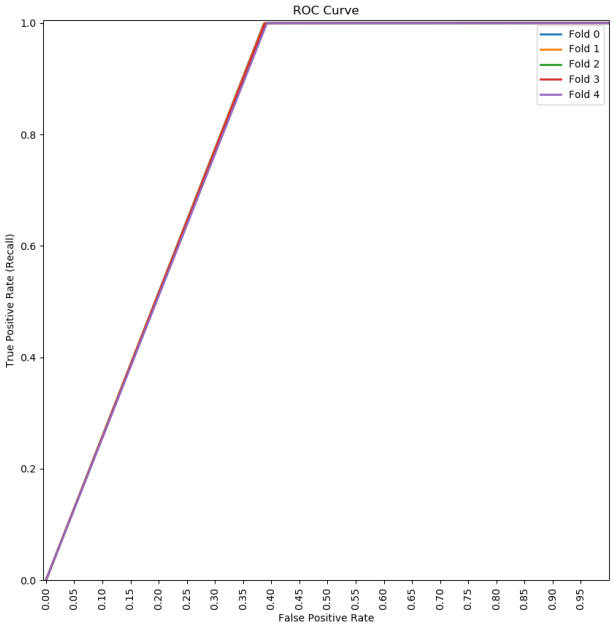
FIGURA 15: MLP proporção 80/20

```
Melhores parametros do MLP para o precision_score
{'hidden_layer_sizes': 7, 'max_iter': 1000, 'tol': 0.01}

Tempo de treinamento para o MLP: 133.3489019870758

Auc Fold 0: 0.756177156262317
Auc Fold 1: 0.7553772478896146
Auc Fold 2: 0.7549926279091757
Auc Fold 3: 0.7566618195903589
Auc Fold 4: 0.7523986999611851
```

FIGURA 16: Curva ROC - 50/50



Por último, as Figuras 18 e 19, apresentam as matrizes de confusão para a predição da classe realizada pelo modelo MLṔ.

5. Validação do Modelo

A validação do modelo foi realizada utilizando os 20% do dataset que estavam separados, justamente para esse fim, além de testar o modelo para outros vetores de características, originados de diferentes datasets.

FIGURA 17: Curva ROC - 80/20

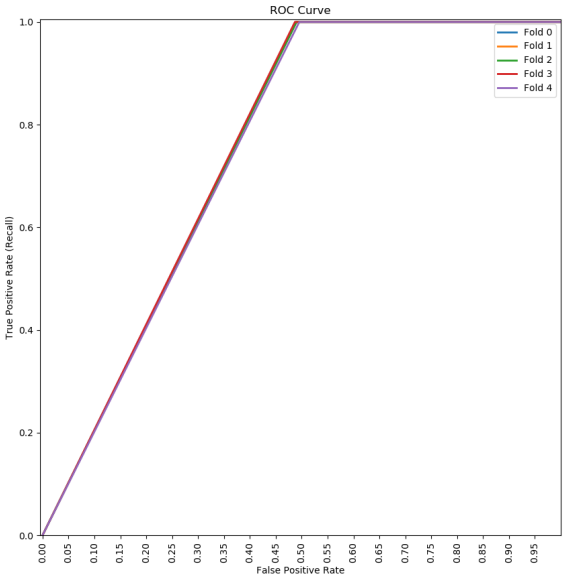
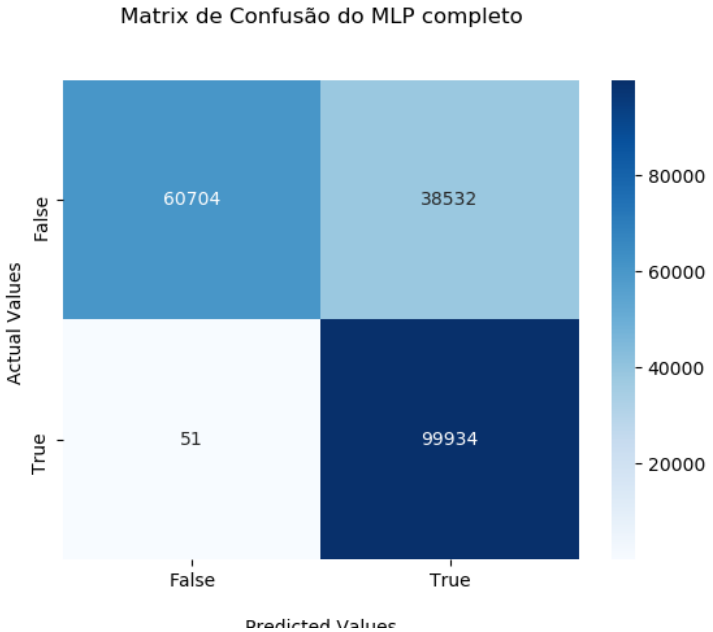


FIGURA 18: Matriz de Confusão 50/50



Na Figura 20, é possível verificar que o modelo do RF com proporção de 80/20 consegue reconhecer todas as amostras que são classificadas como DDoS, tendo 0 Falso Negativos.

FIGURA 19: Matriz de Confusão 80/20

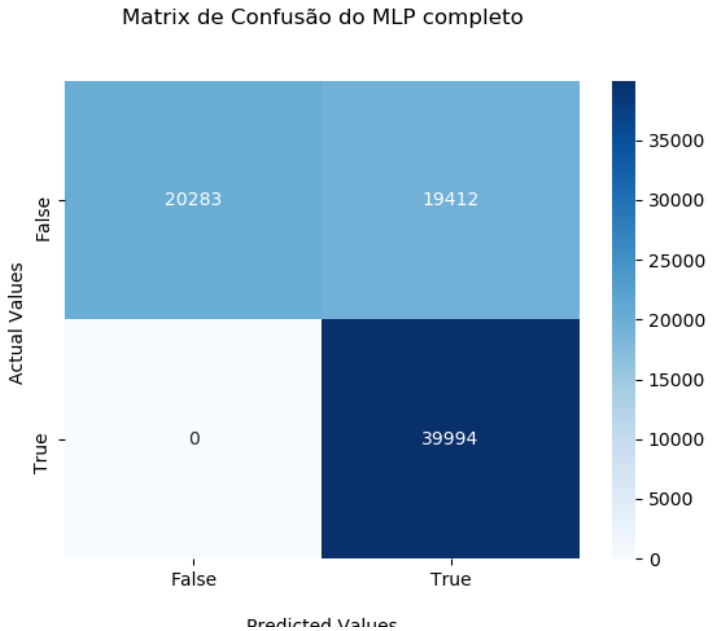
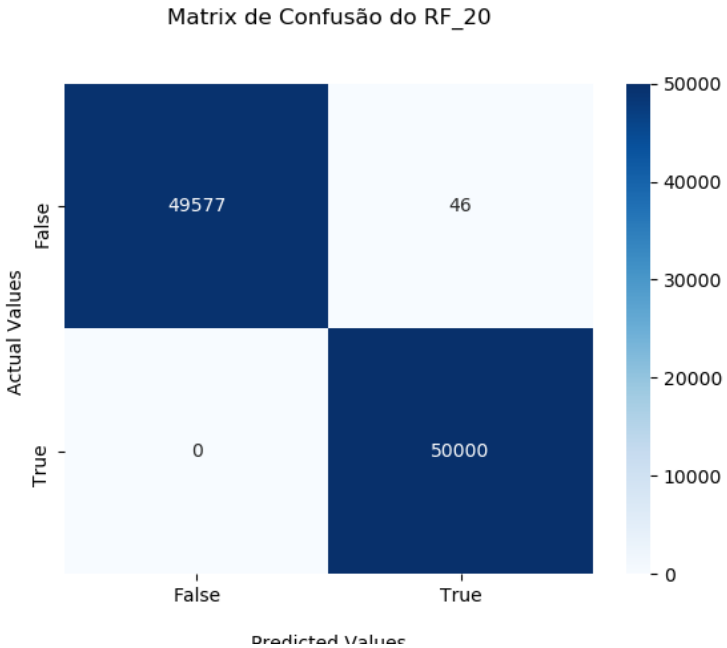
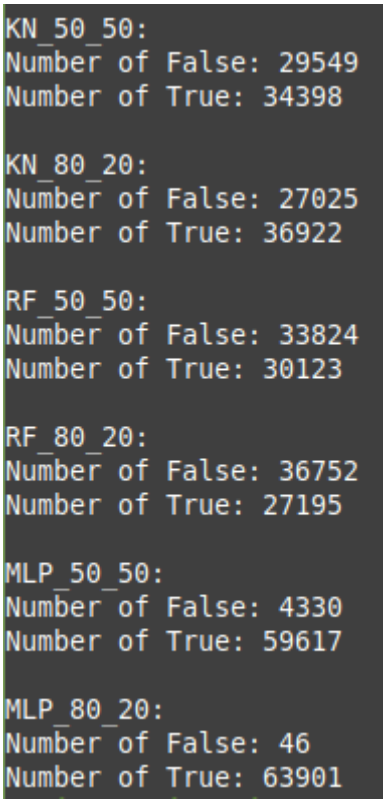


FIGURA 20: Matriz de Confusão 80/20



Já a Figura 21, mostra os resultados obtidos pelo KNN, RF e MLP, tanto para 50/50, quanto para 80/20 ao aplicar no modelo um dataset de DDoS diferente, que contêm apenas dados de ataques. Ao analisar é possível verificar que apenas o MLP 80/20 consegue chegar próximo aos valores que foram obtidos com o dataset original. Entretanto, esse fato pode ser devido ao MLP estar detectando muito mais DDoS que os outros dois modelos, visto que chegou a ter 50% de Falsos Positivos.

FIGURA 21: Resultados obtidos dataset diferente



6. Conclusão

Após os estudos, aplicações dos métodos e obtenção dos resultados, percebe-se que colocar os modelos na prática não é simples como parece, mesmo que seus resultados tenham sido excelentes. Desta forma, apesar de obter bons resultados ao treinar e testar os modelos com o dataset principal, ao mudar esse dataset, já percebe-se uma grande piora nos resultados, a exemplo do RF que obteve valores próximos a 100%, mas não consegue manter essa taxa ao analisar diferentes amostras que são provenientes de outros datasets, tendo uma grande queda na taxa.

Apesar disso, a predição é uma importante área da Ciência de Dados, e pode auxiliar na melhora da Segurança da Internet de diversas formas, assim como no reconhecimento e prevenção de ataques DDoS. Destarte, deve-se continuar e incentivar pesquisas e investimentos na área para que a mesma possa se desenvolver ainda mais.