# Seleção de Características em Stream para a Detecção de Ataques de Rede

**Diego Abreu**[1,2], **Igor Carvalho** [1,2], **Antônio Abelém**[1,2]

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

[2]Universidade Federal do Pará - UFPA

***Abstract.*** *Streaming machine learning makes it possible to detect network attacks in real time by processing the instances and inspecting them only once. However, the network characteristics used by these systems are dynamic and may become redundant or irrelevant during the detection process, which directly impacts the performance of these algorithms, making it difficult to implement these proposals in real environments. In this work we propose the use of stream feature selection techniques to improve attack detection performance. Our results, using three security datasets show that feature selection directly impacts attack detection, with a reduction of up to 93% in detection time.*

***Resumo.*** *O aprendizado de máquina por stream possibilita a detecção de ataques de rede em tempo real, processando as instâncias e inspecionando-as apenas uma vez. Entretanto, as características de rede utilizadas por esses sistemas são dinâmicas e podem se tornar redundantes ou irrelevantes durante o processo de detecção, o que impacta diretamente no desempenho desses algoritmos, dificultando a implementação dessas propostas em ambientes reais. Neste trabalho propomos a utilização de técnicas de seleção de características em stream para melhorar o desempenho da detecção de ataques de rede. Nossos resultados, utilizando três bases de referência em segurança, mostram que a seleção de característica impacta diretamente na detecção dos ataques, com redução de até 93% no tempo de detecção.*

## 1. Introdução

Com o constante crescimento e diversidade das ameaças à segurança da rede, detectar ataques e proteger a rede tem se mostrado uma tarefa difícil [Zhang et al. 2021]. Um sistema de detecção de intrusão (*Intrusion Detection System* - IDS) monitora a rede para encontrar possíveis mudanças nas características de tráfego e a possível ocorrência de ataques à rede.

Um dos principais desafios enfrentados no monitoramento de rede em grande escala é o processamento e análise de grandes quantidades de dados heterogêneos e de fluxo constante. Os dados de monitoramento de rede vêm na forma de fluxos de alta velocidade, que precisam ser processados, e então analisados de forma rápida e contínua. No entanto, as diversas variações nas propriedades estatísticas da rede causadas por ataques tornam a análise de fluxo de dados de aprendizagem uma tarefa desafiadora [Casas et al. 2019].

A aplicação de modelos de aprendizagem para problemas de segurança de rede é geralmente tratada como uma tarefa de aprendizagem *off-line*, na qual os modelos são

treinados uma vez e, em seguida, aplicados ao fluxo de entrada de amostras. Esta abordagem é muito restritiva, principalmente ao lidar com ambientes altamente dinâmicos e adversários, nos quais a mudança de conceito (*concept drift*) ocorre frequentemente. Essa mudança de conceito se manifesta em alterações nas estatísticas do alvo de aprendizagem e de predição, e no fato de que os agentes maliciosos mudam constantemente sua estratégia de ataque para contornar os modelos de detecção construídos anteriormente. Como consequência, os modelos de aprendizagem rapidamente se tornam obsoletos, o que compromete a detecção correta dos ataques [Mulinka et al. 2018].

Os métodos de aprendizado de máquina (AM) por *stream* (fluxo de dados) consistem em processar os dados uma instância, ou conjunto de instâncias, de cada vez, e assim utilizando uma quantidade reduzida de memória e de tempo. Deste modo, os métodos de AM por *stream* possibilitam utilização dos modelos em tempo real, ou com um tempo de atraso ou janelamento pré-determinado. Assim, esses métodos são capazes de serem aplicados no contexto de detecção de ataques em tempo real, nos quais a predição correta dos ataques deve ser realizada no menor tempo possível [Casas et al. 2019].

Entretanto, as características de rede utilizadas por esses sistemas também são dinâmicas. Assim, durante o processo de detecção elas têm os seus valores alterados, o que impacta diretamente na sua relevância para a predição dos ataques. Em determinado momento do fluxo de dados, uma característica pode dar um ganho de informação importante ao detector e ser considerada relevante para o modelo de aprendizagem. Porém, devido à dinâmica da rede, essa mesma característica pode ter valores irrelevantes em um outro momento do fluxo de dados, dando informações que não são mais úteis ao detector, podendo inclusive induzi-lo para uma predição incorreta. Isso impacta diretamente no desempenho dos classificadores por *stream*, dificultando a implementação dessas propostas em ambientes reais [Mulinka and Casas 2018].

Neste trabalho propomos a utilização da seleção de características em *stream* para melhorar a detecção de ataques de rede. As técnicas de seleção de características possibilitam a readequação do subconjunto de características mais importantes a ser utilizado nos modelos de detecção, em tempo real, observando as variações existentes nas mudanças de conceito da rede. Será apresentada uma análise do desempenho de quatro técnicas de seleção de características em *stream* e o impacto que a sua utilização causa no desempenho na detecção de ataques. As bases de dados de referência em segurança NSL-KDD [Dhanabal and Shantharajah 2015], UNSW-NB15 [Moustafa and Slay 2015] e CICIDS17 [Sharafaldin et al. 2018] serão utilizadas para os experimentos.

O restante do artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados; a Seção 3 e 4 apresentam os métodos de classificação e de seleção por *stream*; a Seção 5 apresenta o estudo de caso, enquanto que a Seção 6 apresenta os resultados dos experimentos e as discussões; por fim a Seção 7 apresenta as conclusões do artigo.

## 2. Trabalhos Relacionados

Diversas técnicas de aprendizado de máquina por *stream* têm sido aplicadas para detectar padrões em tempo real e melhorar o desempenho de sistemas em diversos ramos do conhecimento. Nesta seção destacamos alguns trabalhos que aplicam essas técnicas ao contexto de detecção de ataques de redes de computadores, e comparamos com a nossa

proposta.

Moussa et al. (2019) [Moussa et al. 2019] propõe a utilização da seleção de características por *stream* para auxiliar a detecção de ataques de negação de serviço distribuídos (*Distributed Denial of Service Attacks* - DDoSs). Os autores realizaram experimentos com as bases de dados CTU-13 [Garcia et al. 2014] e CICIDS17, selecionando apenas os ataques DDoSs para realizar as suas análises. Utilizando como base a técnica de seleção Alpha-Investing [Zhou et al. 2005] (Alpha), um subconjunto de características específicas é selecionado periodicamente durante as ocorrências dos ataques. Desta forma, é possível analisar o comportamento dos ataques diante das mudanças dos subconjuntos, e a relação de dependência entre as características que se alteram durante o ataque. Ao final do processo, pôde-se obter um subconjunto otimizado, que posteriormente pode ser aplicado a um classificador de ataques. Em nosso trabalho, além de analisarmos diferentes tipos de ataques, a seleção de característica é realizada em conjunto ao método de detecção em *stream*.

Mulinka et al. (2018) [Mulinka et al. 2018] propõe a utilização dos algoritmos de aprendizado de máquina por *stream* Árvore Adaptativa de Hoeffding (*Hoeffding Adaptive Tree* - HAT) [Bifet and Gavalda 2009] e Floresta Randômica Adaptativa (Adaptive Random Forest - ARF) [Gomes et al. 2017] para a detecção de ataques de rede. O HAT e o ARF são implementados com a abordagem de janelamento *ADWIN(ADaptive WINdowing)* [Bifet and Gavaldà 2007], a qual mantém uma memória de amostras para o re-treino do modelo. Enquanto nenhuma mudança de conceito é detectada, o ADWIN aumenta de tamanho, porém, quando ocorre alguma redução da acurácia do modelo, a janela do ADWIN diminui, e o modelo de aprendizagem é re-treinado utilizando as amostras contidas nessa janela. Desta forma é possível adaptar o modelo ao novo comportamento da rede. A detecção de ataques pelos métodos de *stream* é avaliada utilizando a base de dados de segurança WIDE [Fontugne et al. 2010]. Neste trabalho também utilizaremos os algoritmos ARF e HAT com a abordagem de janelamento ADWIN para realizar a detecção de ataques. Entretanto, neste trabalho iremos realizar também o processo de seleção de características em conjunto com métodos de detecção em *stream*.

Schuartz et al. (2019) propõe uma comparação entre sistemas de detecção de ataques de redes, *stream* e em lotes (*batch*) [Schuartz et al. 2019]. Assim os algoritmos HAT e Bayes Ingênuo (*Naive Bayes* - NB) [Rish et al. 2001] são implementados tanto na versão em *stream* quanto em lotes. É utilizada a base de dados de segurança CICIDS17 para avaliar os métodos de detecção, nos cenários *stream* e em lotes. As características de rede são escolhidas de forma *off-line*, utilizando todo o conjunto de dados de treino e, em seguida, é realizada a detecção utilizando o NB e o HAT, considerando apenas o conjunto de características escolhidas.

Nosso trabalho se diferencia das demais propostas por considerar que o processo de seleção de características deve ser realizado ao mesmo tempo em que é realizado o processo de detecção dos ataques. Além disso, neste trabalho são feitos experimentos com três bases de dados de referência em segurança, cada uma contendo diversos tipos de ataques, quatro métodos de detecção em *stream* e quatro métodos de seleção de características. Desta forma, é possível fazer uma análise mais profunda do impacto da seleção de características na detecção dos ataques.

## 3. Métodos de Classificação por *Stream*

Nesta seção são apresentados os algoritmos de classificação por *stream* utilizados em nossos experimentos: Bayes Ingênuo, Árvore de Hoeffding [Hulten et al. 2001], Árvore Adaptativa de Hoeffding e Floresta Randômica Adaptativa. Esses algoritmos foram escolhidos por serem amplamente utilizados no contexto de detecção de ataques de rede por *stream* [Schuartz et al. 2019] [Al Nuaimi et al. 2019].

### 3.1. Bayes Ingênuo

O algoritmo de Bayes Ingênuo é um classificador probabilístico, no qual cada característica é considerada independente e não influencia no valor das demais, simplificando a predição das classes. Quando uma nova amostra chega ao sistema, o método calcula a probabilidade de cada característica pertencer a uma determinada classe. Assim, a classe que tiver a maior probabilidade é escolhida para a classificação da amostra [Schuartz et al. 2019].

O fato de não considerar a influência das características em conjunto faz com que o modelo gerado pelo NB não tenha uma predição tão robusta. Por outro lado, por ser um algoritmo considerado simples e rápido, o NB se torna uma opção viável para classificação em *stream* [Moraes et al. 2019].

### 3.2. Árvore de Hoeffding

A Árvore de Hoeffding (*Hoeffding Tree* - HT) é um algoritmo de aprendizado de máquina baseado em árvore de decisão incremental. Esse método foi projetado para aprender com um número limitado de amostras, facilitando a sua implementação em aplicações por *stream*. O HT se baseia no conceito da fronteira de Hoeffding, o qual quantifica um limiar de amostras necessárias para se realizar uma predição precisa.

Uma limitação do HT é que esse método assume que a distribuição dos dados não muda com o tempo. Desta maneira, o modelo criado pelo HT se torna suscetível a redução de precisão na presença de mudanças de conceito.

### 3.3. Árvore Adaptativa de Hoeffding

A Árvore Adaptativa de Hoeffding é uma modificação do HT proposta para lidar com a ocorrência da mudança de conceito. O HAT utiliza a janela adaptativa ADWIN para monitorar a performance dos galhos (*branchs*) das árvores de decisão, substituindo-os quando a performance deles diminui. Desta forma, com a chegada de novas amostras, o HAT atualiza o modelo de aprendizagem de acordo com o novo padrão dos dados. Assim, o HAT consegue se adaptar às mudanças na distribuição dos dados em *stream*.

### 3.4. Floresta Randômica Adaptativa

A Floresta Randômica Adaptativa é baseada no método de classificação Floresta Randômica (*Random Forest* - RF). O RF é um método de classificação que utiliza um conjunto de de árvores de decisão, as quais são combinadas para gerar um modelo de predição. O ARF foi projetado para adequar o RF para as mudanças de conceitos, utilizando a janela adaptativa ADWIN. Assim, a cada janela, o ARF treina um grupo de árvores de decisão, e combina os resultados individuais de cada árvore para gerar um modelo de predição mais robusto.

## 4. Detecção de ataques por *stream*

Para a tarefa de detecção dos ataques, os dados presentes nas bases de dados de segurança são interpretados como um fluxo contínuo de dados. O objetivo dos modelos de aprendizado de máquina é classificar os dados entre comportamento de ataque ou comportamento normal. A Figura 1 apresenta o processo de detecção em *stream*.

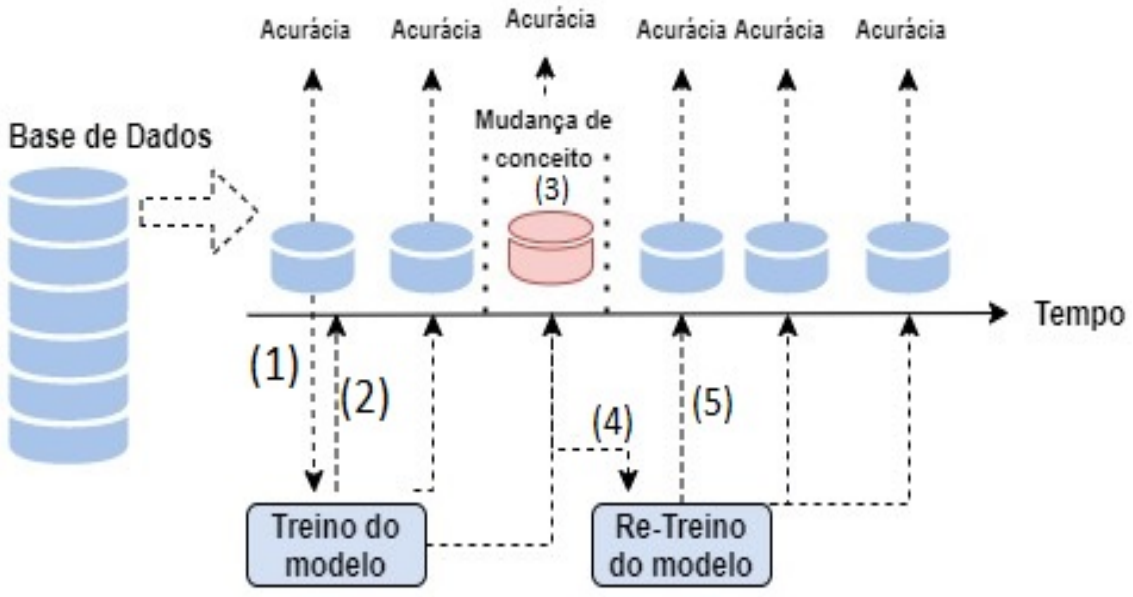


**Figura 1. Processo de detecção em *stream***

A primeira janela (1) de dados é utilizada para o treino do modelo de aprendizagem. Em seguida este modelo é aplicado nas próximas janelas (2), nas quais ocorre a predição dos dados, classificando-os de acordo com as classes treinadas. O tamanho da janela é regulado utilizando-se a técnica de janela adaptativa ADWIN. Assim, quando é detectada uma mudança de conceito (3) , é feito um novo treino (4) do modelo, a fim de adaptar esse modelo ao novo estado da rede. Com o modelo adaptado, é feita a predição para as próximas janelas (5). Este processo segue até que todos os dados sejam lidos.

O processo de seleção de característica é realizado a cada novo treino do modelo. Assim, é importante que esse processo seja rápido o suficiente para não atrasar o processo de adaptação do modelo. Além disso, a seleção de característica deve ajudar a criar modelos mais generalistas, que sejam capazes de realizar a predição mesmo com pequenas mudanças de conceito, e assim evitar o re-treino do modelo.

### 4.1. Seleção de características por *Stream*

Realizar a seleção de características em tempo real é fundamental para se ter um sistema mais rápido, preciso e que forneça modelos com melhor interpretabilidade. Assim, entre as técnicas de seleção de características por *stream* mais utilizadas destacam-se: Ganho de Informação (*Information Gain* - Infogain) [Azhagusundari et al. 2013], Seleção de Características Online (*Online Feature Selection* - OFS) [Wang et al. 2013] e Alpha-Investing [Al Nuaimi et al. 2019].

O Ganho de Informação é uma técnica de seleção de características que utiliza o conceito de entropia para avaliar a relevância de cada característica presente no conjunto de dados. Cada característica é analisada individualmente e recebe um valor relativo ao seu ganho de informação. Ao final do processo do Infogain é gerado um ranking com todas as características.

A técnica Seleção de Características Online é um método baseado em projeção esparsa para seleção de atributos em *stream*. O OFS foi projetado para lidar tanto em casos que o conjunto de dados tem seu domínio completamente conhecido, com todas as possíveis características já conhecidas, quanto quando parte desse domínio não está acessível inicialmente. A técnica identifica um subconjunto de características de acordo com o fluxo de dados, buscando características que não sejam redundantes ou irrelevantes.

O Alpha-Investing é uma adaptação da busca sequencial baseada em penalidade adaptativa. Esse método realiza dinamicamente o ajuste do valor de um limiar chamado *alpha* para diminuir a taxa de falsa descoberta. Com base nesse limiar, o método verifica se uma nova característica pode ser adicionada, ou não, ao subconjunto. Ao final da busca, é gerado um subconjunto com as características adicionadas.

Em nosso trabalho, além dessas técnicas, iremos utilizar a técnica de seleção de características não supervisionada baseada em clusterização (CFS) [Abreu et al. 2020]. Essa técnica utiliza uma busca bidirecional de características, medindo o impacto que cada característica causa na geração de agrupamentos, utilizando o algoritmo *k-mean*. Por se tratar de um método não supervisionado, não é necessário ter as informações dos rótulos dos dados, o que possibilita gerar um modelo menos tendencioso ao sobreajuste (*overfitting*) [Abreu et al. 2020].

## 5. Detecção de Ataques de Rede por *stream* - Estudo de Caso

Para avaliar o impacto da seleção de características por *stream* na detecção de ataques de rede foi realizado um estudo de caso utilizando bases de dados de segurança. Assim, para realizar os experimentos foi utilizada a ferramenta MOA (Massive Online Analysis) [Bifet et al. 2010]. O MOA é uma ferramenta que foi projetada especificamente para lidar com dados em *stream* e mudanças de conceito. Os algoritmos de aprendizagem de máquina por *stream* NB, HT, HAT e ARF foram implementados na ferramenta para se realizar a detecção dos ataques. Do mesmo modo, as técnicas de seleção de características Infogain, OFS, Alpha, e CFS também foram implementadas na ferramenta.

### 5.1. Descrição das Bases de Dados

Para este trabalho são utilizadas três bases de dados de referência em segurança de redes: NSL-KDD, UNSW-NB15 e CICIDS17. Essas bases estão entre as mais utilizadas no contexto de detecção de ataques através de técnicas de AM, e contêm uma grande diversidade de ataques e características de rede coletadas [Khraisat et al. 2019].

#### 5.1.1. Base de dados NSL-KDD

A base de dados NSL-KDD foi criada para ser uma base de referência em segurança. A base contém 41 características de rede, coletadas de quatro tipos de ataques: *Denial of Service Attack* (DoS), *User to Root* (U2R), *Remote to Local* (R2L) e ataques *Probe*.

A Tabela 1 apresenta a distribuição dos ataques na base NSL-KDD, e sua divisão de treino e teste. Para este estudo de caso, os dados de treino e teste serão utilizados em conjunto. Assim, todos os 148.517 atributos serão carregados na ferramenta MOA e interpretados como um único fluxo de dados.

**Tabela 1. Distribuição das classes dos dados na base NSL-KDD.**

| Rótulo | Treino | Teste | Total |
|---|---|---|---|
| Normal | 67.343 | 9.711 | 77.054 |
| DoS | 45.927 | 7.458 | 53.385 |
| U2R | 52 | 200 | 252 |
| R2L | 995 | 2.754 | 3.749 |
| Probe | 11.656 | 2.421 | 14.077 |
| Total | 125.973 | 22.543 | 148.517 |

### 5.1.2. Base de dados UNSW-NB15

UNSW-NB15 é um conjunto de dados de referência de segurança de rede criado para ser usado em testes de aprendizagem de máquina [Moustafa and Slay 2015]. A base de dados consiste de 100 GB de dados de fluxo de rede rotulados, incluindo dados considerados normais e dados provenientes de nove tipos diferentes de ataques. O UNSW-NB15 tem 257.663 instâncias e 48 características de rede. A Tabela 2 apresenta a distribuição das categorias de ataque e fluxo normal, para o conjunto de dados de treinamento e teste. Assim como no caso do NSL-KDD, para este estudo de caso os dados de treino e teste do UNSW-NB15 serão combinados e servirão de entrada para o processo de AM por *stream*.

**Tabela 2. Distribuição das classes dos dados na base UNSW-NB15.**

| Rótulo | Treino | Teste | Total |
|---|---|---|---|
| Normal | 56.000 | 37.000 | 93.000 |
| Generic | 40.000 | 18.871 | 58.871 |
| Exploits | 33.393 | 11.132 | 44.525 |
| Fuzzers | 18.184 | 6.062 | 24.246 |
| DoS | 12.264 | 4.089 | 16.353 |
| Reconnaissance | 10.491 | 3.496 | 13.987 |
| Analysis | 2.000 | 677 | 2.677 |
| Backdoor | 1.746 | 583 | 2.329 |
| Shellcode | 1.133 | 378 | 1.511 |
| Worms | 130 | 44 | 174 |
| Total | 175.341 | 82.332 | 257.673 |

### 5.1.3. Base de dados CICIDS17

CICIDS17 foi construído simulando o comportamento de 25 usuários nos protocolos HTTP, HTTPS, FTP e SSH. Os dados de fluxos de rede foram capturados durante uma

semana, em julho de 2017, e 84 características de rede foram extraídas. A Tabela 3 apresenta a distribuição por dia da semana do número de fluxos gerados pelo comportamento considerado normal e pelos ataques realizados. Para este estudo de caso, serão considerados todos os dias presentes no CICIDS17. Assim, todas as 2.830.743 instâncias serão carregadas na ferramenta MOA e processadas em sequência, de acordo com a ordem dos dias da semana.

**Tabela 3. Distribuição das classes dos dados na base CICIDS17.**

| Dia da Semana | Rótulo | Instâncias |
|---|---|---|
| Segunda | Normal | 529.918 |
| Terça-feira | Normal | 432.074 |
| | SSH-Patator | 5.897 |
| | FTP-Patator | 7.938 |
| Quarta-feira | Normal | 440.031 |
| | DoS Hulk | 231.073 |
| | DoS GoldenEye | 10.293 |
| | DoS Slowloris | 5.796 |
| | DoSSlowhttptest | 5.499 |
| | Heartbleed | 11 |
| Quinta-feira | Normal | 456.752 |
| | Web Attack - Brute Force | 1.507 |
| | Web Attack - Sql Injection | 21 |
| | Web Attack - XSS | 652 |
| | Infiltration | 36 |
| Sexta-Feira | Normal | 414.322 |
| | Bot | 1.966 |
| | Portscan | 158.930 |
| | DDoS | 128.027 |
| | Total | 2.830.743 |

### 5.2. Metodologia e Métricas de Avaliação

Para se avaliar os resultados do estudo de caso foi utilizada a metodologia de avaliação sequencial preditiva (*predictive sequential method*) [Dawid 1992]. Essa metodologia é frequentemente utilizada para se analisar dados em distribuições não estacionárias, como ocorre em nosso estudo de caso [Gama et al. 2013].

Na avaliação sequencial preditiva, o erro de cada predição é calculado e acumulado em uma janela de amostras. Na ferramenta MOA, este erro preditivo é utilizado para se calcular a métrica acurácia sequencial preditiva (*prequential accuracy*). Para este trabalho, iremos utilizar esta acurácia para avaliar o desempenho da detecção dos ataques. Além disso, o tempo de detecção dos ataques também será usado como métrica de avaliação. Os resultados gerados pela ferramenta têm um intervalo de confiança de 95%. Os experimentos foram realizados utilizando uma máquina com processador Intel Core i5-5200U com 2.20 GHz e 8 GB de memória RAM, utilizando o sistema operacional Windows 10 x64.

## 6. Resultados e Discussões

Nesta seção são apresentados os resultados da detecção de ataques utilizando os métodos de AM em *stream* e com as técnicas seleção de características em *stream*. Em todas as bases de dados a redução proposta foi de 75% no número de características. Assim, para a base NSL-KDD as técnicas de seleção de características escolhem 10 das 41 características; para a base UNSW-NB15 os subconjuntos têm tamanho 12, reduzido de 48 atributos; e para a base CICIDS17, os subconjuntos têm tamanho 21, de 84 características originais.

É importante destacar que esses subconjuntos de características são dinâmicos e as características escolhidas em diferentes momentos do fluxo de dados se alteram, apenas o tamanho do subconjunto é fixo. Portanto, com essa redução significativa da quantidade de características é possível avaliar o impacto das técnicas de seleção de características.

A Tabela 4 apresenta o resultado obtido sem a utilização do processo de seleção de características por *stream*. Podemos observar os resultados em termo das métricas acurácia e tempo obtidos com os algoritmos NB, HT, HAT e ARF nas bases de dados NSL-KDD, UNSW-NB15 e CICIDS17.

Na Tabela 4, podemos observar que o método ARF teve o melhor desempenho em acurácia, em todas as bases de dados, seguido do HAT e HT, e por último do NB, que teve o pior desempenho. Por outro lado, temos que o tempo do ARF foi substancialmente maior do que os outros métodos, em todas as bases de dados. Isso ocorre já que o ARF cria diversas árvores de decisão, criando um modelo mais preciso que os outros métodos. Porém, como os resultados demonstram, este processo é mais demorado e afeta a eficiência do ARF. Assim, para as bases utilizadas neste experimento, os algoritmos HT e HAT apresentaram um melhor equilíbrio (*trade-off*) entre acurácia e tempo, já que com estes métodos é possível ter resultados de acurácia similares ao ARF e com um tempo menor.

**Tabela 4. Resultados da detecção dos ataques - utilizando todas as características disponíveis - métricas acurácia (%) e tempo em segundos.**

| | Acurácia | | | | Tempo | | | |
|---|---|---|---|---|---|---|---|---|
| Base de dados | NB | HT | HAT | ARF | NB | HT | HAT | ARF |
| NSL-KDD | 87,51 | 93,71 | 95,75 | **98,20** | 6,34 | 6,62 | 19,45 | 43,50 |
| UNSW-NB15 | 81,74 | 95,64 | 96,94 | **98,77** | 9,63 | 6,86 | 8,87 | 26,19 |
| CICIDS17 | 89,06 | 96,73 | 96,57 | **98,88** | 106,5 | 150,5 | 172,4 | 321,7 |

As Tabelas 5 e 6 apresentam respectivamente os resultados obtidos com a utilização dos métodos de seleção de características por *stream*, nas métricas acurácia e tempo. Na Tabela 5, temos novamente que o método ARF teve, na maioria dos casos, o melhor desempenho em acurácia. Porém, temos que a diferença de acurácia com relação ao HT e HAT diminuiu, sendo que em alguns casos, como na base NLS-KDD utilizando tanto o Infogain como o CFS, os métodos HT e o HAT tiveram uma melhor acurácia que o ARF. Além disso, temos que para a bases NSL-KDD e UNSW-NB15 as maiores acurácias foram obtidas na combinação do CFS com o ARF e HAT, respectivamente. Já para a base CIDIDS17, as maiores acurácias foram na combinação do ARF com o Infogain e com o OFS.

**Tabela 5. Resultados da detecção dos ataques - utilizando as técnicas de seleção de características - métrica acurácia (%).**

| | Alpha | | | | OFS | | | |
|---|---|---|---|---|---|---|---|---|
| Base de dados | NB | HT | HAT | ARF | NB | HT | HAT | ARF |
| NSL-KDD | 85,99 | 88,44 | 87,27 | 88,84 | 84,51 | 87,45 | 91,40 | 91,42 |
| UNSW-NB15 | 74,62 | 89,18 | 88,36 | 88,22 | 80,93 | 89,37 | 89,90 | 90,55 |
| CICIDS17 | 78,21 | 81,40 | 81,41 | 98,79 | 80,02 | 86,33 | 81,47 | 99,22 |
| | Infogain | | | | CFS | | | |
| Base de dados | NB | HT | HAT | ARF | NB | HT | HAT | ARF |
| NSL-KDD | 86,16 | **94,64** | **92,83** | 92,22 | 86,97 | **97,77** | **94,30** | 93,83 |
| UNSW-NB15 | 49,84 | 90,09 | 90,18 | 89,76 | 82,01 | 90,3 | 97,76 | 94,45 |
| CICIDS17 | 73,49 | 90,22 | 76,62 | 99,33 | 85,84 | 95,1 | 94,60 | 98,30 |

**Tabela 6. Resultados da detecção dos ataques - utilizando as técnicas de seleção de características - métrica Tempo em segundos.**

| | Alpha | | | | OFS | | | |
|---|---|---|---|---|---|---|---|---|
| Base de Dados | NB | HT | HAT | ARF | NB | HT | HAT | ARF |
| NSL-KDD | 1,75 | 1,68 | 1,36 | 1,18 | 2 | 2,6 | 2,03 | 2,94 |
| UNSW-NB15 | 1,78 | 1,7 | 1,97 | 1,88 | 2,68 | 3,46 | 3 | 5 |
| CICIDS17 | 15,19 | 22,5 | 26,4 | 20,84 | 50,24 | 53,16 | 61,99 | 133,82 |
| | Infogain | | | | CFS | | | |
| Base de Dados | NB | HT | HAT | ARF | NB | HT | HAT | ARF |
| NSL-KDD | 1,99 | 3,07 | 2,42 | 3,68 | 2,84 | 4,15 | 6,03 | 11,04 |
| UNSW-NB15 | 2,95 | 3,27 | 2,32 | 3,1 | 3,89 | 2,88 | 9,98 | 5,27 |
| CICIDS17 | 51,29 | 62,43 | 61,91 | 129,22 | 18,06 | 20,53 | 33,84 | 91,36 |

Na Tabela 6, podemos observar os resultados de tempo de detecção dos ataques, agora utilizando a seleção de características. Assim como ocorre com a acurácia, os resultados dos métodos de classificação são mais próximos entre si, do que sem a utilização da seleção de características. Além disso, podemos ver que o método Alpha teve, no geral, os menores tempos entre os métodos de seleção, seguido do Infogain e do OFS. O CFS teve o maior tempo nas bases NSL-KDD e UNSW-NB15, na maioria dos casos, porém, na base CICIDS17 teve um tempo melhor que o Infogain e o OFS, e próximo ao tempo do Alpha.

As Tabela 7 e 8 apresentam o impacto em percentual da redução ou crescimento, respectivamente, da acurácia e do tempo a partir da utilização da seleção de características, comparado com os resultados obtidos sem a seleção de características. Na Tabela 7 temos, para os métodos Alpha, OFS, Infogain e CFS, o impacto da seleção das características na acurácia. Por exemplo, temos que, para a combinação do NB com o método de seleção Alpha houve uma redução de 2% na acurácia com relação ao desempenho obtido com o NB sem nenhum método de seleção de característica. Assim, de forma geral, podemos observar que a acurácia da detecção dos ataques reduziu com a utilização dos métodos de seleção de características. Por outro lado, nas bases NSL-KDD

**Tabela 7. Diferença percentual (%) da acurácia com e sem a utilização das técnicas de seleção de características.**

| | Alpha | | | | OFS | | | |
|---|---|---|---|---|---|---|---|---|
| Base de Dados | NB | HT | HAT | ARF | NB | HT | HAT | ARF |
| NSL-KDD | -2% | -6% | -9% | -10% | -3% | -7% | -5% | -7% |
| UNSW-NB15 | -9% | -7% | -11% | -11% | -1% | -6% | -9% | -8% |
| CICIDS17 | -12% | -16% | -16% | 0% | -10% | -11% | -15% | 0% |
| | Infogain | | | | CFS | | | |
| Base de Dados | NB | HT | HAT | ARF | NB | HT | HAT | ARF |
| NSL-KDD | -2% | **1%** | -3% | -5% | -1% | **4%** | -1% | -4% |
| UNSW-NB15 | **-39%** | -6% | -9% | -9% | 0% | -6% | -1% | -4% |
| CICIDS17 | -18% | -7% | **-21%** | 0% | -4% | -2% | -2% | -1% |

e UNSW-NB15 esta redução pode ser considerada pequena, em vários casos menor que 5%, sendo que em algum casos a acurácia teve um pequeno aumento como no caso da combinação de HT com o Infogain e com o CFS, na base NSL-KDD.

Com relação aos métodos de seleção de características, temos que com o CFS a redução da acurácia foi menor, sempre abaixo dos 6%. Por outro lado, com o método Infogain, a redução da acurácia foi significativa em alguns casos, como na base UNSW-NB15 com o NB e na base CICIDS17 com o HAT. Os métodos Alpha e o OFS também tiveram algumas reduções mais significativas, acima de 10%. Assim, temos que o CFS demostra ser um método mais robusto, e pode ser aplicado em conjunto com os classificadores em *stream*, sem ter uma redução de acurácia significativa.

**Tabela 8. Diferença percentual (%) do tempo de detecção com e sem a utilização das técnicas de seleção de características.**

| | Alpha | | | | OFS | | | |
|---|---|---|---|---|---|---|---|---|
| Base de Dados | NB | HT | HAT | ARF | NB | HT | HAT | ARF |
| NSL-KDD | -72% | -75% | -93% | -97% | -68% | -61% | -90% | -93% |
| UNSW-NB15 | -82% | -75% | -78% | -93% | -72% | -50% | -66% | -81% |
| CICIDS17 | -86% | -85% | -85% | -94% | -53% | -65% | -64% | -58% |
| | Infogain | | | | CFS | | | |
| Base de Dados | NB | HT | HAT | ARF | NB | HT | HAT | ARF |
| NSL-KDD | -69% | -54% | -88% | -92% | -55% | -37% | -69% | -75% |
| UNSW-NB15 | -69% | -52% | -74% | -88% | -60% | -58% | 13% | -80% |
| CICIDS17 | -52% | -59% | -64% | -60% | -83% | -86% | -80% | -72% |

Na Tabela 8, podemos observar que com a utilização dos métodos de seleção de características o tempo teve uma diminuição significativa. Isso significa, que mesmo com a adição de um processo a mais, a seleção das características, no treino e re-treino dos modelos, o tempo para realizar a detecção dos ataques diminui.

Isso confirma a nossa hipótese, de que com a redução da dimensionalidade, o treino fica mais rápido e, com a seleção das características mais adequadas, os modelos criados são mais robustos às variações causadas pelas mudanças de conceito. Assim, o re-treino do modelo é evitado, o que reduz ainda mais o tempo de detecção.

Assim, temos que com as técnicas de seleção de características por *stream*, a acurácia da detecção dos ataques, na maioria dos casos, tem uma pequena redução. Porém, ocorre uma redução significativa no tempo da detecção dos ataques. Desta forma, é possível realizar a detecção dos ataques, de forma mais rápida e sem ter uma mudança significativa na acurácia.

## 7. Conclusão e Trabalhos Futuros

A detecção de ataques de rede é uma tarefa desafiadora, principalmente quando se leva em consideração as diversas mudanças no comportamento da rede. Esse artigo apresentou a utilização de técnicas de seleção de características por *stream* como forma de melhorar a detecção de ataques. Foi apresentado um estudo de caso com três bases de dados de referência em segurança, nas quais quatro métodos de classificação por *stream* foram testados, com e sem técnicas de seleção de características. Nossos resultados indicam que a seleção de características causa impacto positivo na detecção dos ataques, reduzindo o tempo de detecção em até 93%. Além disso, nossos resultados mostram que a redução na acurácia da detecção pode ser pequena em comparação com o ganho de desempenho no tempo detecção, dependendo da técnica de seleção utilizada.

Como trabalhos futuros, pretende-se realizar a avaliação do impacto das técnicas de seleção de características por *stream*, alterando fatores como a estratégia de janelamento e o número de características a ser reduzido, sendo que estes podem ser fixos ou variáveis. Além disso, outros trabalhos podem explorar a aplicação de outras técnicas de seleção de características e outros métodos de detecção em *stream*. Por fim, existe espaço para pesquisas envolvendo outros cenários de segurança, com outras bases de dados, outros tipos de ataques e aplicações de rede.

## 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

Abreu, D., Carvalho, I., Abelém, A. J., Menasché, D., Leão, R. M., and Silva, E. (2020). Seleção de Características por Clusterização para Melhorar a Detecção de Ataques de Rede. In *Proceedings of the 38th Brazilian Symposium on Computer Networks and Distributed Systems*, pages 295–308, Porto Alegre, RS, Brasil. SBC.

Al Nuaimi, N., Masud, M., Serhani, M., and Zaki, N. (2019). Streaming feature selection algorithms for big data: A survey. *Applied Computing and Informatics*, ahead-of-print.

Azhagusundari, B., Thanamani, A. S., et al. (2013). Feature selection based on information gain. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 2(2):18–21.

Bifet, A. and Gavalda, R. (2009). Adaptive learning from evolving data streams. In *International Symposium on Intelligent Data Analysis*, pages 249–260. Springer.

Bifet, A. and Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. *Proceedings of the 7th SIAM International Conference on Data Mining*, 7.

Bifet, A., Holmes, G., Pfahringer, B., Kranen, P., Kremer, H., Jansen, T., and Seidl, T. (2010). Moa: Massive online analysis, a framework for stream classification and clustering. In *Proceedings of the First Workshop on Applications of Pattern Analysis*, pages 44–50. PMLR.

Casas, P., Mulinka, P., and Vanerio, J. (2019). Should i (re)learn or should i go(on)? stream machine learning for adaptive defense against network attacks. In *Proceedings of the 6th ACM Workshop on Moving Target Defense*, MTD'19, page 79–88, New York, NY, USA. Association for Computing Machinery.

Dawid, A. P. (1992). Prequential data analysis. *Lecture Notes-Monograph Series*, pages 113–126.

Dhanabal, L. and Shantharajah, S. (2015). A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(6):446–452.

Fontugne, R., Borgnat, P., Abry, P., and Fukuda, K. (2010). Mawilab : Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. *Proceedings of the 6th International Conference on Emerging Networking Experiments and Technologies, Co-NEXT'10*, page 8.

Gama, J., Sebastiao, R., and Rodrigues, P. P. (2013). On evaluating stream learning algorithms. *Machine learning*, 90(3):317–346.

Garcia, S., Grill, M., Stiborek, J., and Zunino, A. (2014). An empirical comparison of botnet detection methods. *computers & security*, 45:100–123.

Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfharinger, B., Holmes, G., and Abdessalem, T. (2017). Adaptive random forests for evolving data stream classification. *Machine Learning*, 106(9):1469–1495.

Hulten, G., Spencer, L., and Domingos, P. (2001). Mining time-changing data streams. In *ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 97–106. ACM Press.

Khraisat, A., Gondal, I., Vamplew, P., and Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(1):1–22.

Moraes, M. B. d. et al. (2019). Avaliação de desempenho de algoritmos de seleção de atributos aplicados à classificação de fluxos de dados com mudanças de conceito. *Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Tecnologia, Limeira, SP.*

Moussa, A. A., Nogueira, M., and Guedes, A. L. (2019). Seleção online de features em streaming baseada em alpha-investing para dados de ataques ddos. In *Anais do XXIV Workshop de Gerência e Operação de Redes e Serviços*, pages 43–56. SBC.

Moustafa, N. and Slay, J. (2015). Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, pages 1–6.

Mulinka, P. and Casas, P. (2018). Stream-based machine learning for network security and anomaly detection. In *Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, Big-DAMA '18, page 1–7, New York, NY, USA. Association for Computing Machinery.

Mulinka, P., Wassermann, S., Marín, G., and Casas, P. (2018). Remember the Good, Forget the Bad, do it Fast - Continuous Learning over Streaming Data. In *Continual Learning Workshop at NeurIPS 2018*, Montréal, Canada.

Rish, I. et al. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46.

Schuartz, F., Munaretto, A., and Fonseca, M. (2019). Uma comparação entre os sistemas de detecção de ameaças distribuídas de rede baseado no processamento de dados em fluxo e em lotes. In *Anais do XXIV Workshop de Gerência e Operação de Redes e Serviços*, pages 29–42, Porto Alegre, RS, Brasil. SBC.

Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSp*, pages 108–116.

Wang, J., Zhao, P., Hoi, S. C., and Jin, R. (2013). Online feature selection and its applications. *IEEE Transactions on Knowledge and Data Engineering*, 26(3):698–710.

Zhang, D., Wang, Q.-G., Feng, G., Shi, Y., and Vasilakos, A. V. (2021). A survey on attack detection, estimation and control of industrial cyber–physical systems. *ISA Transactions*.

Zhou, J., Foster, D., Stine, R., and Ungar, L. (2005). Streaming feature selection using alpha-investing. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 384–393.