Projeto Final - IoT
Análise do tráfego de dispositivos inteligentes

Igor Varejão[1]

[1]Departamento de Informática – Universidade Federal do Espírito Santo (UFES)

igor.varejao@edu.ufes.br

Abstract. *At every instant, a range of data is transmitted through intelligent devices in our environment, making the Internet of Things a technology that is increasingly present in our daily lives. This area is composed of a vast list of heterogeneous devices that make it difficult to understand the reality of this area. In this context, data analysis should be used to recognize patterns in the communication of these devices, to help in recognizing their types. Therefore, in this article, a dataset representing the network traffic of smart devices is presented and, from it, a model is built that classifies the type of device based on the data of the packets transmitted by it in an arbitrary period of time. The results were promising with an accuracy of* 84.4% *and showed a research area with great room to be explored. The data as well as the code used in this labour is available at Smart devices traffic analysis.*

Resumo. *A cada instante uma gama de dados são transmitidos por meio de dispositivos inteligentes em nosso ambiente, tornando a Internet das Coisas uma tecnologia cade vez mais presente no dia-a-dia. Tal área é composta por uma vasta lista de dispositivos heterogêneos que dificultam a compreensão da realidade dessa área. Nesse contexto, a análise de dados deve ser usados para reconhecer padrões no modo de comunicação desses dispositivos, para ajudar no reconhecimento dos tipos dos mesmos. Portanto, nesse artigo, é apresentado um dataset que representa o tŕafego de rede de dispositivos inteligentes e, a partir dele, é construído um modelo que classifica o tipo de dispositivo com base nos dados dos pacotes transmitidos por este em um período de tempo arbitrário. Os resultados foram promissores com uma acurácia de* 84.4% *e evidenciaram uma área de pesquisa com grande espaço para explorar. Os dados assim como o código desenvolvido nesse trabalho está disponível em Anaĺise do tráfego de dispositivos inteligentes.*

1. Introdução

O cenário da Internet das Coisas é uma realidade bem diversa, no qual há uma variedade de dispositivos que constantemente estão se intercomunicando por meio de pacotes de rede. Saber identificar a origem de um determinado pacote é de extrema importância para entender se os dados compartilhados podem ser individualizados e, portanto, caracterizados.

Dessa forma, foi proposto nesse artigo um modelo capaz de identificar o tipo de dispositivo inteligente a partir de meta-dados da captura de pacotes em determinado período de tempo. Com isso, na Seção 2, 3 é descrito o dataset utilizado para treinar o

modelo assim como o pré-processamento aplicado, já na Seção 4 e 5 descreve o processo de treinamento e avaliação do modelo, por fim, na Seção 6 e 7 descrevem os problemas encontrados durante o desenvolvimento desse modelo como também discute sobre os resultado encontrados e futuros trabalhos.

2. Descrição do dataset

O dataset utilizado foi retirado do capítulo 5 do livro `Machine Learning for Cybersecurity Cookbook` [2] do qual foi disponibilizado todo o código fonte utilizado no livro na plataforma GitHub. O dataset consiste de amostras que representam os pacotes transmitidos em determinado período de tempo por um dispositivo inteligente, logo, uma amostra contém o período de início e fim da captura desses dados, a quantidade de pacotes transmitidas nesse período, a média de bytes, além de outras características, que totalizam 297 características.

As amostras foram rotuladas pelo tipo do dispositivo inteligente, dessa forma, o dataset se caracteriza como supervisionado por já antever a classe das amostras. O conjunto de classes contidos no dataset é listada abaixo:

- `baby_monitor`
- `lights`
- `motion_sensor`
- `security_camera`
- `smoke_detector`
- `socket`
- `thermostat`
- `TV`
- `watch`
- `water_sensor`

O dataset foi disponibilizado já divido em 2 arquivos, `iot_devices_train.csv` e `iot_devices_test.csv`, no qual o arquivo de treino possui 900 amostras divididas igualmente entre 9 classes e, o arquivo de teste, 1000 amostras divididas igualmente entre 10 classes, tal qual o conjunto de classes do arquivo de treino C_{train} está contida no conjunto de classes do arquivo de teste C_{test}, ou seja, $C_{train} \subset C_{test}$. Dessa forma, há uma classe que está contida no conjunto de teste e não no conjunto de treino, logo, haveria duas possíveis abordagens:

1. Eliminar a classe do conjunto de teste, o que implicaria na perda de dados, e pior, de uma classe.
2. Desconsiderar a divisão inicial de treino e teste e juntá-los, distribuindo a classe **`water_sensor`** no treino e teste.

A abordagem escolhida e, a mais recomendada pela literatura, foi a que não implica na perda de dados e, portanto, a divisão feita previamente foi desconsiderada. Sendo assim, apesar da classe `water_sensor`, o dataset possui uma distribuição quase uniforme das classes como mostra a Figura 1.

3. Pré-processamento

Pré-processamento de dados refere-se às etapas de preparação e transformação dos dados brutos antes de serem utilizados em uma análise ou em um modelo de *machine learning*.

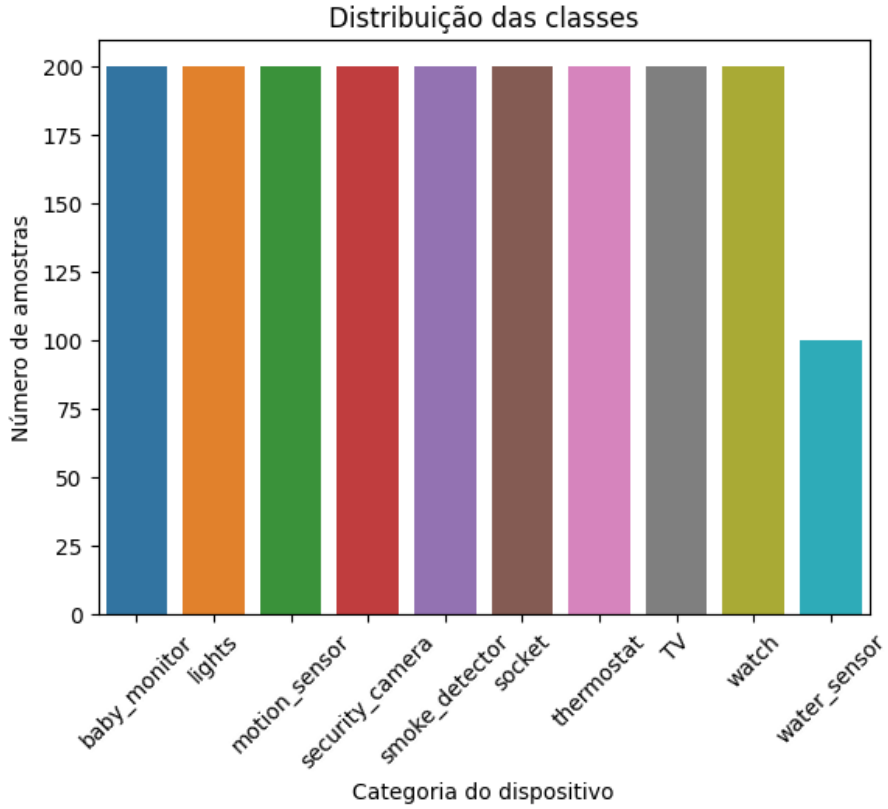

Figura 1. Distribuição das classes no dataset

Essas etapas são essenciais para garantir que os dados estejam em um formato adequado, consistente e pronto para serem utilizados de forma eficaz. Portanto, foram analisados 2 tipos de problemas, dados faltantes 3.1 e variáveis correlacionadas 3.2.

3.1. Dados faltantes

A primeira tratativa dos dados foi lidar com dados faltantes, apesar de ser uma tratativa bem comum no contexto de *IoT*, esse dataset específico não consta com nenhum dado faltante, provavelmente, por ser um dataset sintético retirado de um livro, feito em condições controladas. Dessa forma, esse processo não modificou nenhum aspecto do dataset.

3.2. Variáveis correlacionadas

Variáveis correlacionadas são aquelas que apresentam algum tipo de relação ou associação entre si. Essa relação pode ser medida através de um coeficiente de correlação, que indica a força e direção dessa associação. Dessa forma, a existência destas implica num aumento da dimensionalidade dos dados ao passo que não agrega mais informações, tal fato é perigoso pois pode ocasionar o **mal da dimensionalidade** [1] que são os diversos problemas ocasionados por conjuntos com alta dimensionalidade, como a dificuldade na visualização dos dados, aumento da complexidade do modelo, necessidade de uma quantidade maior de dados, entre outros.

Tendo em vista esse cenário foi aplicado a seguinte metodologia para eliminar variáveis fortemente correlacionadas:

1. A partir da matriz de correlação , figura 2, entre as variáveis foram selecionados os pares de variáveis fortemente correlacionadas.
2. De cada par, foi eliminado a variável que possui menor correlação com a classe.

Pela matriz de correlação, figura 2, há a presença de algumas áreas consideráveis com variáveis fortemente correlacionadas, isso mostra que há uma grande quantidade de variáveis com esse requisito e, por isso, foram removidas 166 características das 297 presentes previamente, cerca de 55% das variáveis.

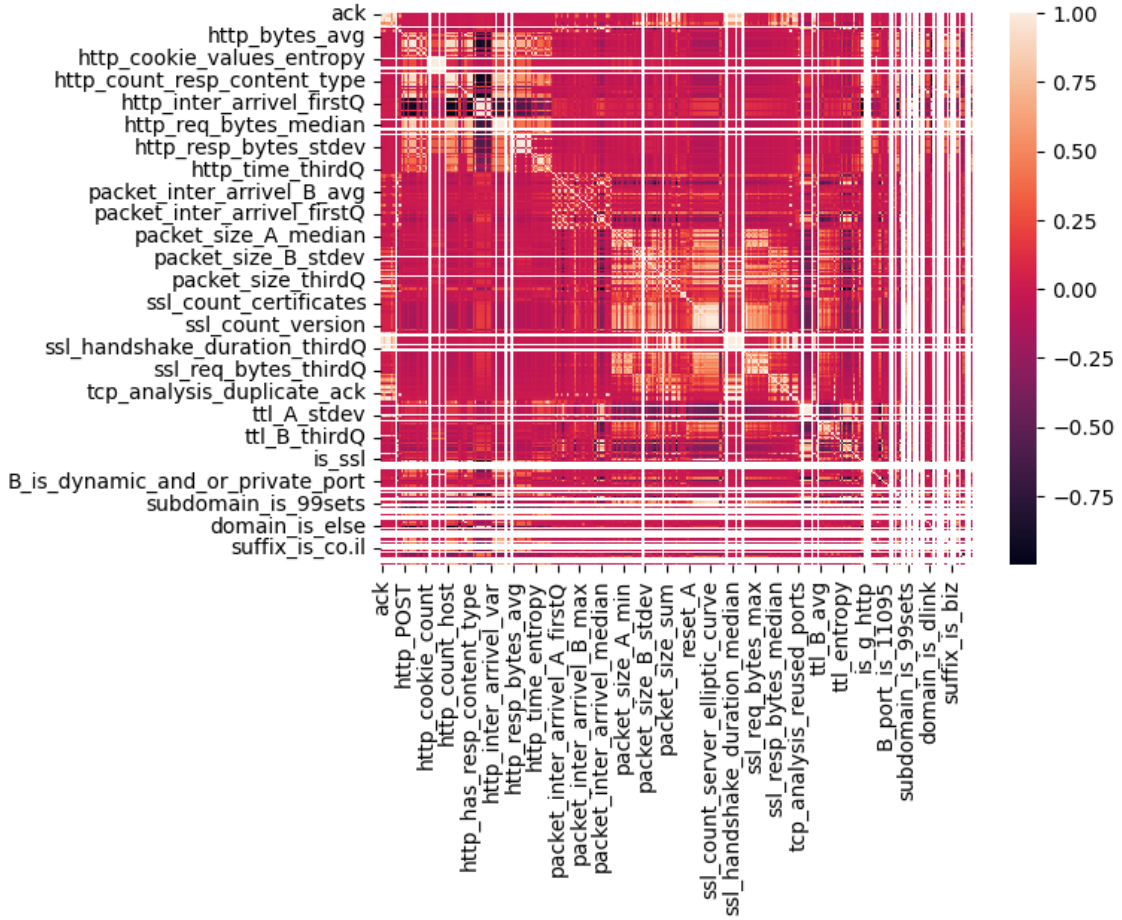

Figura 2. Matriz de correlação das variáveis do dataset

4. Treinamento

Um grande problema na área de aprendizado de máquina são os modelos enviesados. Os modelos enviesados são aqueles que desempenham muito bem no conjunto de dados que foram expostos, mas quando são alimentados com dados totalmente novos, acabam tendo um desempenho muito abaixo do reportado em seu treinamento.

Como os dados são escassos, devemos amostŕa-los com o objetivo de melhor representar a realidade dos dados de forma honesta, para que o modelo possa ser o mais robusto possível e generalize melhor o conhecimento do conjunto em mãos.

Dessa forma, nesse artigo, foi utilizado o método de amostragem de validação cruzada aninhada, utilizando uma busca em grade no ciclo interno para procurar os melhores

parâmetros do classificador C no conjunto de treino. No ciclo externo, foi usado uma validação cruzada com $5folds$ em que em um primeiro momento foi só responsável por delimitar o 5 conjuntos de treino no qual haveria a busca dos parâmetros. Esse esquema pode ser visto na Figura 3.

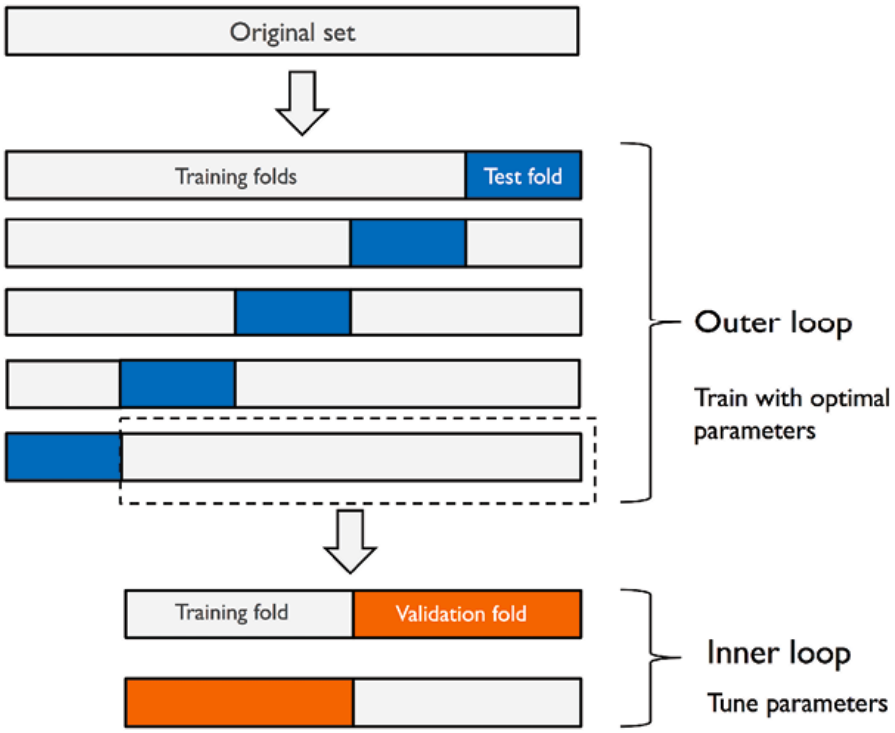

Figura 3. Validação cruzada aninhada

Com isso, foram selecionados 5 conjuntos de parâmetros do classificador C. Para escolher de fato o melhor conjunto de parâmetros, foi feito um processo de escolha onde o valor do parâmetro mais frequente dentro os 5 conjuntos era o escolhido e, em caso de empate, o primeiro a ser visto era o escolhido.

Definido esse processo, foram selecionados 5 classificadores arbitrariamente, conhecido na literatura por terem bom desempenho com dados tabulares, são eles:

- XGBClassifier
- RandomForestClassifier
- AdaBoostClassifier
- GradientBoostingClassifier
- BaggingClassifier

A partir do processo definido, foram executados o processo descrito para cada classificador fazendo a escolha dos melhores parâmetros para cada um destes, ou seja, foram executados 5 validações cruzadas aninhadas.

Em razão de se ter um classificador mais robusto e com uma predição mais estável foi criado um classificador de votação V a partir destes 5 classificadores, já com os parâmetros otimizados, no qual a classificação de V combina a predição dos seus estimadores. O classificador de votação funciona com um conjunto de classificadores de

tal forma que o a predição do classificador dependa da combinação dos classificadores componentes do mesmo, uma das formas de fazer essa escolha, e como foi implementada nesse artigo, é por meio da votação majoritária das predições dos classificadores que compõe o *ensemble*, ou seja, a classe mais predita é a escolhida, a Figura 4 exemplifica esse funcionamento.

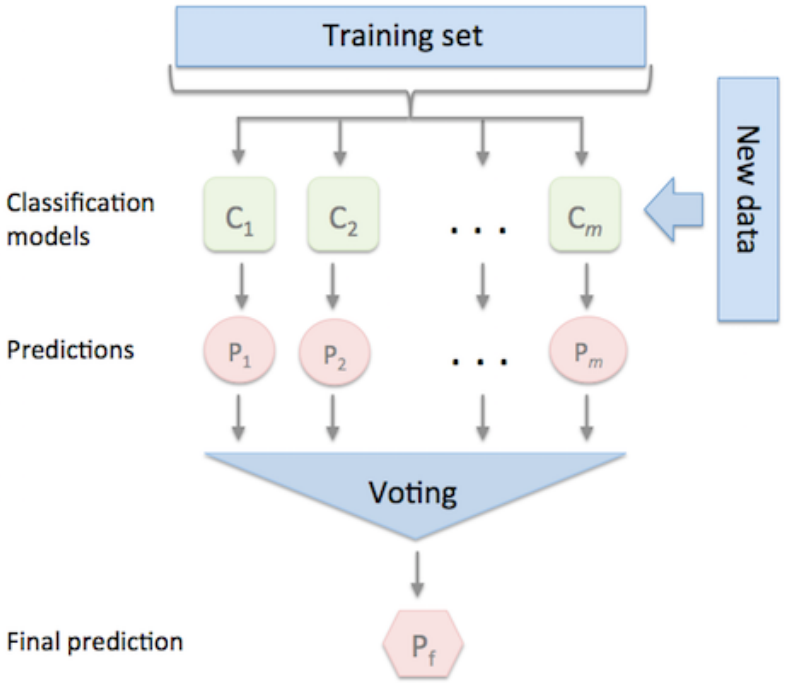

Figura 4. Funcionamento de um classificador de votação

5. Avaliação do desempenho

Para avaliar o modelo V foi utilizado os *folds* de teste do ciclo externo da busca de parâmetros, que foram utilizados apenas para prover *insights* do desempenho do classificador separado, tal que esta informação não influenciou em nenhuma decisão, sem acarretar *test-leaking*. De forma resumida, foi utilizada um validação cruzada simples com $5 folds$ no dataset inteiro para medir o desempenho deste *ensemble* V.

Foram analisadas 2 métricas nessa avaliação, a **acurácia** e o **f1-score macro**, a primeira é a mais usada nas análises por ser de fácil compreensão, contudo, esta pode esconder um comportamento enviesado do modelo. Dado isso, foi usado a outra métrica **f1-score macro** que leva em conta o *recall* e o *precision* de cada classe, dessa forma, o modelo não consegue obter um bom desempenho ignorando da existência de uma classe específica.

O modelo conseguiu uma média de $84,4\%$ de acurácia nos 5 folds e um *f1-score* de 0.825. A matriz de confusão, Figura 5, gerada a partir das predições dos folds de teste, mostram que o modelo desempenhou bem na maioria das classes contudo há 2 ressalvas:

- O modelo teve problema em diferenciar o dispositivos de luz com os *sockets* e vice-versa
- O desempenho da classe `water_sensor` foi muito abaixo das outras classes, o que se explica pelo menor número de amostras dessa classe.

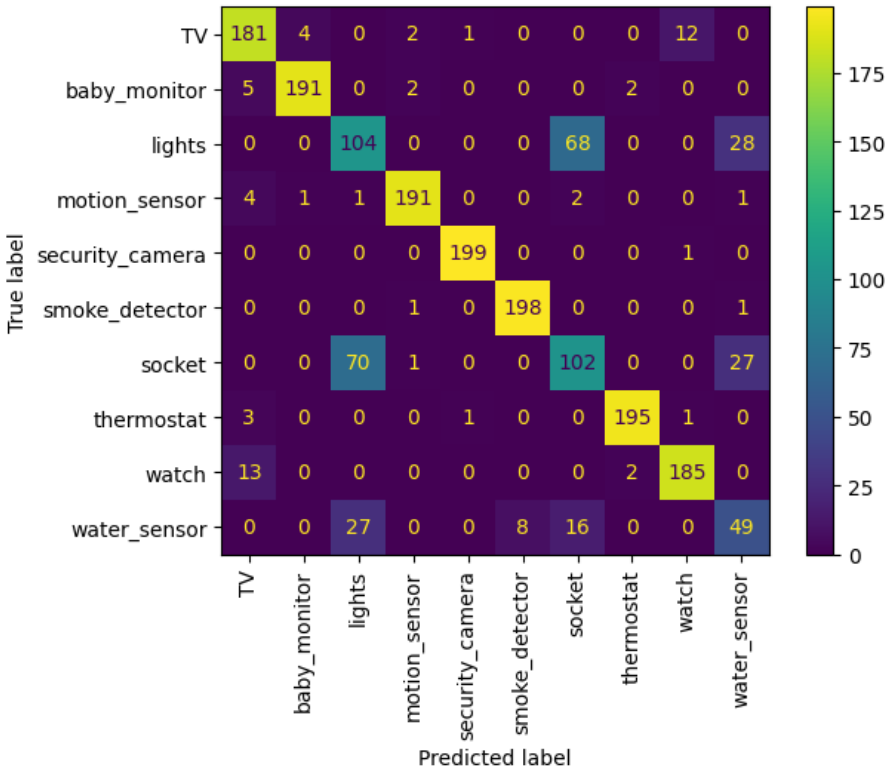

Figura 5. Matriz de confusão

6. Problemas encontrados

Durante a procura de um dataset que caracterizasse o tráfego de dispositivos inteligentes, foram encontrados vários voltados para a segurança, mais especificamente, na identificação de ataques *DDoS* (*Distributed Denial-of-Service*), o que indica que há uma escassez de dados de tráfego de dispositivos inteligentes. Além disso, os que existem, não possuem um grande volume de dadaos, o que é um problema para os modelos de aprendizado de máquina já que estes são extremamente dependentes.

7. Conclusão

Apesar do dataset possuir relativamente poucos dados, o classificador desenvolvido teve um desempenho razoável em comparação com trabalhos realizado no mesmo dataset, além do que o resultado encontrado, devido a sua amostragem, possui um grau de confiança maior, sendo o modelo desenvolvido mais robusto.

Os resultados mostraram que é possível classificar dispositivos inteligentes pelo seu tipo só por informações básicas de seus pacotes transmitidos, o que levanta questões sobre privacidade e vulnerabilidade da rede.

Diante disso, seria interessante no futuro, treinar o modelo com um dataset mais volumoso ou, tentar abordagens mais gananciosas como utilizar transferência de aprendizado para fazer um pré-treino do modelo a partir de dados de pacotes de rede em geral e, depois, treiná-los nesse conjunto mais reduzido de pacotes de dispositivos inteligentes.

De toda forma, a anáĺise do tráfego de dispositivos inteligentes provou ser uma

área com bastante espaço para pesquisa para assim, podermos entender melhor o ambiente em que vivemos e as tecnologias que nos rodeiam.

Referências

[1] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[2] Emmanuel Tsukerman. *Machine Learning for Cybersecurity Cookbook*. Packt, 2019.