Seleção de Características por Clusterização para Melhorar a Detecção de Ataques de Rede

Diego Abreu[1], **Igor Carvalho** [1], **Antônio Abelém**[1], **Daniel S. Menasché,** [2]
Rosa Leão[2], **Edmundo de Souza e Silva**[2]

[1] Grupo de Estudos em Redes Comunicação Multimídia - GERCOM - UFPA

[2]Universidade Federal do Rio de Janeiro - UFRJ

Abstract. *Machine Learning (ML) based Intrusion Detection Systems (IDSs) have come as a key tool to detect malicious traffic and network attacks. However, these approaches still struggle to detect different and constantly improving attacks. In this context, an open issue, among the required steps in ML-based evaluation, is the feature selection. It plays an important role to provide better efficiency in the detection of anomalies and network attacks. This paper aims to tackle this problem through a cluster-based feature selection approach in order to detect network attacks, as well as ranking the features that added to a high detection in each evaluated attack. Our approach outperformed all the other evaluated proposals for five different types of network attacks in terms of F1 score.*

Resumo. *Sistemas de Detecção de Intrusão (IDSs) baseados em aprendizado de máquina (AM) vêm sendo amplamente utilizados para detectar tráfego malicioso e ataques às redes. Entretanto, essas abordagens ainda apresentam grandes dificuldades para detectar os diferentes tipos de ataques que vêm se aprimorando. Neste contexto, uma questão em aberto, dentre os passos requeridos para uma avaliação baseada em AM, é a seleção de características. Essa etapa tem grande importância para propiciar maior eficiência na detecção de anomalias e ataques de rede. Este artigo propõe uma abordagem que realiza a seleção de características baseada em clusters para melhorar a detecção de ataques e tráfegos anômalos na rede. A proposta cria também um ranque com as características de tráfego que mais contribuíram para o incremento nos acertos dos algoritmos. Os resultados mostraram um desempenho superior às demais propostas avaliadas para cinco diferentes tipos de ataques, considerando a medida F1 score.*

1. Introdução

A detecção de tráfego malicioso e de ataques à rede tem, cada vez mais, se mostrado um desafio para os Sistemas de Detecção de Intrusão (*Intrusion Detection System - IDS*) atuais [Boutaba et al. 2018]. Como uma alternativa para solucionar este problema, a detecção de ataques baseada em Aprendizado de Máquina (AM) tem sido amplamente utilizada, devido à robustez e à autonomia que tais sistemas apresentam para aprender tanto, o comportamento do tráfego, por capturar possíveis alterações no padrão do tráfego, como encontrar correlações complexas nos dados de tráfego analisados [Boutaba et al. 2018].

A crescente sofisticação dos ataques e suas constantes mudanças, no entanto, vêm tornando o processo de seleção de características *(Feature Selection)* do tráfego bastante

desafiador e crucial para identificar anomalias, bem como detectar possíveis ataques na rede [Zuech and Khoshgoftaar 2015]. A grande quantidade de informações obtidas a partir da coleta de tráfego, os tipos de ataques realizados e os respectivos mecanismos utilizados por eles são alguns dos fatores que dificultam a seleção adequada de características do tráfego para uma alta detecção por parte dos algoritmos de AM [Zander et al. 2005].

Os métodos que realizam o processo de seleção de características podem ser classificados em filtro, *wrapper* e *embedded* [Guyon and Elisseeff 2003]. Na técnica de filtro, as características são selecionadas na etapa de pré-processamento, sem o auxílio de um classificador. Para avaliá-las, considera-se as seguintes métricas: ganho de informação *(information gain)*, que mede o quanto de "informação" uma dada característica provê a respeito de uma classe [Kullback and Leibler 1951]; razão do ganho de informação (*gain ratio*), que é uma modificação no ganho de informação para reduzir algum favoritismo injustificado por uma classe [Quinlan 1986]. Já a correlação, através do coeficiente de correlação, estima o grau de correlação entre um conjunto de características, bem como a inter-correlação entre as mesmas [Karegowda et al. 2010][Hall 2000]. Os métodos *wrapper* buscam selecionar um subconjunto de características usando um algoritmo de aprendizado de máquina, como um classificador. Assim, tal abordagem torna-se dependente do algoritmo utilizado. Por fim, a técnica *embedded* busca realizar o processo de seleção de características durante o treinamento do modelo. A principal desvantagem dos métodos *embedded* e *wrapper* é o alto custo computacional da seleção das características, que resulta da avaliação de cada subconjunto do atributo considerado [Kohavi and John 1997].

A técnica de clusterização é amplamente utilizada no contexto de aprendizado não supervisionado, para fazer agrupamento da base de dados. Na seleção de características ela costuma ser usada tanto previamente [Ni et al. 2017] à aplicação dos métodos apresentados anteriormente para a seleção das características quanto posteriormente à seleção, como classificador [Bisol et al. 2016]. Este artigo propõe utilizar a técnica de clusterização como o método para realizar a seleção de características. Nossa hipótese é que, utilizando medidas intrínsecas à formação dos *clusters*, é possível agrupar as características de tráfego mais relevantes para a detecção de ataques, independentemente do tipo de ataque realizado.

O trabalho objetiva discutir e responder as seguintes questões: (1) Quantas e quais características de fluxo são suficientes para detectar ataques com boa precisão? (2) Qual o comportamento do método proposto quando comparado com outros métodos tradicionais de seleção de características? (3) Que inferências sobre os diferentes ataques podem ser feitas a partir das características selecionadas? A proposta foi comparada com outros métodos em um estudo de caso de detecção de ataques em um cenário de rede doméstica criado em laboratório. Os resultados obtidos indicaram que a abordagem proposta apresentou maior precisão na detecção de ataques em relação às demais para todos os ataques avaliados.

O restante do artigo está organizado da seguinte forma: a seção 2 apresenta os trabalhos relacionados; a seção 3 detalha a proposta; a seção 4 apresenta o estudo de caso e a descrição do cenário de avaliação; a seção 5 apresenta os resultados obtidos e a discussão dos mesmos e a seção 6 conclui o artigo.

2. Trabalhos Relacionados

A utilização de métodos de seleção de características tem um grande potencial em melhorar o desempenho do aprendizado de máquina para a detecção de ataques. No trabalho proposto por Bisol et al. [Bisol et al. 2016], os autores fazem a utilização do algoritmo *Sequential Backward Selection* (SBS) para selecionar características de fluxo com objetivo de melhorar a classificação de tráfego. O SBS cria um subconjunto inicial com todas as características disponíveis. Em seguida, as características são removidas uma a uma, a fim de que, a cada iteração, é avaliado se o subconjunto tem seu desempenho melhorado ou não. Uma vez removida, tal característica não será adicionada novamente ao subconjunto. Por fim, este processo é finalizado quando todos os possíveis subconjuntos não alcançam um desempenho superior ao já obtido em iterações anteriores. Por conta disso, o SBS pode obter apenas um subconjunto de máximo local e não realmente o melhor subconjunto possível. Os autores utilizaram uma base de dados própria, gerada pelo emulador Mininet com dados de fluxo de rede, contendo tráfego considerado normal e tráfego de ataques de negação de serviço. Foram utilizadas trinta e três características criadas a partir de contadores nativos do OpenFlow. O desempenho SBS é comparado com o PCA *(Principal Component Analysis)* e com um algoritmo genético, como método de seleção de características [Haupt and Haupt 1998]. O PCA é um método de extração de características, o qual transforma os atributos originais da base de dados em combinações dos mesmos chamadas de componentes principais. Esses componentes podem ser então utilizados por classificadores no lugar dos atributos originais. No entanto, o PCA tem como desvantagem a perda da semântica contida nas características originais [Hyalika 2019]. Os melhores resultados, em termos da métrica acurácia, foram obtidos com o SBS e o PCA utilizados em conjunto com os algoritmos SVM (*Support Vector Machine*) e do *K-means* como classificadores.

Na proposta de Andreoni et al. [Lopez et al. 2017], os autores abordam a seleção de características por correlação baseado no trabalho de Hall [Hall 1999]. O algoritmo proposto utiliza o coeficiente de *Pearson* - (CFS - *Correlation Feateure Selection*) - para avaliar a correlação entre as características. Basicamente, o algoritmo considera a soma das correlações para atribuir um peso para cada característica. Quanto maior o peso, maior a independência linear entre o par de características avaliado. Como resultado, tem-se uma lista ordenada de características, em forma decrescente de correlação. A proposta foi comparada com os métodos PCA, SFS (*Sequential Feature Selection*), e a Seleção Recursiva de Eliminação por Máquinas de Vetores de Suporte *(Support Vector Machine Recursive Feature Elimination SVM- RFE)* e o ReliefF. Este último utiliza a diferença das distâncias das amostras mais próximas da mesma classe com as amostras mais próximas de uma classe diferente. Com isso, foi possível criar um ranking com as características mais importantes a serem utilizadas. A avaliação utilizou uma Árvore de Decisão, uma Rede Neural e o SVM como classificadores. O método proposto teve o melhor resultado, seguido do ReliefF e do PCA.

No trabalho de Guerra-Manzanares et al. [Guerra-Manzanares et al. 2019], é realizada uma comparação de desempenho entre métodos de seleção de características no contexto de detecção de *botnets* em redes IoT. Os autores utilizam uma base de dados criada por [Koroniotis et al. 2018] contendo ataques *botnets* em redes IoT, como Mirai

e o Bashlite, disponível no site do projeto[1]. Os métodos avaliados são: seleção por correlação; seleção baseado no *Fisher's score*; *Sequential Forward Feature Selection* e *Sequential Backward Feature Elimination*, métodos semelhantes ao SBS. Como classificadores são utilizados Floresta Randômicas, Árvore de Decisão e kNN. Os métodos são comparados em termos da métrica F1, obtendo melhorias significativas com a utilização dos métodos de seleção de características, sendo que as Floresta Randômicas obtêm os melhores resultados na maioria dos casos.

A proposta de seleção de características apresentada neste artigo é baseada na clusterização. A proposta se difere dos demais métodos por utilizar uma técnica não supervisionada, o que possibilita uma maior independência da base de dados de treinamento, ao remover a necessidade de se conhecer previamente as classes dos dados analisados, tornando-a mais eficiente em diferentes cenários. As abordagens que utilizam *infogain* e *gainratio*, por serem métodos simples, fazem o uso do conceito de entropia, podendo ter seu desempenho superado por outros métodos existentes. O PCA apresenta-se como um bom método, no entanto, ele não permite que os componentes gerados sejam interpretados como características, dificultando a análise. O método SBS faz uma pesquisa e gradualmente adiciona novas características por uma função de avaliação que minimiza o erro quadrático médio. A principal desvantagem do SBS é que, ao adicionar uma característica ao conjunto, o método não é capaz de removê-la se ela tiver maior erro, após a adição de outras. O método baseado em correlação (CFS - *Correlation Feateure Selection*) tem seu desempenho dependente do cenário avaliado e da diferença do valor de correlação entre as características. Por fim, o ReliefF, que é uma melhoria do método Relief, trata de classes múltiplas usando a técnica dos k-vizinhos mais próximos, diferentemente do Relief que só consegue lidar apenas com duas classes. No entanto, é um método supervisionado, o que o torna inadequado para aplicações de monitoramento de rede e detecção de ameaças, já que os fluxos de rede chegam aos classificadores sem rótulos, sendo necessário o uso de algoritmos não-supervisionados para tal.

3. Seleção de Características baseada em Clusters

Para selecionar as características que melhor capturam o comportamento do tráfego e permitam detectar ataques de forma mais eficiente, adotou-se uma abordagem baseada em clusterização. A clusterização é uma técnica de aprendizado de máquina não supervisionado, em que os dados são agrupados para formarem *clusters*, baseado em alguma medida de semelhança existente entre os dados. Um algoritmo bastante utilizado é o *K-Means*, que realiza o agrupamento dos dados de entrada em k *clusters*, considerando a distância entre um dado e o centroide do *cluster* [Jain and Dubes 1988]. Para a nossa proposta, foi utilizado o *K-Means*, tendo sido adotada a distância euclidiana para o cálculo de distâncias. Para um melhor entendimento do funcionamento da seleção de características por clusterização, a Figura 1 detalha o funcionamento da proposta, bem como os principais entes que geram a lista de características de tráfego selecionadas.

Primeiramente, os dados de tráfego são coletados para serem utilizados como entrada para o processo de seleção de características. A partir dessa base de dados, é realizado o processo de extração de características. Esse processo é necessário para que se conheça todas as características que poderão ser selecionadas pelo método proposto. A

[1]Disponível em "www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets"

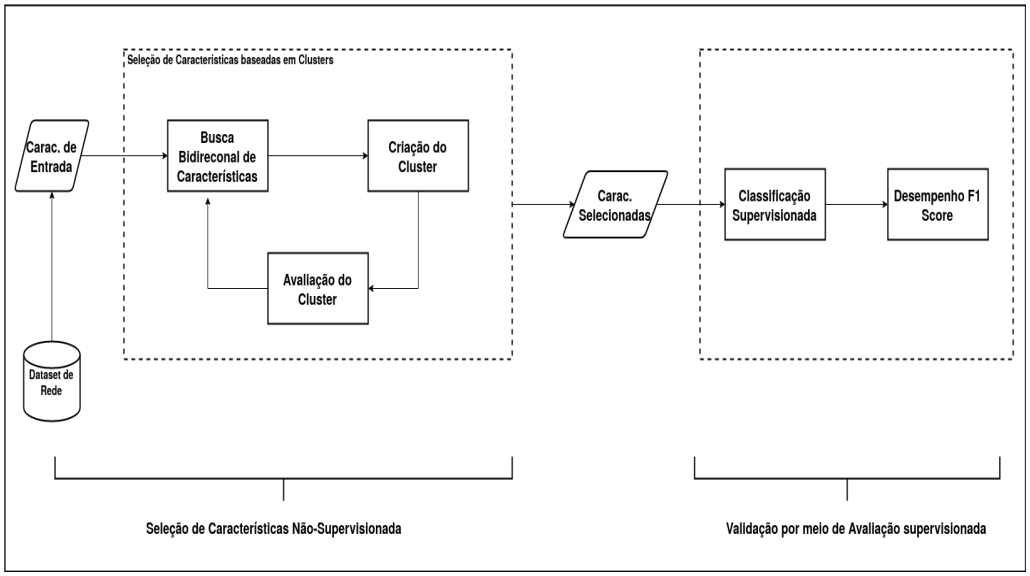

Figura 1. Funcionamento da Proposta

seleção de características baseada em *clusters* realiza a busca por características, gera os *clusters* e avalia o impacto que tais características causam na qualidade dos mesmos.

Inicialmente todas as características disponíveis são consideradas para criação dos *clusters*, utilizando o *K-means* para isso. Em seguida, avalia-se o impacto na qualidade do *cluster*, quando as características são adicionadas e/ou removidas. Como forma de avaliar a qualidade dos *clusters*, existem dois importantes índices, os quais são:

- **Índices Externos**: Baseiam-se em um conhecimento prévio sobre os dados. Isto implica que os resultados de um algoritmo de clusterização são avaliados considerando uma estrutura pré-definida, a qual é imposta no dataset, isto é, uma informação externa não contida no dataset.
- **Índices Internos**: Eles analisam a estrutura dos *clusters* utilizando informações envolvendo os vetores do próprio dataset, sem necessariamente haver um padrão esperado para os *clusters*. Além disso, utilizam critérios com relação à estrutura interna dos *clusters*.

Uma vez que a proposta utiliza uma técnica não-supervisionada, será utilizado como índice de avaliação da qualidade do *cluster* a *Silhouette*, que é do tipo interno. A *Silhouette* mede o quão similar um dado objeto é ao seu respectivo *cluster*, levando em consideração o outro *cluster*. Quanto maior a *Silhouette*, melhor será a configuração do *cluster*. O objetivo é criar o subconjunto de características que resultem em melhores *clusters*, em termos da métrica *Silhouette*.

Assim, utilizando a *Silhouette* como critério de avaliação do *cluster* gerado, é realizada a busca bidirecional, na qual as características podem ser adicionadas e removidas do subconjunto de características. A busca encerra quando se atinge a quantidade de características desejáveis. O subconjunto de características escolhido deve ser capaz de fornecer informações suficientes para que sejam criados *clusters* com maior valor de *Silhouette*. Desta forma, espera-se como resultado um subconjunto de característica que, também, maximize o desempenho dos classificadores.

Na segunda fase, após a realização da seleção de características, serão utilizados algoritmos de aprendizado de máquina como classificadores. O método de validação de resultados utilizado na avaliação será o *k-fold cross-validation*, que consiste em dividir a base de dados em k partes aleatórias de mesmo tamanho, sendo $k - 1$ partes para treino e 1 para teste. Este processo será repetido até que todas essas sub-partes sejam usadas uma vez com base de teste. Após isso, obtêm-se a média dos valores de teste. Como métrica de comparação dos resultados obtidos, será utilizada a medida F1 *score*. Essa medida é a média harmônica entre a precisão e revocação. A precisão é a razão entre os casos corretamente identificados como positivos e o número total de casos classificados como positivos. A revocação é a razão entre os casos corretamente identificados como ataque e o total de ataque (identificados ou não). Assim, utilizando a medida F1, considera-se não só a quantidade de acertos, mas também a quantidade de falsos positivos e falsos negativos.

4. Avaliação da Proposta

Para avaliar a proposta, foi realizado um estudo de caso de detecção de ataques em um cenário de rede doméstica criado em laboratório. O cenário consiste de variados dispositivos domésticos que serão alvos de cinco diferentes tipos de ataques. Com os dados obtidos, foram geradas 5 bases de dados, contento o tráfego de ataque e o tráfego do comportamento normal dos dispositivos. Para cada uma das bases de dados, será avaliado o funcionamento do método proposto em comparação com outros métodos de seleção de características. As características selecionadas em cada método serão utilizadas por classificadores para se obter um melhor desempenho na detecção dos ataques.

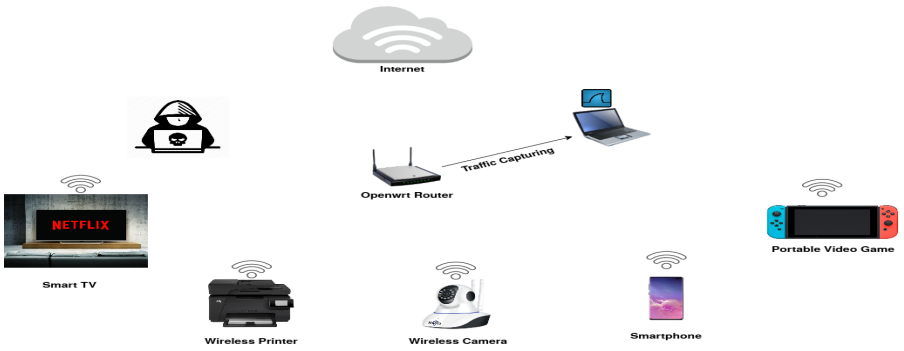

Figura 2. Cenário de Avaliação

4.1. Descrição do Cenário

A Figura 2 apresenta o cenário de experimentação utilizado, representando um cenário doméstico com a presença de dispositivos conectados à Internet. Os dispositivos utilizados foram: uma *smart tv* fazendo uso de *streaming* de vídeo, uma impressora, um video game portátil (*Nintendo Switch*), um *smartphone* e uma câmera IP. Os dados foram coletados a partir do espelhamento de porta de um roteador executando *Openwrt*, utilizando, para isso, a ferramenta *Wireshark*. O tempo para a coleta do tráfego de todos os dispositivos levou 60 minutos.

Para a coleta de tráfego de ataque foram realizados 5 diferentes tipos de ataques: *ACK Flood, ARP Spoofing, TCP SYN Flood, UDP Flood* e o Mirai. Os ataques foram

realizados a partir de uma máquina atacante fora da rede, direcionando o ataque para os dispositivos domésticos presentes no cenário de experimentação. Os ataques foram realizados ao longo do tempo de coleta do tráfego normal.

Os ataques *ACK Flood, TCP SYN Flood e UDP Flood* são ataques de negação de serviço que exploraram características de comportamento, respectivamente, dos protocolos TCP e UDP. O objetivo é "inundar" a vítima com requisições de conexões, o que gera a sobrecarga e a indisponibilidade de um dado serviço. Para o cenário avaliado, esses ataques foram gerados utilizando a ferramenta hping3 [2], e direcionados para os dispositivos domésticos apresentados na Figura 2. Cada um dos cinco ataques teve um tempo de duração de 30 segundos, porém, cada ataque teve parâmetros importantes alterados para tornar o cenário o mais real possível. Os parâmetros variados foram tamanho do pacote (0 e 120 bytes), tamanho do *payload* (0 e 120 bytes) e tempo de chegada entre os pacotes (*Flood*, a cada 5 segundos e a cada 10 segundos). O ARP Spoofing é um ataque do tipo homem no meio (*man-in-the-middle*) que envia mensagens ARP com o objetivo de associar o seu endereço MAC com o endereço IP da vítima, assim possibilitando a interceptação dos pacotes destinados a esse usuário. O ataque foi realizado utilizando a ferramenta Ettercap.[3] O Mirai é um ataque recentemente criado do tipo *botnet*, que tem como principal alvo dispositivos de IoT, assim como os dispositivos domésticos. O Mirai procura na rede dispositivos que estejam vulneráveis, adicionando-os a sua rede de "bots". A partir desse momento, o Mirai utiliza a sua rede de dispositivos infectados para realizar ataques de negação de serviço distribuídos (*Distributed Denial of Services Attacks*- DDOS). O ataque foi realizado utilizando o código fonte do Mirai disponível em repositório no Github [4]. Para cada um dos ataques, foi criada uma base de dados contendo os dados do ataque e os dados coletados dos dispositivos domésticos, os quais serão considerados de comportamento normal da rede. Assim, foram criadas 5 base de dados com 25% de dados de ataque e 75% de dados normais.

4.2. Extração de Características

Para cada uma das 5 bases criadas, foi realizado o processo de extração de características de fluxo utilizando a ferramenta *Netflowmeter Tool* [5]. Cada fluxo agrupou os pacotes que possuem a mesma combinação de IP de origem, IP destino, Porta de origem e Porta de Destino. A ferramenta gerou, originalmente, 86 características de rede para cada fluxo obtido. Entretanto, para o estudo de caso realizado não foram utilizadas características de identificadores do fluxo como a "hora do dia", o IP e porta de origem/destino, assim como características que sempre se apresentaram como nulas, nas bases de dados geradas. Dessa forma, restaram 56 características do fluxo, listadas na Tabela 1.

As características criadas incluem a *duração e tamanho do fluxo, quantidade e tamanho dos pacotes, taxas e tempo entre a chegada dos pacotes, entre outra características*. Além disso, foram utilizadas variações desses valores, a partir do cálculo da média, valor máximo e mínimo e desvio padrão. Por fim, foram obtidas características considerando as direções *forward* ou *backward* ou as duas direções do fluxo.

[2]Disponível em "http://www.hping.org/"
[3]Disponível em "https://www.ettercap-project.org/"
[4]Disponível em "https://github.com/jgamblin/Mirai-Source-Code"
[5]Disponível em "http://netflowmeter.ca/"

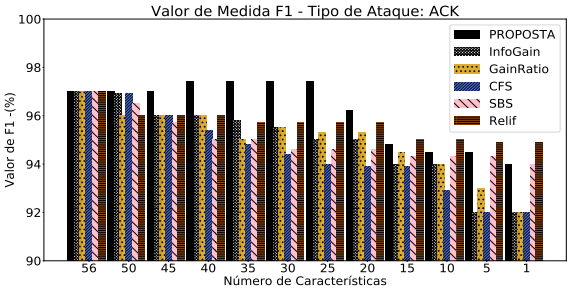

Figura 3. Desempenho (F1) x Nº de Caract. Ataque: ACK Flood

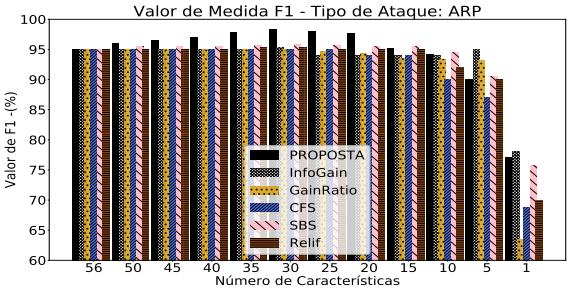

Figura 4. Desempenho (F1) x Nº de Caract. Ataque: ARP Spoofing

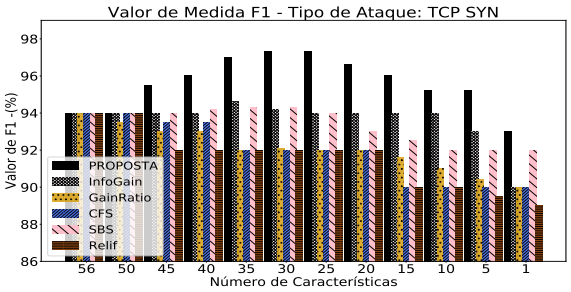

Figura 5. Desempenho (F1) x Nº de Caract. Ataque: TCP SYN Flood

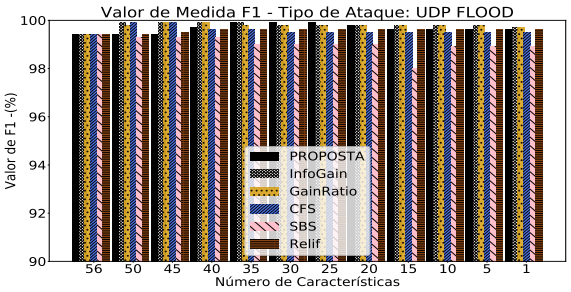

Figura 6. Desempenho (F1) x Nº de Caract.Ataque: UDP Flood

4.3. Seleção das características

A *maldição da dimensionalidade* aponta para o fato de que o desempenho de um classificador tende a diminuir a partir de uma quantidade de atributos, mesmo que eles possam ser considerados relevantes [Kouiroukidis and Evangelidis 2011]. Assim, adicionar novas características não significa que o desempenho de um classificador melhore. É necessário realizar um processo de seleção de características, para saber quais e quantos atributos devem ser utilizados para se tenha um desempenho adequado [Kouiroukidis and Evangelidis 2011].

Para este estudo de caso, foi verificado o desempenho das metodologias de seleção de características com a utilização de todas as 56 características. Tais metodologias buscam reduzir a quantidade de características a serem utilizadas no processo de aprendizado de máquina. Os métodos de seleção de características foram comparados entre si. Foram avaliados os seguintes métodos: a Proposta; SBS, conforme proposto em [Bisol et al. 2016]; CFS, com as alterações propostas em [Lopez et al. 2017]; ReliefF, conforme utilizado em [Lopez et al. 2017]; Ganho de Informação (*InfoGain*) [Kullback and Leibler 1951] e Razão do Ganho de Informação (*GainRatio)* [Quinlan 1986]. Os métodos de seleção de características foram aplicados em cada uma das 5 bases de dados criadas no experimento. Utilizou-se a separação da base de dados em 75% treino e 25% teste. Nesta análise, buscou-se avaliar qual o impacto no desempenho da métrica avaliada, quando o número de características variou de 56 até apenas 1 característica. Dependendo da quantidade de características iniciais a serem removidas e da qualidade das características utilizadas, essa redução pode ter impacto decisivo no desempenho da métrica.

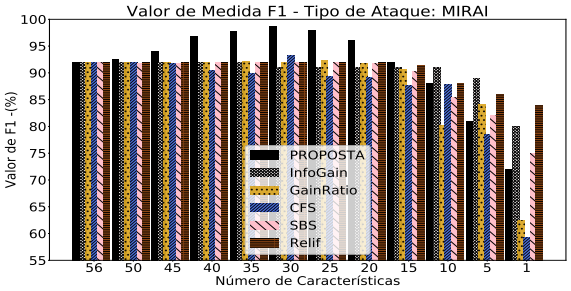

Figura 7. Desempenho (F1) x Nº de Caract. Ataque: Mirai

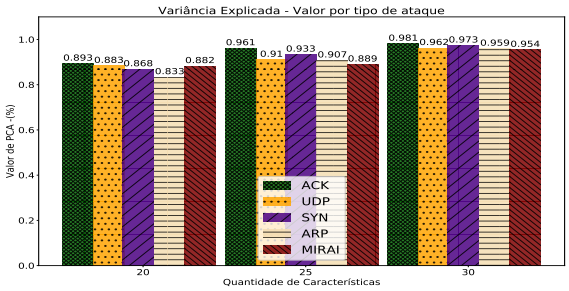

Figura 8. Variância Explicada para cada tipo de ataque

Considerando as Figuras 3 - 7, o desempenho foi avaliado através da medida F1 para o algoritmo de aprendizagem de máquina *Random Forest*, variando-se a quantidade de características selecionáveis em cada método investigado. Cada gráfico representa uma das bases de dados utilizadas. Utilizando todas as 56 características, todos os métodos de seleção de características possuem o mesmo desempenho, já que na prática não houve variação das características utilizadas. Reduzindo-se a quantidade de características, observa-se o desempenho do classificador tende a melhorar entre 20 e 30 características. Essa quantidade de características corrobora com resultados obtidos a partir do uso do PCA, conforme indicado na Figura 8, o qual foi utilizado para a criação de novos atributos ou componentes, a partir das características iniciais. Na faixa de 20 e 30 componentes é possível obter valores de variância explicada próximos ou acima de 90%. Em função disso, e para avaliar comparativamente as propostas abordadas, foi utilizada uma faixa de valores entre 20 e 30 características.

4.4. Aplicação dos Classificadores

Como forma de melhor avaliar o comportamento das características selecionadas na fase anterior, utilizou-se três algoritmos de aprendizado de máquina como classificadores: Floresta Randômica (*Random Forest*-RF)[Ho 1995], Árvore de Decisão (*Decision Tree*- DT) e algoritmo KNN (*K Nearest Neighbors*)[Altman 1992]. O KNN é constantemente apontado como uma das técnicas de classificação mais utilizadas por ser considerado simples de ser aplicado em diversos contextos [Okfalisa et al. 2017] [Agrawal 2014]. Esse método classifica objetos utilizando a distância entre os objetos, como a distância euclidiana, como medida de proximidade.

O algoritmo de Árvore de Decisão utilizado foi o C4.5 [Salzberg 1994]. Esse algoritmo escolhe o atributo dos dados que melhor divide o conjunto das amostras em subconjuntos de classificação. O critério para essa divisão é baseado no *InfoGain*, realizado internamente no algoritmo. Nas Florestas Randômicas são construídas múltiplas árvores de decisões. Cada árvore gera a sua predição e o modelo mais votado se torna o modelo para a classificação. É um método randômico devido ao fato de que cada árvore só pode escolher os atributos dos nós de forma aleatória [Ho 1995]. Para evitar o *overfitting*, a Árvore de Decisão e a Floresta Randômica tiveram suas profundidades limitadas a 10. Utilizou-se um valor de $k = 10$ para o *k-fold cross-validation*. Para cada uma das bases de dados, foi testada a utilização dos métodos de seleção de características, apresentados na Seção 4.3. A medida F1 *score* foi utilizada como medida de comparação dos resultados obtidos. Os resultados obtidos têm um intervalo de confiança de 95%.

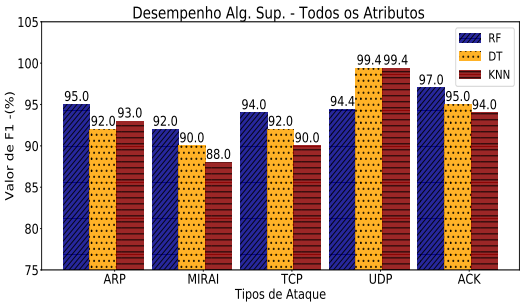

Figura 9. Medida F1 - Todos os Atributos

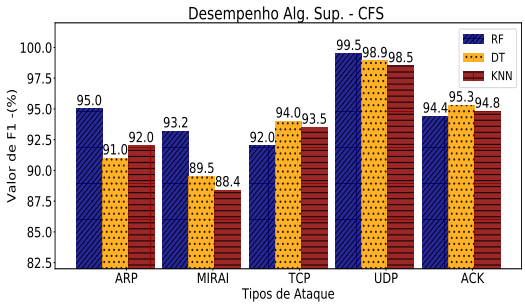

Figura 10. Medida F1 - CFS

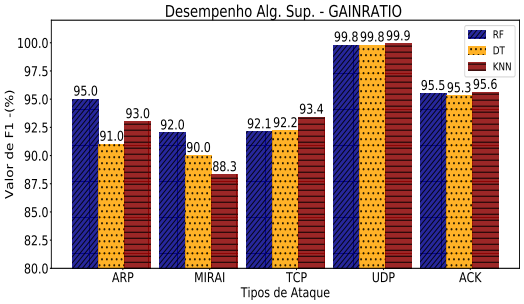

Figura 11. Medida F1 - GainRatio

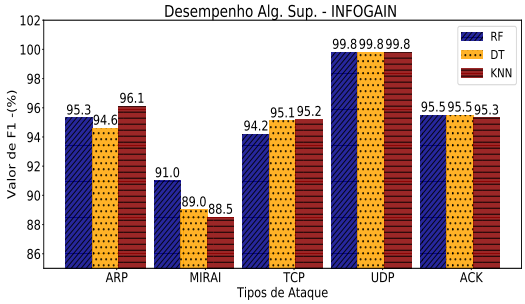

Figura 12. Medida F1 - InfoGain

5. Análise dos Resultados

Nesta seção, os resultados obtidos pelos algoritmos classificadores são apresentados para cada tipo de ataque para a medida F1. Primeiramente, são feitas as análises das figuras contendo o desempenho dos classificadores e, em seguida, a discussão dos resultados.

As Figuras 9-16 apresentam o desempenho dos três algoritmos classificadores para os cinco diferentes ataques avaliados, para cada método de seleção de características analisado (conforme descrito na Seção 4.3). A metodologia proposta teve o melhor desempenho em todos os cenários de ataques, conforme pode ser observado na Figura 14. Entretanto, observou-se que cada método de seleção de característica teve variações em termos de desempenho para cada um dos cenários avaliados. No caso do UDP, houve pouca variação entre os resultados obtidos. Porém, ainda assim a solução proposta melhorou o desempenho em relação à abordagem que não utiliza seleção de características (Figura 9, com todos os atributos). No caso do Mirai, pode-se observar uma melhora significativa do desempenho dos classificadores com a utilização da proposta em comparação com os outros métodos de seleção de características.

Em termos de desempenho dos classificadores, observa-se que o método de Floresta Randômicas teve os melhores resultados na maioria dos casos, de forma similar ao que foi obtido por [Guerra-Manzanares et al. 2019]. Esse resultado expressivo deve-se ao fato de se tratar de um dos métodos de *ensemble learning*, os quais são mais robustos e tendem a ter melhor desempenho [Opitz and Maclin 1999]. Entretanto, observa-se que em alguns casos, como no cenário UDP, métodos mais simples como o kNN e a Árvore de Decisão podem ter o mesmo desempenho e até resultados melhores do que as Floresta Randômicas.

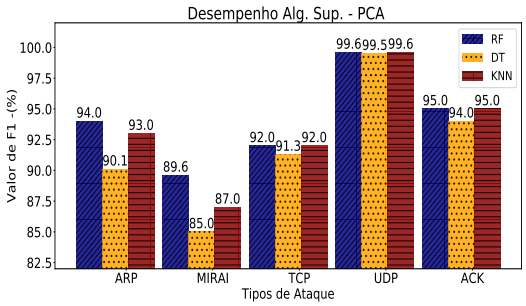

Figura 13. Medida F1 - PCA

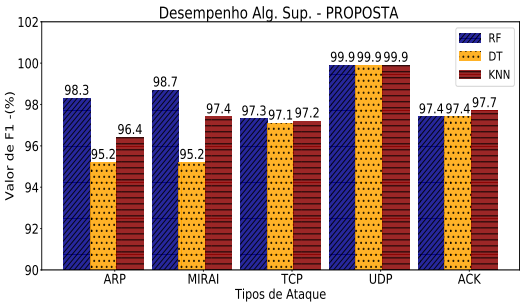

Figura 14. Medida F1 - PROPOSTA

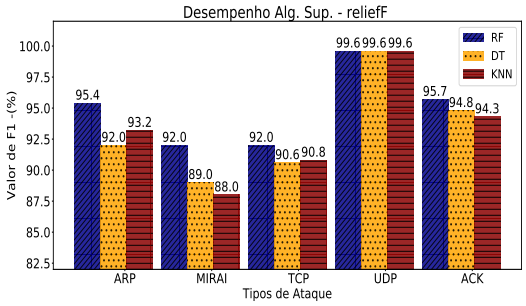

Figura 15. Medida F1 - reliefF

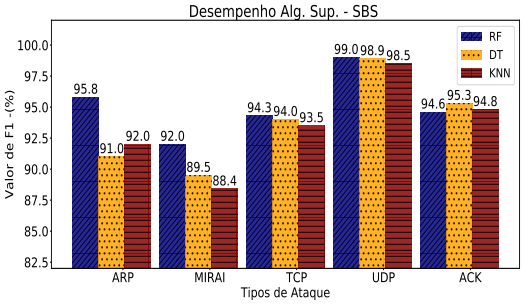

Figura 16. Medida F1 - SBS

A Figura 17 mostra o histograma da frequência de utilização de todas as 56 características disponíveis, considerando os cinco tipos de ataques realizados para todas as propostas avaliadas. Já na Tabela 2, observa-se que cada método seleciona características diferentes em cada base de dados. Nota-se também que, na mesma base de dados, cada método atribui ordem de importância diferente para as características. Esse resultado era esperado, já que cada método utiliza maneiras diferentes de avaliar a importância das características. Ainda assim, é possível observar algumas semelhanças nas características escolhidas por cada algoritmo e em cada base de dados.

As características relacionadas ao tamanho de pacote tiveram destaque na maioria das bases de dados, como na base relativa ao ataque ACK (35,37,36,10,13,38) e ARP (6,36,51,35). Embora os ataques realizados possuam uma ampla variação do tamanho dos pacotes, essa característica, quando considerada suas variações, (média, desvio padrão, máximo e mínimo), ainda mostra-se relevante para os métodos avaliados. Na base de dado relativa ao ataque ARP, também é possível observar que as características relativas ao tempo de atividade do fluxo (15,14,13) têm grande importância nos métodos avaliados. Já no caso da base de dados relativa ao ataque MIRAI, além do tamanho do pacote (6,35,9,10,36), as características relativas ao tempo de inatividade do fluxo (55,56) são colocadas constantemente entre as mais relevantes nos métodos de seleção de características avaliados.

Em termos gerais, nossa proposta conseguiu selecionar as melhores características para diferentes tipos de ataques. A escolha da seleção através de clusterização favoreceu o agrupamento dos dados baseado em medidas de semelhança entre eles. A adoção da *Silhouette* como índice de avaliação da qualidade do *cluster* utilizado, também contribuiu para criar *clusters* com características mais relevantes. Por fim, nossa proposta faz buscas

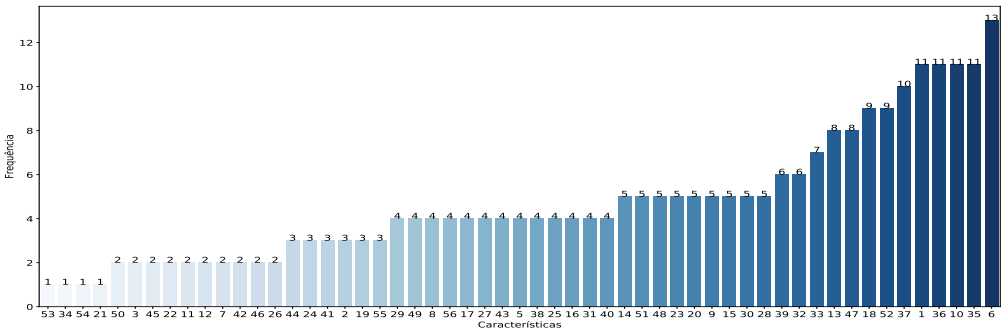

Figura 17. Freq. de utilização das características para a detecção dos ataques

Tabela 1. Lista de características Avaliadas

Característica	Descrição	Característica	Descrição
1 - Flow duration	Duração do Fluxo (μ s)	**29 - Bwd IAT Min**	Tempo Mínimo entre chegada dos Pacotes BWD
2 - Total FWD Packet	Total de Pacotes FWD	**30 - Fwd Header Length**	Total de Bytes FWD
3 - Total Bwd packets	Total de Pacotes FWD	**31 - Bwd Header Length**	Total de Bytes BWD
4 - Total Length of Fwd Packet	Tamanho Total de Pacotes FWD	**32 - FWD Packets/s**	Taxa de Pacotes FWD
5 - Total Length of Bwd Packet	Tamanho Total de Pacotes	**33 - FWD Packets/s**	Taxa de Pacotes FWD
6 - Fwd Packet Length Min	Tamanho Minimo do Pacote FWD	**34 - Min Packet Length**	Tamanho Mínimo do Pacote
7 - Fwd Packet Length Max	Tamanho Máximo do Pacote FWD FWD	**35 - Max Packet Length**	Tamanho Máximo do Pacote
8 - Fwd Packet Length Mean	Tamanho Máximo do Pacote FWD	**36 -Packet Length Mean**	Tamanho Médio do Pacote
9 - Fwd Packet Length Std	Desvio Padrão do Tamanho do Pacote FWD	**37 - Packet Length Std**	Desvio Padrão do Tamanho do Pacote
10 - Bwd Packet Length Min	Tamanho Minimo do Pacote BWD	**38 - Packet Length Variance**	Variância do Tamanho do Pacote
11 - Bwd Packet Length Max	Tamanho Máximo do Pacote BWD	**39 - Down/Up Ratio**	Taxa de Download e Upload
12 - Bwd Packet Length Mean	Tamanho Médio do Pacote BWD	**40 - Average Packet Size**	Média do Tamanho de Segmento
13 - Bwd Packet Length Std	Desvio Padrão do Tamanho do Pacote BWD	**42 -fAvgSegmentSize**	Média do Tamanho de Segmento FWD
14 - Flow Byte/s	Taxa de Bytes por segundo	**43 - bAvgSegmentSize**	Média do Tamanho de Segmento BWD
15 - Flow Packets/s	Taxa de Pacotes por segundo	**44 - sflow_fpacket**	Média de pacotes por fluxo FWD
16 - Média do Tempo entre pacotes	Média do Tempo entre pacotes	**45 -sflow_fbytes**	Média de bytes por fluxo FWD
17 - Flow IAT Std	Desvio Padrão do Tempo entre pacotes	**46 -sflow_bpacket**	Média de pacotes por fluxo BWD
18 - Flow IAT Max	Máximo Tempo entre pacotes	**47 - sflow_bbytes**	Média de bytes por fluxo BWD
19 - Flow IAT Min	Mínimo Tempo entre pacotes	**47 - sflow_bbytes**	Média de bytes por fluxo BWD
20 - Fwd IAT Total	Tempo Total entre pacotes FWD	**48 -Act_data_pkt_forward**	Contador de Pacotes com TCP
21 - Fwd IAT Mean	Média dos Tempos entre pacotes FWD	**49 - Active Mean**	Média do Tempo de Atividade do Fluxo
22 - Fwd IAT Std	Desvio do Tempo entre chegada dos Pacotes FWD	**50 - Active Std**	Desvio Padrão do Tempo de Atividade do fluxo
23 - Fwd IAT Max	Tempo Máximo entre chegada dos Pacotes FWD	**51 - Active Max**	Máximo Tempo de Atividade do fluxo
24 - Fwd IAT Min	Tempo Mínimo entre chegada dos Pacotes FWD	**52 - Active Min**	Mínimo Tempo de Atividade do fluxo
25 - Bwd IAT Total	Tempo Total entre pacotes BWD	**53 - Idle Mean**	Média do Tempo de Inatividade do Fluxo
26 - Bwd IAT Mean	Média dos Tempos entre pacotes BWD	**54 - Idle Std**	Desvio Padrão do Tempo de Inatividade do fluxo
27 - Bwd IAT Std	Desvio do Tempo entre chegada dos Pacotes BWD	**55 - Idle Max**	Máximo Tempo de Inatividade do fluxo
28 - Bwd IAT Max	Tempo Máximo entre chegada dos Pacotes BWD	**56 - Idle Min**	Mínimo Tempo de Inatividade do fluxo

bidirecionais, possibilitando que as características possam ser removidas/adicionadas do subconjunto em qualquer momento da busca.

6. Conclusões e Trabalhos Futuros

Esse artigo apresentou uma proposta de seleção de características por clusterização no contexto da detecção de ataques a redes domésticas. Mostrou-se que uma maior quantidade de características selecionadas nem sempre resultará em um melhor desempenho pelos classificadores. Os resultados também indicaram que não há um conjunto fixo de características mais relevantes para todos os tipos de ataques. Na verdade, as características mais relevantes são definidas em função dos tipos de ataques e do contexto em que eles ocorrem. A partir de um conjunto de características analisadas, a nossa proposta obteve um desempenho superior por realizar uma escolha de características mais qualificada na classificação do tráfego, quando comparada com as outras abordagens (SBS, CFS, ReliefF, InfoGain, GainRatio e PCA). Por fim, os resultados também mostraram que os diferentes tipos de ataques possuem particularidades que são melhor identificadas por diferentes características. Como trabalhos futuros, pretende-se realizar a avaliação da proposta em outros cenários, com maior diversidade de dispositivos e de ataques.

Tabela 2. Rank de Características - Ordem Decrescente de Importância

MÉTODO DE SELEÇÃO DE CARACTERÍSTICAS																														
Infogain						GainRatio					SBS					CFS					Relif					Proposta				
ARP	ACK	MIRAI	SYN	UDP	ARP	ACK	MIRAI	SYN	UDP	ARP	ACK	MIRAI	SYN	UDP	ARP	ACK	MIRAI	SYN	UDP	ARP	ACK	MIRAI	SYN	UDP	ARP	ACK	MIRAI	SYN	UDP
33	35	47	33	19	52	13	9	20	1	48	48	43	43	14	39	39	55	35	35	6	33	6	33	6	51	51	9	33	14
47	10	35	1	33	7	40	47	1	18	43	44	2	2	9	6	6	56	37	10	25	14	10	15	52	49	49	35	52	15
18	36	6	16	16	27	37	11	47	19	2	41	22	52	6	36	36	18	6	37	10	15	35	14	13	52	39	36	47	39
1	38	18	20	18	15	38	6	16	33	46	52	48	30	52	35	49	23	10	6	1	35	1	35	1	47	52	37	38	35
28	37	10	47	1	32	36	34	33	16	40	9	30	45	27	49	35	17	13	1	35	10	13	32	37	39	47	6	53	37
32	40	36	15	30	31	10	7	15	31	5	8	45	51	48	52	51	28	1	13	37	37	55	10	36	19	50	55	15	6
31	13	28	22	31	26	35	29	24	30	12	56	3	14	56	51	52	1	18	11	20	1	56	22	23	50	19	18	35	10
16	46	29	28	32	39	12	28	21	32	53	24	14	3	41	37	37	6	20	20	13	20	20	8	17	29	29	1	37	9
26	5	33	25	47	28	42	44	23	36	36	27	17	49	44	32	32	36	25	18	36	6	37	1	20	54	8	10	6	13
25	42	5	26	43	30	5	6	17	40	10	51	39	48	13	47	23	27	36	55	18	25	18	24	25	8	41	23	10	38

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, e da chamada cooperada RNP-NSF para pesquisa e desenvolvimento em segurança cibernética, através do projeto INSaNE (Improving Network Security at the Network Edge), financiado pela National Science Foundation (NSF) e pelo Ministério Brasileiro de Ciência, Tecnologia, Inovação e Comunicação (MCTIC), através da RNP e CTIC.

Referências

Agrawal, R. (2014). K-Nearest Neighbor for Uncertain Data. *International Journal of Computer Applications*, volume 105, páginas 13-16(11).

Altman, N. S. (1992). An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, volume 46, páginas 175-185(3).

Bisol, R., Silva, A., Machado, C., Granville, L., and Schaeffer-Filho, A. (2016). Coleta e Análise de Características de Fluxo para Classificação de Tráfego em Redes Definidas por Software. XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos.

Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., and Caicedo, O. M. (2018). A Comprehensive Survey on Machine Learning for Networking: Evolution, Applications and Research Opportunities. *Journal of Internet Services and Applications*, volume 46, página 16(1).

Guerra-Manzanares, A., Bahsi, H., and Nõmm, S. (2019). Hybrid Feature Selection Models for Machine Learning Based Botnet Detection in IoT Networks. In *2019 International Conference on Cyberworlds (CW)*, pages 324–327.

Guyon, I. and Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *The Journal of Machine Learning Research*, volume 3, páginas 1157-1182.

Hall, M. A. (2000). Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 359–366.

Haupt, R. L. and Haupt, S. E. (1998). *Practical Genetic Algorithms*. John Wiley & Sons, Inc., New York, NY, USA.

Ho, T. K. (1995). Random Decision Forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1)*, ICDAR '95, pages 278–, Washington, DC, USA. IEEE Computer Society.

Hyalika, H. (2019). Understanding Principal Components Analysis (PCA). *https://medium.com/datadriveninvestor/principal-components-analysis-pca-71cc9d43d9fb*, Dezembro 2019.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Karegowda, A. G., Manjunath, A., and Jayaram, M. (2010). Comparative Study of Attribute Selection Using Gain Ratio and Correlation Based Feature Selection. *International Journal of Information Technology and Knowledge Management*, volume 2, páginas 271-277(2).

Kohavi, R. and John, G. H. (1997). Wrappers for Feature Subset Selection. *Artificial Intelligence*, volume 97, páginas 273-324.

Koroniotis, N., Moustafa, N., Sitnikova, E., and Turnbull, B. (2018). Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset. *Future Generation Computer Systems*, volume 100, páginas 779-796.

Kouiroukidis, N. and Evangelidis, G. (2011). The Effects of Dimensionality Curse in High Dimensional knn Search. In *Informatics (PCI), 2011 15th Panhellenic Conference on*, pages 41–45.

Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, volume 22, páginas 79-86(1).

Lopez, M. A., Lobato, A., Mattos, D., Alvarenga, I., Duarte, O., and Pujolle, G. (2017). Um algoritmo não supervisionado e rápido para seleção de características em classificação de tráfego.

Ni, C., Liu, W.-S., Chen, X., Gu, Q., Chen, D.-X., and Huang, Q.-G. (2017). A cluster based feature selection method for cross-project software defect prediction. *Journal of Computer Science and Technology*, 32(6):1090–1107.

Okfalisa, Gazalba, I., Mustakim, and Reza, N. G. I. (2017). Comparative Analysis of K-nearest Neighbor and Modified K-nearest Neighbor Algorithm for Data Classification. In *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pages 294–298.

Opitz, D. and Maclin, R. (1999). Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, volume 11, páginas 169-198(1).

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, volume 1, páginas 81-106(1).

Salzberg, S. L. (1994). C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning*, volume 16, páginas 235-240(3).

Zander, S., Nguyen, T., and Armitage, G. (2005). Automated Traffic Classification and Application Identification Using Machine Learning. In *The IEEE Conference on Local Computer Networks, 30th Anniversary.*, volume 1,páginas 250-257. IEEE.

Zuech, R. and Khoshgoftaar, T. (2015). A Survey on Feature Selection for Intrusion Detection. *21st ISSAT International Conference on Reliability and Quality in Design*, páginas 150-155.