Universidade Federal do Paraná
Setor de Ciências Exatas
Departamento de Estatística
Programa de Especialização em *Data Science* e *Big Data*

Reinaldo Mota Junior

# Exploração de algoritmos de classificação para reconhecimento de dispositivos da rede

**Curitiba**
**2020**

Reinaldo Mota Junior

# Exploração de algoritmos de classificação para reconhecimento de dispositivos da rede

Monografia apresentada ao Programa de Especialização em Data Science e Big Data da Universidade Federal do Paraná como requisito parcial para a obtenção do grau de especialista.

Orientador: Prof. André Ricardo Abed Grégio

Curitiba
2020

Especialização em Data Science e Big Data
Universidade Federal do Paraná
*dsbd.leg.ufpr.br*

# Exploração de algoritmos de classificação para reconhecimento de dispositivos da rede

Reinaldo Mota Junior[1]
Prof. André Ricardo Abed Grégio[2]

## Resumo

A Internet das Coisas deixou de ser uma promessa e já é uma das tecnologias emergentes que mais movimenta o mercado de Tecnologia da Informação. Junto dos seus bilhões de dispositivos amplamente disponíveis e comercializados, os riscos de segurança da informação envolvendo a falta de controle sobre os dispositivos IoT de uma rede propiciam a utilização de inteligência artificial para ajudar na identificação destes dispositivos, com o intuito de classificar com acurácia o dispostivo mediante análise do tráfego da rede e os padrões de comportamento dos dispositivos. Neste artigo, pretendo demonstrar com base em um dataset público compartilhado por Sivanathan et al.que pôde-se obter acurácia consideravelmente superior em um *pipeline* mais enxuto, utilizando outras *features* disponíveis no *dataset*.
**Palavras-chave**: iot, classificação, saco de palavras, aprendizado de máquina.

## Abstract

*The Internet of Things is no longer a promise and is already one of the emerging technologies that most moves the Information Technology market. Along with its billions of widely available and commercialized devices, information security risks involving the lack of control over a network's IoT devices enable the use of artificial intelligence to help identify these devices, in order to accurately classify the device by analyzing network traffic and device behavior patterns. In this article, I intend to demonstrate based on a public dataset shared by Sivanathan et al. that considerably greater accuracy could be obtained in a leaner pipeline, using other features available in the dataset.*
***Keywrods**: iot, classification, bag of words, machine learning.*

## 1 Introdução

A Internet das Coisas *(Internet of Things - IoT)* é um termo que define a possibilidade de conectar os mais variados tipos de objetos provenientes de mundos totalmente heterogêneos, virtual e físico, à Internet, fazendo com que haja interatividade entre eles [1].

Este novo paradigma deu origem a bilhões de dispositivos de baixo custo, como sensores de presença e temperatura, lâmpadas e câmeras a possibilidade de se conectarem à Internet, e junto de cerca de 50 bilhões outros dispositivos, constituem o que chamamos de coisas inteligentes *(smart things)* [2]. No entanto, o crescimento exponencial na quantidade destes dispositivos, nem sempre desenvolvidos por empresas com *know-how* e projetados para prover cibersegurança desde os requisitos até a implementação, resultou em um aumento na possibilidade de ataques e explorações de vulnerabilidades por meio da rede na qual eles estão conectados.
Muitas dessas vulnerabilidades, inclusive, já foram exploradas por atacantes, por exemplo os sensores do aquário de um cassino [3] e máquina de vendas de uma universidade [4]. Ambos os incidentes ocorreram nos Estados Unidos e evidenciaram a importância do monitoramento e da visibilidade dos dispositivos inteligentes de uma determinada rede sob a ótica da segurança da informação. Portanto, o grande número de novos dispositivos a cada dia, a facilidade com que são conectados e desconectados e sua suscetibilidade para ataques cibernéticos trazem à tona a dificuldade das organizações em monitorar os dispositivos *IoT* conectados em sua rede e além disso, de garantir com que eles estejam funcionando de acordo com o esperado.
Este mercado apresenta uma enorme quantidade de novos dispositivos *IoT* sendo comercializados e sua diversidade (sensores de temperatura, câmeras, lâmpadas, plugues, etc) que está cada vez maior, faz com que o reconhecimento do dispositivo seja de extrema importância e isso traz novos desafios para os administradores de rede [5].
No contexto acima, por exemplo, saber que há uma câmera de segurança de uma determinada fabricante conectada à rede, ajuda o administrador a especificar regras de segurança que não permitirão que a câmera faça algo que não se espera dela [6]. Esse tipo de reconhecimento também é utilizado para bloquear o acesso à rede de dispositivos considerados vulneráveis. Nesse momento, tendo em vista a enorme quantidade de dados produzidos por esses dispositivos e o advento do *big data* faz com que uma análise ou filtro manual sejam inviá-

[1]Aluno do programa de Especialização em Data Science & Big Data, reinaldomota.jr@gmail.com.
[2]Professor do Departamento de Informática - DInf/UFPR.

veis para os padrões atuais. Com isso, se faz necessária a utilização da tecnologia, em sua mais avançada forma: a utilização de algoritmos de *machine learning* focadas em classificação de dispositivos *IoT*. Há uma série de algoritmos de *machine learning* que podem ser utilizados para classificar os dispositivos, cada um com sua especificidade, tais como ***Random Forest, Decision Tree, SVM, k-Nearest Neighbors (KNN), Artificial Neural Network (ANN), Naïve Bayes, Logistic Regression e Stochastic Gradient Descent Classifier (SGD)***. Foi evidenciado por [6] a obtenção de 99.9% de acurácia na classificação dos dispositivos de seu estudo por meio de ***Random Forest***. Esse número é muito satisfatório e mostra a importância da utilização de *machine learning* para essa finalidade.
Nesse artigo, irei por meio da aplicação de algoritmos de classificação, identificar o mais acurado para a finalidade de identificação do dispositivo e utilizando um *dataset* público contendo o monitoramento do tráfego de rede dos dispositivos, comparar com os resultados existentes na literatura.

## 2 Trabalhos relacionados

Já existem alguns artigos na literatura que abordam a classificação de dispositivos *IoT* mediante o monitoramento da rede por um determinado período de tempo.
Conforme citado por [6], Y. Meidan et al. propuseram um modelo de aprendizado de máquina *(machine learning)* baseado na análise do tráfego de rede para identificar dispositivos *IoT* e criar *white lists* de *devices* autorizados a acessar a rede.
Sivanathan et al. [7], utilizaram, além da periodicidade temporal, outras características oriundas do tráfego de rede gerado por 28 dispositivos *IoT* para classificá-los, por meio de um *multi-stage machine learning* de duas etapas *(stage-0 e stage-1)*. No *stage-0*, aplicaram a técnica de saco de palavras *(bag of words)*, separadamente, em cada uma das três *features* (`Número de portas`, `Nomes de domínio` e cipher suite).
Esta técnica foi originalmente desenvolvida para representação de documentos, tendo como objetivo a definição de um livro de código contendo um conjunto de palavras-código e então, representar um documento como um histograma das palavras-código, onde cada entrada contém a contagem de uma palavra-código ocorrida no documento (chave/valor) [8].
Finalmente, aplica-se o algoritmo de classificação ***Naïve Bayes Multinomial*** em cada um destes três *bag of words*.
Para o *stage-1*, utilizaram o algoritmo ***Random Forest*** sobre cada resultado do *stage-0* visto que os atributos do *stage-1* não são linearmente separáveis e as saídas dos classificadores do *stage-0* são valores nominais, além do fato desse algoritmo possuir alta tolerância ao *over-fitting* comparado com outros classificadores baseados em árvore de decisão. Por fim, obtiveram acurácia de 99% ao final do *stage-1*, ou seja, ao analisar o tráfego de dados da rede simulada em laboratório, pôde-se classificar um dispositivo IoT com 99% de acurácia.
Porém, apesar de a literatura apresentar uma série de trabalhos sobre as características do tráfego de rede de uma maneira geral, os estudos relacionados à caracterização do tráfego de dados de dispositivos IoT ainda estão em sua infância [9].
Ao analisar a literatura, é possível identificar uma série de benefícios quanto a classificação de dispositivos *IoT* por meio da análise dos dados trafegados por eles. Dado acima, o objetivo do trabalho é apresentar opções acuradas de classificação dos dispositivos IoT e comparativo por algoritmo, por meio da aplicação de inteligência artificial.
Na literatura analisada não se comenta ou se compara sobre qual algoritmo de classificação possui uma maior acurácia na classificação dos dispositivos, deixando uma lacuna a ser estudada.

## 3 Premissas acerca do *dataset*

O *dataset* utilizado no desenvolvimento do artigo foi configurado por Sivanathan et al. [7] em um laboratório interno da Universidade de Sidney.
Nele simulam o funcionamento de uma *smart home* composta por 28 únicos dispositivos *IoT*, contendo câmeras de monitoramento, lâmpadas, plugues, sensores de presença, entre outros, e foi parcialmente disponibilizado pelos autores através do site https://iotanalytics.unsw.edu.au/iottraces.
O *dataset* original utilizado no referido artigo foi gerado a partir do tráfego de rede dos dispositivos *IoT* pelo período de seis meses, ou seja, durante todo esse período de tempo.
No entanto, a base disponibilizada e utilizada neste trabalho contém dados relacionados a 20 dias de monitoramento, totalizando 21 milhões e 61 mil linhas.

Para a realização deste artigo, foi feita uma amostra aleatória de 4 dias, sendo um dia por semana dentre os 20 dias disponibilizados. Essa seleção culminou num *dataset* de 3 milhões e 863 mil linhas.

## 4 Tratamento da base e Enriquecimento de dados

O *dataset* publicado por [7] possui 10 colunas, sendo elas populadas com as seguintes informações: ***Packet ID*** (identificação do pacote enviado pelo dispositivo), ***TIME*** (tempo de execução do pacote), ***Size*** (tamanho do pacote), ***eth.src (MAC Address*** do dispositivo de origem), ***eth.dst (MAC Address*** do dispositivo de destino), ***IP.src*** (endereço IP de origem), ***IP.dst*** (endereço IP de destino), ***IP.proto*** (protocolo de rede), ***port.src*** (número da porta do local de origem) e ***port.dst*** (número da porta do local de origem).
De modo a otimizar o desempenho das análises por seguintes, removemos do *dataset* as linhas cujo endereço de destino do pacote fosse um endereço IP local, tais como (192.168.x.x, 172.16.x.x e 10.x.x.x). Para a sequência do

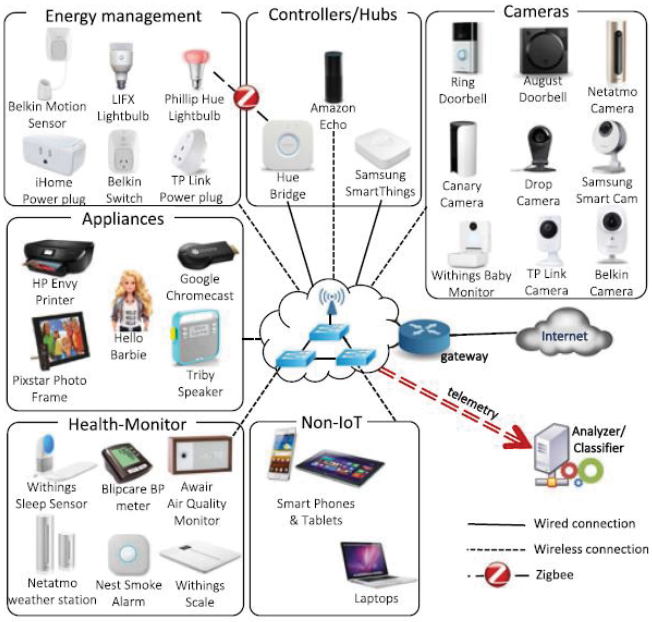


Figura 1: Arquitetura mostrando os 28 dispositivos *IoT* conectados no laboratório da Universidade de Sidney [7]

| Dispositivos | MAC ADDRESS | Tipo de conexão |
|---|---|---|
| Smart Things | d0:52:a8:00:67:5e | Wired |
| Amazon Echo | 44:65:0d:56:cc:d3 | Wireless |
| Netatmo Welcome | 70:ee:50:18:34:43 | Wireless |
| TP-Link Day Night Cloud camera | f4:f2:6d:93:51:f1 | Wireless |
| Samsung SmartCam | 00:16:6c:ab:6b:88 | Wireless |
| Dropcam | 30:8c:fb:2f:e4:b2 | Wireless |
| Insteon Camera | 00:62:6e:51:27:2e | Wired |
| Insteon Camera | e8:ab:fa:19:de:4f | Wireless |
| Withings Smart Baby Monitor | 00:24:e4:11:18:a8 | Wired |
| Belkin Wemo switch | ec:1a:59:79:f4:89 | Wireless |
| TP-Link Smart plug | 50:c7:bf:00:56:39 | Wireless |
| iHome | 74:c6:3b:29:d7:1d | Wireless |
| Belkin wemo motion sensor | ec:1a:59:83:28:11 | Wireless |
| NEST Protect smoke alarm | 18:b4:30:25:be:e4 | Wireless |
| Netatmo weather station | 70:ee:50:03:b8:ac | Wireless |
| Withings Smart scale | 00:24:e4:1b:6f:96 | Wireless |
| Blipcare Blood Pressure meter | 74:6a:89:00:2e:25 | Wireless |
| Withings Aura smart sleep sensor | 00:24:e4:20:28:c6 | Wireless |
| Light Bulbs LiFX Smart Bulb | d0:73:d5:01:83:08 | Wireless |
| Triby Speaker | 18:b7:9e:02:20:44 | Wireless |
| PIX-STAR Photo-frame | e0:76:d0:33:bb:85 | Wireless |
| HP Printer | 70:5a:0f:e4:9b:c0 | Wireless |
| Samsung Galaxy Tab | 08:21:ef:3b:fc:e3 | Wireless |
| Nest Dropcam | 30:8c:fb:b6:ea:45 | Wireless |
| Android Phone | 40:f3:08:ff:1e:da | Wireless |
| Laptop | 74:2f:68:81:69:42 | Wireless |
| MacBook | ac:bc:32:d4:6f:2f | Wireless |
| Android Phone | b4:ce:f6:a7:a3:c2 | Wireless |
| IPhone | d0:a6:37:df:a1:e1 | Wireless |
| MacBook/Iphone | f4:5c:89:93:cc:85 | Wireless |
| TPLink Router Bridge LAN (Gateway) | 14:cc:20:51:33:ea | |

Figura 2: Dispositivos, endereço físico e tipo de conexão encontrados no laboratório da *UNSW* [7]

trabalho, criamos outras duas colunas, sendo a *NomeDominio* e a *NomeServico*, que seriam duas de nossas *features* na execução dos algoritmos. Para alimentar a coluna *NomeDominio*, por meio da biblioteca *socket*, foi utilizada a função ***gethostbyaddr***, que ao ser chamada utiliza o parâmetro passado a ela ***(IP.dst)***(endereço IP de destino) e retorna uma tupla contendo o *hostname*, um alias para o endereço IP passado como parâmetro, se houver, e por fim, o endereço IP do *host*.
Selecionamos como retorno o *hostname* do ***IP.dst*** para posteriormente, o tratar e extrair dele o nome do domínio desse *host*. Para os casos em que não for encontrado o *hostname* do ***IP.dst***, alimentamos a coluna com "N/A" e assim como os IPs locais, removemos as linhas.
Após enriquecer o *dataset* com o *hostname*, foi feito o tratamento dessa coluna de modo a obter o nome do domínio relacionado a ele. Conforme evidenciado por Sivanathan et al.[7], os dispositivos *IoT* são distinguíveis pelos nomes dos domínios com os quais se comunicam, uma vez que é possível identificar que um dispositivo sempre acessa determinados domínios.
Na sequência, de modo a alimentar a coluna *"NomeServico"*, obtivemos o arquivo ***"/etc/service"*** do sistema operacional ***Linux*** e o utilizamos como uma base de nomes das portas de serviços utilizados pelos dispositivos, uma vez que cada linha desse arquivo funciona como uma referência a determinada porta. Também foi evidenciado em [7] uma relação entre os dispositivos e as portas de serviços acessados por eles, como por exemplo, dispositivos do mesmo fabricante compartilhando as mesmas portas.
Nota-se, portanto, um padrão no comportamento. Por fim, após atribuir os nomes dos serviços às portas, também removemos do *dataset* os nomes de portas não identificadas.
Com isso, após o enriquecimento de dados do *dataset* e a remoção das linhas que geram ruído na análise posterior, chegamos a uma base de um milhão, duzentos e vinte e três mil, duzentos e setenta e duas linhas.

## 5 Classificação baseada em Aprendizado de Máquina

A análise apresentada por [7] estrutura sua arquitetura em estágios ***(multi-stage classifier)***, onde o estágio 0 é composto por três sacos de palavras *(bag of words)*.
Os *bag of words* foram feitos utilizando as colunas "número das portas", "nomes dos domínios" e "*cipher suites*" com cada uma sendo a entrada de dados de um classificador baseado em ***Naïve Bayes Multinomial***.
Posteriormente, trataram o tráfego da rede composta pelo *flow volume* (formado pela *5-tuple* composta por ***srcIP, dstIP, srcPort, dstPort e Proto)***, *duration/rate* (calculado com base no volume de atividade apontado em *flow volume)*, *sleep time* (intervalo de tempo em que o dispositivo *IoT* não tem atividade de rede), *DNS interval* e *NTP interval* (intervalo de tempo para comunicação entre os servidores *DNS* e *NTP*) junto do *output* da primeira classificação como *features* do estágio 1, onde, por fim, aplicaram o algoritmo ***Random Forest*** para obtenção da acurácia na identificação dos dispositivos.
Na análise realizada neste artigo, diferentemente da literatura, foi realizada a conversão da coluna contendo as "portas", que são originalmente um campo do tipo numérico para os nomes dos serviços relacionados a cada

uma delas, e na sequência, o tratamento dos *hostnames* com base na utilização da função ***gethostbyaddr***.
Selecionamos essas duas *features* pois de acordo com a literatura [7], ambas são características que podem identificar o padrão comportamental de determinado dispositivo na rede. Desconsidera-se, portanto, a *feature cipher suites*, pois apesar de também poder identificar exclusivamente um dispositivo, não trouxe uma acurácia satisfatória tal qual muitos dispositivos *IoT* não usam serviços seguros de comunicação, sendo esse, inclusive, uma das principais características relacionadas à vulnerabilidade de segurança dos dispositivos.
Por fim, adicionamos as *features* `Size` (tamanho do pacote) e `IP.proto` (protocolo de rede) junto às demais citadas acima e iremos comparar a acurácia obtida usando três algoritmos de classificação em apenas um estágio, com os resultados obtidos por [7].

### 5.1 Estrutura de aplicação

Após as etapas de enriquecimento de dados e preparação do *dataset* com as novas colunas, foi necessário o tratamento dos campos `NomeDominio` e `NomeServico` com o intuito de transformar os valores categóricos em numéricos. Isso foi feito por meio do utilitário ***LabelEncoder*** da biblioteca *scikit-learn* [10]. Com isso, foi possível otimizar a aplicação dos algoritmos, visto que não houve alternância entre *datasets*.Posteriormente, foram definidas as divisões entre as bases de treino e teste, sendo 70% e 30% respectivamente.
Feito isso, deu-se início a fase de classificação dos dados por meio da utilização de três algoritmos, sendo ***MultinomialNB, svm.SVC e Random Forest***.
O intuito desta análise é identificar o algoritmo com a maior acurácia na classificação dos dispositivos IoT dentro da rede, ou seja, evidenciar que determinado dispositivo é ele mesmo com base em seu próprio histórico de comunicação com o roteador. Ao concluir a execução dos algoritmos, chegamos ao resultado mostrado na Tabela 1:

| | **Base de teste** | |
|---|---|---|
| | Stage único (Features: Size, IP.proto, NomeDominio e NomeServico) | Stage-0 (Features combinadas: Port Numbers, Domain Names e Cipher Suites) |
| **Algoritmos** | **Este** | **[7]** |
| Random Forest | 96.89% | - |
| C-Support Vector Classification (SVC) | 78.93% | - |
| Multinomial Naïve Bayes | 74.00% | 97.39% |

Tabela 1: Acurácia medida do experimento (Este) em comparação com os resultados obtidos da literatura ([7]).

Primeiramente, recordamos que foram utilizadas como *features* os campos `Size`, `IP.proto`, `NomeDominio` e `NomeServico`, ao passo que a literatura tratou de aplicar separadamente o algoritmo sobre cada um dos três *bag of words* supracitados.
O resultado da melhor classificação testada foi o ***Random Forest***, com 96.89% na base de teste, quase que a mesma obtida pela literatura ao final da última etapa (*stage-1*) de sua arquitetura de testes.

Já os algoritmos ***Naïve Bayes Multinomial*** e ***C-Support Vector Classification (SVC)*** apresentaram, respectivamente, acurácia de 47% e 78.93%. Sendo, portanto, valores inferiores aos obtidos pela literatura quando comparamos a acurácia do mesmo algoritmo, no caso do primeiro citado.

## 6 Conclusão

Neste artigo, diferentemente da abordagem utilizada por [7], propusemos aplicar diretamente sob o *dataset* amostral, ou seja, em um *pipeline* de execução mais enxuto, além do algoritmo de classificação ***Naïve Bayes Multinomial***, os algoritmos ***Random Forest*** e ***C-Support Vector Classification (SVC)***, utilizando somente um estágio. Adicionalmente, consideramos como *features* os campos `Size`, `IP.proto`, `NomeDominio` e `NomeServico` com o objetivo de identificar o algoritmo com a maior acurácia na identificação dos dispositivos *IoT*.
Os resultados obtidos nos mostraram uma acurácia consideravelmente superior ao encontrado na literatura utilizando ***Random Forest*** em apenas um estágio ao compararmos individualmente com cada uma das *features* utilizadas no *stage-0* da literatura. Também, obteve-se um resultado comparável com o resultado da classificação utilizando ***Random Forest*** com o resultado da combinação das *features* `Número de portas`, `Nomes de domínio` e cipher suite) da literatura. Cabe ressaltar perante os resultados que a literatura obteve acurácia superior ao final do *stage-1*, visto que após aplicar as classificações de maneira separada, ainda agregou mais *features* para enfim, chegar ao final de sua arquitetura de teste com acurácia de 99.88%.

Como possibilidade de trabalhos futuros, tendo em vista a acurácia obtida nos resultados deste artigo para um *dataset* real, porém, amostral, seria expandir a aplicação dos algoritmos para a base completa disponível, a fim de verificar se os resultados relacionados a acurácia sofrem alguma alteração.

## Agradecimentos

Gostaria de agradecer à minha esposa, que cuidou de nossa filha recém nascida para que eu pudesse desenvolver o trabalho, à família e amigos que me apoiaram ao longo de todo o curso de especialização, sobretudo na etapa final para a entrega do artigo de conclusão. Também gostaria de agrader ao meu orientador André Grégio por todo o suporte nesta fase final.

## Referências

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, *Internet of Things for Smart Cities*, (in IEEE Internet of Things Journal, vol. 1, no. 1, pp. 22-32, Feb. 2014) doi: 10.1109/JIOT.2014.2306328

[2] A. Nordrum, *"The internet of fewer things [News]"*, (in IEEE Spectrum, vol. 53, no. 10, pp. 12-13, October 2016) doi: 10.1109/MSPEC.2016.7572524.

[3] A. Schiffer, *"How a fish tank helped hack a casino, 2017"* Disponível: "https://www.washingtonpost.com/news/innovations/wp/2017/07/21/how-a-fish-tank-helped-hack-a-casino/"

[4] Ms. Smith, *"University attacked by its own vending machines, smart light bulbs & 5,000 IoT devices, 2017"* Disponível: https://www.csoonline.com/article/3168763/university-attacked-by-its-own-vending-machines-smart-light-bulbs-and-5-000-iot-devices.html

[5] Khan, M., & Salah, K. (2018) *"IoT security: Review, blockchain solutions, and open challenges. Future Gener. Comput. Syst., 82, 395-411."*

[6] M. R. Shahid, G. Blanc, Z. Zhang and H. Debar, *"IoT Devices Recognition Through Network Traffic Analysis, 2018"*, (IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 5187-5192, doi: 10.1109/BigData.2018.8622243.)

[7] Sivanathan, Arunan & Habibi Gharakheili, Hassan & Loi, Franco & Radford, Adam & Wijenayake, Chamith & Vishwanath, Arun & Sivaraman, Vijay. (2018). *"Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics."*, (IEEE Transactions on Mobile Computing. PP. 1-1. 10.1109/TMC.2018.2866249.)

[8] Wang, J., Liu, P., She, M., Nahavandi, S., & Kouzani, A. (2013). *"Bag-of-words representation for biomedical time series classification."*, (Biomed. Signal Process. Control., 8, 634-644.)

[9] Muhammad Zubair Shafiq, Lusheng Ji, Alex X. Liu, Jeffrey Pang, and Jia Wang. 2012. *"A first look at cellular machine-to-machine traffic: large scale measurement and characterization."*, (SIGMETRICS Perform. Eval. Rev. 40, 1 (June 2012), 65–76. DOI: https://doi.org/10.1145/2318857.2254767)

[10] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay *"Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011."*