Comparative Study of Different Feature Sets for Intrusion Detection in Computer Networks

André Oliveira de Alcântara, Rodrigo S. Miani, Elaine Ribeiro de Faria

[1]Faculdade de Computação – Universidade Federal de Uberlândia
Uberlândia – MG – Brasil

{andre.alcantara,miani,elaine}@ufu.br

Abstract. *Intrusion detection in computer networks is a fundamental task to ensure information security in digital systems. However, the effectiveness of intrusion detection systems depends on the data quality and the selection of attributes used to train classification models. This study aims to examine how different attribute sets impact intrusion detection in data streams. We conducted an experimental analysis using the CICIDS2017 database, comparing various attribute sets from existing literature along with a set we propose. We also considered different delays in obtaining true labels. Our findings suggest that while delayed labels affect classifier performance, the choice of attribute set used in training also plays a significant role.*

Resumo. *A detecção de intrusão em redes de computadores é uma tarefa fundamental para garantir a segurança da informação em sistemas digitais. Porém, a eficácia dos sistemas de detecção de intrusão depende da qualidade dos dados e da escolha dos atributos utilizados para treinamento dos modelos de classificação. Este trabalho tem como objetivo identificar o impacto das diferentes seleções de grupos de atributos na tarefa de detecção de intrusão para cenários de fluxos contínuos de dados. Um estudo experimental foi realizado usando a base de dados CICIDS2017, diferentes conjuntos de atributos encontrados na literatura, bem como um conjunto proposto neste trabalho, e diferente atrasos na entrega dos rótulos. Os resultados indicam que o desempenho do classificador é afetado pelo atraso na entrega dos rótulos, mas também pelo conjunto de atributos usados no seu treinamento.*

1. Introdução

A era digital avançou significativamente nos últimos tempos, trazendo consigo preocupações sobre a segurança e integridade de dados e sistemas digitais. De acordo com o relatório *Cyber Security Report* 2023 da *Check Point Research*, no terceiro trimestre de 2022 houve um aumento de 28% nos ataques a organizações em comparação ao mesmo período de 2021 [Check Point Research 2023]. Este cenário evidencia a necessidade de mecanismos que permitam uma resposta eficaz a essas formas de ataque.

A fragilidade na segurança das redes exige métodos eficazes para detectar possíveis ameaças à segurança dos dados e à integridade dos sistemas. Entre os sistemas mais comuns estão os Sistemas de Detecção de Intrusão (*Intrusion Detection System*, IDS), que monitoram o tráfego de rede e identificam comportamentos suspeitos e potenciais ameaças [Amudha et al. 2013]. Os IDS podem ser baseados em assinaturas, que

detectam ataques conhecidos, ou em anomalias, que identificam comportamentos fora do padrão [Bhuyan et al. 2017].

Os trabalhos recentes para detecção de intrusão [Molina et al. 2020], [Olímpio et al. 2023], [Prasath et al. 2022], [Ribeiro et al. 2020], [Shao et al. 2021] tem tratado o problema como um fluxo contínuo de dados (do inglês *data streams* - DS). Os fluxos contínuos se constituem de sequências de dados gerados continuamente, geralmente em grande velocidade [Bifet et al. 2010]. Em geral, estes trabalhos usam algoritmos de classificação para DS, os quais adaptam o modelo de decisão a fim de identificar mudanças na distribuição dos dados ou o surgimento de novos tipos de ataque [Olímpio et al. 2023].

Apesar dos recentes avanços, há uma lacuna significativa na pesquisa sobre o impacto das etapas de pré-processamento e seleção de atributos no desempenho dos algoritmos de detecção de intrusão. Essa avaliação é importante visto que a eficácia dos IDS depende da qualidade dos dados e da seleção adequada de atributos para o treinamento e teste dos modelos de classificação [Fayyad et al. 1996, Molina et al. 2020, Nwagu et al. 2017].

Em sistemas de detecção de intrusão, os atributos usados para classificação são frequentemente derivados da agregação de dados de tráfego de rede. Uma técnica de agregação comum na literatura é o fluxo de rede. Um fluxo consiste em uma sequência de pacotes transmitindo dados entre dois *hosts*, compartilhando propriedades idênticas: endereços IP de origem e destino, protocolo (TCP ou UDP), e portas de origem e destino. A partir de um fluxo, podem-se extrair atributos como quantidade de pacotes, volume de bytes transferidos e tamanho médio dos pacotes.

Atualmente, diferentes trabalhos da literatura tem usado diferentes conjuntos de atributos para descrever os fluxos da rede. Os trabalho de [Olímpio et al. 2023] removeu os atributos relacionados a IP e também aqueles cuja correlação de *Pearson* com a classe era igual a 0, resultando em 11 atributos. Já o trabalho de [Prasath et al. 2022] foram removidos dez atributos considerados constantes e atributos ligados a IP e porta, totalizando 66 atributos. Já o trabalho de [Wang et al. 2014], todos os atributos não numéricos foram removidos, objetivando a aplicação de algoritmos baseados em árvores. No entanto, trabalhos que façam análise experimental sobre quais conjuntos de atributos fornecem um melhor desempenho preditivo dos classificadores ainda são escassos.

Este trabalho propõe uma análise experimental de diferentes seleções de atributos, avaliando seu impacto na classificação de tráfego de rede para detecção de intrusão. Os conjuntos de atributos analisados foram selecionados com base em trabalhos da literatura. Nos experimentos, utilizaram-se os algoritmos de classificação de fluxos contínuos de dados VFDT (*Very Fast Decision Tree*) e *Naive Bayes*, considerando diferentes atrasos na entrega dos exemplos rotulados para a atualização do modelo.

Este trabalho está estruturado da seguinte forma: a Seção 2 revisa a literatura relacionada ao tema; a Seção 3 detalha o método proposto; a Seção 4 descreve os experimentos realizados e discute os resultados obtidos; e a Seção 5 apresenta as conclusões e sugere direções para pesquisas futuras.

2. Trabalhos relacionados

Diversos trabalhos recentes têm tratado da tarefa de detecção de intrusão. No contexto de IDS, é comum a utilização de bases de dados públicas para garantir a reprodutibilidade e a comparação de resultados. A base CICIDS2017 [Sharafaldin et al. 2018], amplamente utilizada, como por exemplo, [Schuartz et al. 2019], [Olímpio et al. 2023], [Prasath et al. 2022], contém registros de ataques como *HeartBleed*, DoS, DDoS, entre outros. Outra base popular é a CTU-13 [Garcia et al. 2014], utilizada por [Ribeiro et al. 2020], [da Costa et al. 2018], [Torres et al. 2019], que contém cenários de ataques de *Botnet*. A base *KDD CUP 1999* [Stolfo et al. 1999], utilizada por [Yuan et al. 2018], [Wang et al. 2014], [Breve and Zhao 2013], é criticada por estar desatualizada, mas ainda é referência em estudos históricos.

O pré-processamento é um passo importante que permite a limpeza e seleção dos atributos presentes no conjunto de dados utilizado, potencialmente aumentando o desempenho preditivo do classificador e reduzindo os recursos computacionais consumidos no processo. Nesse sentido, técnicas de redução de dimensionalidade são comumente aplicadas. Um exemplo, é o trabalho [Schuartz et al. 2019], que selecionou os dez atributos mais relevantes da base de dados em estudo.

Em [Sousa and Silva 2022], além da redução em relação aos atributos considerados mais relevantes pelos autores, também houve a remoção de todos os atributos relacionados a endereço IP. Da mesma forma, em [da Costa et al. 2018], os autores aplicaram a remoção dos atributos relacionados a IP. Em [Ribeiro et al. 2020], os autores também realizaram a redução de dimensionalidade removendo os atributos relacionados a endereços IP e ainda criaram um atributo artificial a partir da média de bytes contidos nos pacotes de rede. Já em [Olímpio et al. 2023], os autores aplicaram o cálculo da correlação de *Pearson* para encontrar os atributos mais relacionados com a classe e remove aqueles relacionados a endereço IP. Já no trabalho [Wang et al. 2014] foi aplicada a redução de dimensionalidade removendo os atributos não numéricos da base de dados e redimensiona a base para que as instâncias de ataques representem 1% do total de registros.

A anonimização também é outra técnica empregada nesta etapa do processo de KDD. Em [Torres et al. 2019], os autores optaram pela anonimização dos atributos relacionados a endereços de IP de origem e destino e também a portas de origem e destino, buscando evitar o enviesamento do algoritmo em relação a estes atributos.

Apesar da diversidade de conjuntos de atributos usados no treinamento de modelos para detecção de intrusão, nenhum dos trabalhos apresentados nesta seção investigou o impacto do uso de diferentes conjuntos de atributos no desempenho dos modelos de classificação.

A escolha do algoritmo de classificação utilizado no processo de mineração é essencial para os trabalhos na área de detecção de intrusão, com um enfoque recente em mineração de fluxos contínuos de dados. Classificadores baseados em árvore, como *Hoeffding Adaptative Tree (HAT)*, *Random Forest*, e *Very Fast Decision Tree (VFDT)*, foram utilizados por [Schuartz et al. 2019], [Ribeiro et al. 2020], [da Costa et al. 2018], [Prasath et al. 2022], [Shao et al. 2021], [Torres et al. 2019]. Algoritmos de comitê, como *Oza Bag, Ozabag Adwin, Ozaboost*, foram explorados por [Sousa and Silva 2022], [Ribeiro et al. 2020], [Olímpio et al. 2023]. Além disso,

Naive Bayes [Schuartz et al. 2019], [Sousa and Silva 2022], [Shao et al. 2021], KNN (K-vizinhos mais próximos) [Wang et al. 2014], [Sousa and Silva 2022], [Yuan et al. 2018], e redes neurais [Hernández-Pereira et al. 2009], [Prasath et al. 2022] também são utilizados.

Outro ponto a ser destacado é que poucos destes trabalhos investigaram o impacto no desempenho preditivo dos modelos quando há um atraso na entrega de instâncias rotuladas para a atualização dos mesmos. Além disso, nenhum destes trabalhos avaliou diferentes conjuntos de atributos em cenários mais realísticos para detecção de intrusão, que inclui a entrega atrasada de instâncias rotuladas. A avaliação adequada dos modelos de classificação é crucial para garantir a eficácia dos IDSs. O método *Prequential*, com variações como *Prequential delayed*, tem sido adotado em trabalhos, tais como, [Olímpio et al. 2023], [Ribeiro et al. 2020], [Shao et al. 2021].

3. Método para avaliar o desempenho de diferentes conjuntos de atributos na tarefa de detecção de intrusão

O objetivo desta seção é apresentar o método utilizado para comparar o uso de diferentes conjuntos de atributos e seu impacto preditivo na classificação de detecção de intrusão em fluxos contínuos de dados.

3.1. Visão Geral do Método

A Figura 1 apresenta uma visão geral do método experimental proposto. Inicialmente, a base de dados passa pelo estágio de pré-processamento, no qual um conjunto de atributos é extraídos. Neste trabalho, diferentes conjuntos de atributos foram testados. Após esta etapa, o conjunto de dados é dividido entre um conjunto de treinamento inicial, totalmente rotulado, referenciado na figura como Dados Treino, e um fluxo contínuo, referenciado como Dados Teste, permitindo uma abordagem de aprendizagem online. Utilizando o conjunto de treinamento inicial, o algoritmo é treinado para gerar o modelo inicial, que é então utilizado como ponto de partida classificar as instâncias do fluxo contínuo de dados.

As instâncias do fluxo contínuo de dados são submetidas ao modelo gerado, passando pelo processo de classificação. Em seguida, o rótulo verdadeiro da instância é obtido, considerando as situações com ou sem atraso na obtenção de tais rótulos. A partir da comparação entre o rótulo verdadeiro e a previsão do modelo, medidas são usadas para avaliar a qualidade do modelo. Sempre que o rótulo verdadeiro de uma instância é obtido, o modelo é atualizado de forma incremental, adaptando-se às possíveis mudanças no fluxo de dados.

3.2. Seleção de Atributos

Foram adotadas três abordagens para a seleção de atributos, mostradas na Tabela 1: *Baseline*, seleções de atributos da literatura, e uma seleção de atributos proposta pelos autores.

3.2.1. Seleção de atributos *Baseline*

Inicialmente, foi desenvolvido um teste de referência, chamado de *baseline*, onde foram removidas as colunas relacionadas a endereços de IP de origem e destino, e portas de

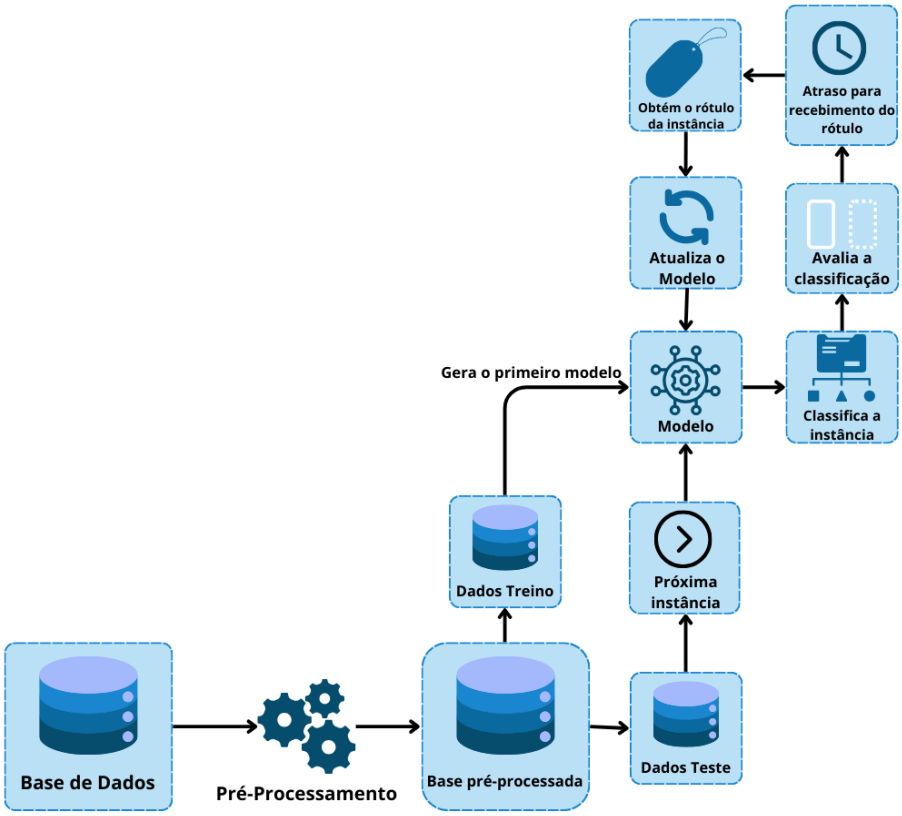


Figura 1. Esquema geral do método proposto

Tabela 1. Seleções de Atributos

Abordagem	**Número de Atributos**	**Métodos aplicados**
Baseline	77	Remoção de atributos relacionados a endereços de IP e portas.
Seleção Olímpio	11	Remoção de atributos irrelevantes e com baixa correlação com a classe
Seleção *Prasath*	66	Remoção de atributos constantes, reescala de atributos numéricos e remoção de atributos relacionados a endereços de IP e Portas
Seleção Proposta	75	Remoção de atributos similares à *baseline*, adição de dois novos atributos compostos.

origem e destino, flow bytes, ID e timestamp, totalizando 77 atributos. Instâncias com valores ausentes ou nulos foram removidas e a coluna de classe foi binarizada, com 0 para tráfego normal e 1 para qualquer classificação de ataque.

3.2.2. Seleção de atributos da Literatura

Foram aplicadas seleções de atributos baseadas em estudos anteriores:

- No estudo de Olímpio [Olímpio et al. 2023], foram removidos os atributos relacionados a endereços de IP e aqueles cuja correlação de *Pearson* com a classe alvo é igual a 0, resultando em 11 atributos.
- No estudo de *Prasath* [Prasath et al. 2022], foram removidos dez atributos constantes e atributos ligados a endereços de IP e portas. Além disso, todos os atributos numéricos contínuos foram normalizados entre 0 e 1, totalizando 66 atributos.

3.2.3. Proposta de Seleção de atributos

Foi realizada uma seleção de atributos proposta pelos autores deste estudo. Foram removidas colunas de "*Source IP*", "*Source Port*", "*Destination IP*", "*Destination Port*", "*Flow Bytes*", "*ID*", "*Timestamp*"e criados novos atributos como "TotalPackets"(soma de "TotalFwdPackets"e "TotalBackwardPackets") e "TotalLengthofPck"(soma de "TotalLengthofFwdPck"e "TotalLengthofBwdPck"). A coluna de classe foi binarizada, resultando em 75 atributos.

3.3. Algoritmos para Classificação

Para a realização dos experimentos, foram utilizados dois algoritmos de classificação: VFDT (*Very Fast Decision Tree*) e *Naive Bayes*. Esses algoritmos foram escolhidos devido à sua ampla adoção na literatura e adequação para cenários de fluxos de dados.

3.4. Método e Medidas de Avaliação

O método de avaliação utilizado foi *Prequential Delayed*, que permite a avaliação considerando um atraso na entrega do rótulo utilizado para a atualização do modelo, simulando um ambiente real de maneira mais fidedigna. Os experimentos foram realizados com duas configurações de atraso: 0 e 5000 *timestamps*.

Além disso, foi utilizado o método de avaliação por janelas, onde o fluxo contínuo de dados foi dividido em janelas de 5000 instâncias. O modelo foi inicialmente treinado com a primeira janela e, subsequentemente, submetido às janelas seguintes para classificação e avaliação do desempenho.

As medidas de avaliação utilizadas foram precisão, revocação e F1-Score. A classe de ataque foi considerada como positiva. Além disso, como as medidas de avaliação foram calculadas a partir das janelas de dados, quando uma janela não possuía instâncias da classe ataque, o seu valor de verdadeiros positivos (TP) é zero, as suas medidas de avaliação foram marcadas com o valor -1. Esta estratégia foi adotada para distinguir do cenário em que modelo não fez nenhuma predição correta para a classe positiva, e portanto também tem o valor de verdadeiros positivos (TP) igual a 0.

3.5. Ferramenta - *Massive Online Analysis Framework*

A ferramenta selecionada para a realização da experimentação foi o *Massive Online Analysis Framework* (MOA). Essa escolha decorre do fato de ser um *framework* de código aberto, amplamente utilizado, especialmente em trabalhos e estudos relacionados ao contexto de fluxo contínuo de dados [Bifet et al. 2010].

O MOA oferece uma variedade de algoritmos específicos para fluxo contínuo de dados, abrangendo diversas tarefas. Entre esses algoritmos, destaca-se a VFDT e o *Naive Bayes*, escolhidos para serem utilizados nas experimentações deste trabalho. Além disso, o MOA disponibiliza diversos métodos de avaliação, incluindo o método *Prequential Delayed* que possibilita configurar atraso na entrega dos rótulos, e avaliar o modelo usando diferentes métricas de desempenho.

4. Experimentos

Esta seção apresenta a base de dados utilizada, os experimentos realizados para o estudo, os resultados obtidos a partir deles e a discussão dos resultados.

A base de dados CICIDS2017 [Sharafaldin et al. 2018] foi a escolhida para os experimentos devido à sua amplitude em relação aos diversos tipos de ataques presentes e à sua ampla adoção na literatura recente. Ela contém 84 características e é organizada em cinco conjuntos de dados, cada um representando um dia da semana, com tipos específicos de ataques. Os experimentos foram conduzidos em dois dias da semana, quarta e sexta. A quarta tem uma grande quantidade de amostras maliciosas provenientes de ataques DoS, um tipo de ataque extremamente relevante para a área. Já a sexta é o dia com o maior número de tipos de ataques.

Os experimentos foram realizados com dois algoritmos de classificação: VFDT (*Very Fast Decision Tree*) e *Naive Bayes*. As Seções 4.1 e 4.2 tratam sobre os experimentos realizados na base de dados com cenários sem e com atraso na entrega dos rótulos, respectivamente. A Seção 4.3 apresenta as discussões em relação aos resultados obtidos.

4.1. Experimentos sem atraso na entrega dos rótulos

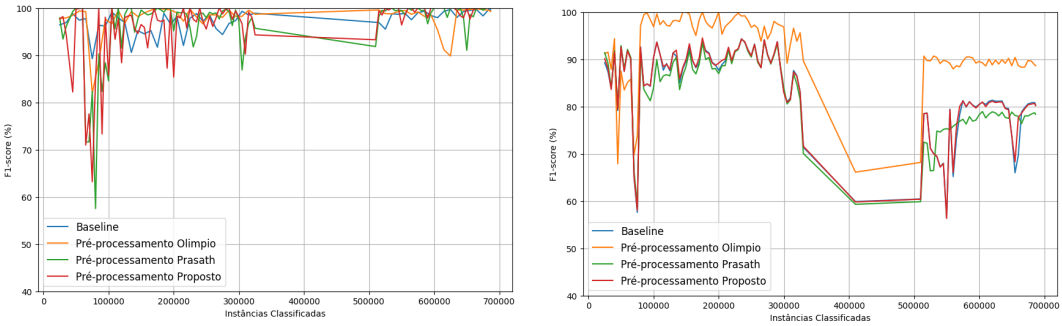
Nesta seção são apresentados os resultados dos experimentos realizados sem atraso na entrega dos rótulos, para os dias de quarta-feira e sexta-feira.

4.1.1. Base de Quarta-feira - sem atraso

As Figuras 2(a) e 2(b) apresentam os resultados de F1-Score obtidos pelo modelo VFDT e *Naive Bayes*, respectivamente, na base de quarta-feira, sem atraso na entrega dos rótulos.

Considerando o algoritmo VFDT, o modelo treinado com a seleção de atributos Olímpio apresentou o melhor desempenho geral, com alta F1-Score. O modelo treinado com *baseline* apresentou métricas um pouco abaixo das alcançadas pela seleção Olímpio. Os modelos treinados com as seleções *Prasath* e a seleção proposta tiveram grandes quedas nas métricas de desempenho.

Considerando o *Naive Bayes*, o modelo treinado com a seleção de atributos Olímpio teve um desempenho significativamente melhor em comparação com as outras seleções. Os demais modelos apresentaram maiores quedas nas métricas de desempenho.

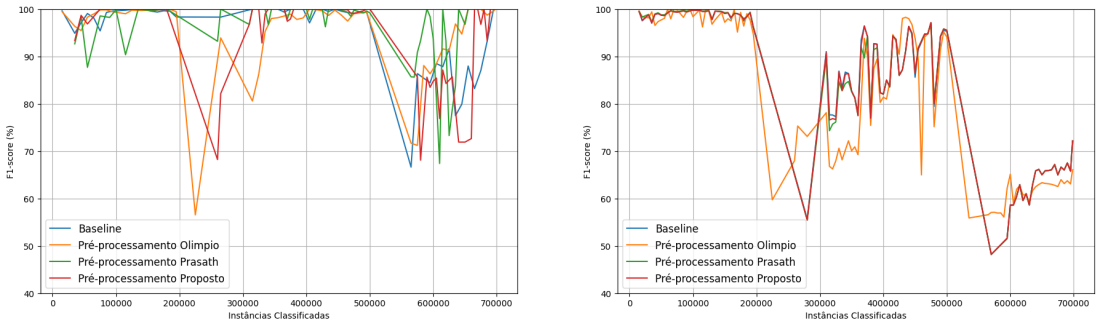


(a) VFDT (b) Naive Bayes

Figura 2. Desempenho do Modelo VFDT e *Naive Bayes* sem Atraso na entrega dos rótulos - F1-Score - Quarta-Feira

4.1.2. Base de Sexta-feira - sem atraso

As Figuras 3(a) e 3(b) apresentam os resultados de F1-Score obtidos pelos modelos VFDT e *Naive Bayes*, respectivamente, na base de sexta-feira, sem atraso na entrega dos rótulos.



(a) VFDT (b) Naive Bayes

Figura 3. Desempenho do Modelo VFDT e *Naive Bayes* sem Atraso na entrega dos rótulos - F1-Score - Sexta-Feira

O modelo VFDT treinado com a seleção de atributos *baseline* apresentou o melhor desempenho geral, com poucas quedas ao longo do fluxo. O modelo treinado com a seleção *Prasath* apresentou métricas um pouco abaixo das alcançadas pela seleção *baseline*. Os modelos treinados com as seleções Olímpio e a seleção proposta tiveram quedas mais bruscas nas métricas no decorrer do fluxo.

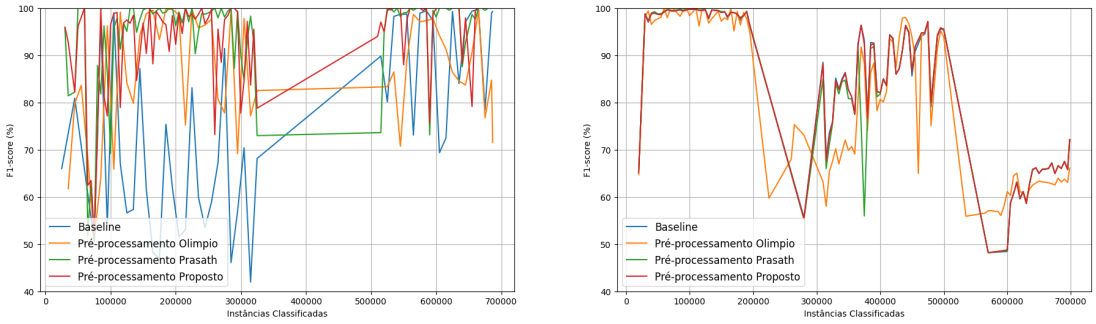
Para o modelo *Naive Bayes*, as seleções de atributos *baseline* e *Prasath* apresentaram baixo desempenho e dificuldades de previsão neste cenário. Os modelos treinados com as seleções Olímpio e a seleção proposta, obtiveram um desempenho razoável, porém mantendo métricas mais consistentes ao longo do tempo.

4.2. Experimentos com atraso na entrega dos rótulos

Nesta seção são apresentados os resultados dos experimentos realizados com atraso na entrega dos rótulos, para os dias de quarta-feira e sexta-feira.

4.2.1. Base de Quarta-feira - com atraso

As Figuras 4(a) e 4(b) apresentam os resultados de F1-Score obtidos pelo modelo VFDT e *Naive Bayes*, respectivamente, na base de quarta-feira, com atraso de 5000 *timestamps* na entrega dos rótulos.



(a) VFDT (b) Naive Bayes

Figura 4. Desempenho dos Modelos VFDT e *Naive Bayes* com Atraso na entrega dos rótulos - F1-Score - Quarta-Feira

No cenário com atraso na entrega dos rótulos, o modelo VFDT treinado com a seleção de atributos *baseline* e Olímpio tiveram o desempenho afetado, apresentando várias quedas de desempenho no decorrer do fluxo de dados. A seleção *Prasath* também apresentou algumas quedas, porém obteve melhores recuperações em relação aos outros modelos. A seleção de atributos proposta teve um resultado intermediário.

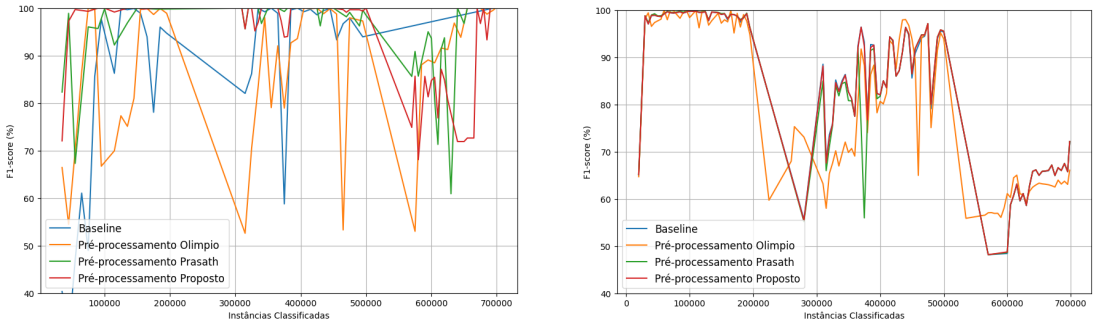
O *Naive Bayes* também foi afetado pelo atraso na entrega dos rótulos, porém apresentando quedas de desempenho menos significativas. A seleção de atributos Olímpio se destacou, mas com uma redução no desempenho em relação ao cenário sem atraso.

4.2.2. Base de Sexta-feira - com atraso

As Figuras 5(a) e 5(b) apresentam os resultados de F1-Score obtidos pelos modelos VFDT e *Naive Bayes*, respectivamente, na base de sexta-feira, com atraso de 5000 *timestamps* na entrega dos rótulos.

No cenário com atraso nas entregas dos rótulos, o modelo VFDT treinado com a seleção *baseline* e Olímpio tiveram o desempenho afetado, apresentando várias quedas de desempenho no decorrer do fluxo de dados. A seleção de atributos proposta também apresentou quedas menores e melhores recuperações em relação aos outros modelos. A seleção *Prasath* teve um resultado intermediário, apresentando maiores quedas em relação ao *baseline*.

O *Naive Bayes* também foi impactado pelo atraso na entrega dos rótulos na base de sexta-feira, mostrando uma redução no desempenho ao longo do tempo. Os modelos treinados com as seleções Olímpio e a proposta, mantiveram um desempenho razoável, mantendo métricas mais consistente ao longo do tempo. As seleções de atributos *baseline* e *Prasath* mantiveram o baixo desempenho com quedas de desempenho mais acentuadas.



(a) VFDT (b) Naive Bayes

Figura 5. Desempenho dos Modelos VFDT e *Naive Bayes* com Atraso na entrega dos rótulos - F1-Score - Sexta-Feira

4.3. Discussão dos resultados

É notável que o atraso dos rótulos aumenta consideravelmente a dificuldade de previsão dos classificadores com ambos os algoritmos. A diferença de desempenho no algoritmo VFDT com os métodos de seleção de atributos *baseline* e Olímpio fica mais acentuada ao se analisar os casos de teste no cenário com atraso de 5000 *timestamps* na entrega dos rótulos para o modelo, ambos apresentam diversas quedas nos valores das métricas de desempenho. Enquanto isso, a seleção *Prasath* atua com uma menor queda de desempenho, tendo destaque no cenário com atraso de rótulo. A seleção de atributos proposto obteve um desempenho intermediário. Para o algoritmo *Naive Bayes*, as seleções de atributos *baseline* e *Prasath* obtiveram um desempenho particularmente baixo na base de dados de sexta-feira. Todos as seleções de atributos tiveram quedas similares de desempenho quando testadas no cenário com atraso na entrega dos rótulos.

Tabela 2. Discussão dos Resultados

	Baseline	**Olímpio**	**Prasath**	**Proposta**
Sem Atraso	Bom desempenho em ambos os datasets e algoritmos.	Bom desempenho em ambos os algoritmos para quarta-feira. Desempenho intermediário em ambos os algoritmos para sexta-feira.	Baixo desempenho em ambos os datasets e algoritmos.	Baixo desempenho em ambos os datasets e algoritmos.
Com Atraso	Bom desempenho para sexta-feira com VFDT. Baixo desempenho para quarta-feira com NB.	Baixo desempenho em ambos os datasets e algoritmos.	Bom desempenho para ambos os datasets com VFDT. Baixo desempenho para ambos os datasets com NB.	Desempenho intermediário em ambos os datasets e algoritmos.

As diferenças na seleção de atributos demonstraram ter um impacto considerável no desempenho dos classificadores. A seleção de atributos Olímpio apresentou desempenho significativamente superior nos experimentos com atraso de rótulos, enquanto o *baseline* mostrou-se mais eficiente nos experimentos sem atraso na entrega dos rótulos. Esses resultados evidenciam a importância de uma escolha adequada de atributos, que pode influenciar diretamente na eficácia e eficiência do modelo em diferentes cenários.

5. Conclusão

Esse trabalho se propôs a realizar um estudo comparativo entre técnicas de seleção de atributos para o problema de detecção de intrusão em redes de computadores. A motivação

do estudo era comparar a diferença de desempenho do modelo de classificação com diferentes seleções de atributos de uma mesma base de dados e determinar, por meio de métricas de avalização, se diferentes seleções de atributos teria um desempenho superior aos demais para o cenário do projeto.

A principal contribuição oferecida por este estudo trata-se da análise experimental de diferentes seleções de atributos e a avaliação do impacto no desempenho do classificador, medido por métricas adequadas para o contexto, na tarefa de detecção de intrusão em redes de computadores.

A partir deste estudo, propõem-se trabalhos futuros que investiguem o impacto de técnicas de transformações de dados nos resultados de modelos de classificação para detecção de intrusão. Também é recomendado testar outros algoritmos de classificação além do VFDT e Naive Bayes para obter outra perspectiva na solução do problema. Por fim, sugere-se a experimentação de diferentes técnicas de seleção de atributos, bem como diversificar a base de dados, de forma a ser possível identificar e justificar o comportamento de diferentes conjuntos de atributos em cada uma delas.

Referências

Amudha, P. et al. (2013). Classification techniques for intrusion detection-an overview. *International Journal of Computer Applications*, 76(16).

Bhuyan, M. H. et al. (2017). Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys & Tutorials*, 16(1):303–336.

Bifet, A., Holmes, G., and Pfahringer, B. (2010). Leveraging bagging for evolving data streams. In *Proceedings of the 2010th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, pages 135–150. Springer.

Breve, F. and Zhao, L. (2013). Semi-supervised learning with concept drift using particle dynamics applied to network intrusion detection data. In *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*, pages 335–340. Ieee.

Check Point Research (2023). Cyber security report 2023.

da Costa, V. G. T., Zarpelão, B. B., Miani, R. S., and Junior, S. B. (2018). Online detection of botnets on network flows using stream mining. In *Anais do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 225–238. SBC.

Fayyad, U. M. et al. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37.

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

Hernández-Pereira, E., Suárez-Romero, J. A., Fontenla-Romero, O., and Alonso-Betanzos, A. (2009). Conversion methods for symbolic features: A comparison applied to an intrusion detection problem. *Expert Systems with Applications*, 36(7):10612–10617.

Molina, D. R. et al. (2020). A survey of data mining and knowledge discovery process models and methodologies. *Journal of Computing Sciences in Colleges*, 35(5):62–80.

Nwagu, I. et al. (2017). Knowledge discovery from databases: An overview. *Journal of Applied Sciences and Environmental Management*, 21(5):887–893.

Olímpio, G., Camargos, L., Miani, R. S., and Faria, E. R. (2023). Model update for intrusion detection: Analyzing the performance of delayed labeling and active learning strategies. *Computers & Security*, 134:103451.

Prasath, S., Sethi, K., Mohanty, D., Bera, P., and Samantaray, S. R. (2022). Analysis of continual learning models for intrusion detection system. *IEEE Access*, 10:121444–121464.

Ribeiro, G. H., de Faria Paiva, E. R., and Miani, R. S. (2020). A comparison of stream mining algorithms on botnet detection. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, ARES '20, New York, NY, USA. Association for Computing Machinery.

Schuartz, F. C., 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. SBC.

Shao, Z., Yuan, S., and Wang, Y. (2021). Adaptive online learning for IoT botnet detection. *Information Sciences*, 574:84–95.

Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, 1:108–116.

Sousa, W. T. M. and Silva, C. A. (2022). Análise de desempenho em algoritmos de aprendizagem de máquina na detecção de intrusão baseada em fluxo de rede usando o conjunto de dados unsw-nb15. *Revista de Sistemas e Computação-RSC*, 12(2).

Stolfo, S., Fan, W., Lee, W., Prodromidis, A., and Philip, C. (1999). KDD Cup 1999 Data. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C51C7N.

Torres, J. L. G., Catania, C. A., and Veas, E. (2019). Active learning approach to label network traffic datasets. *Journal of Information Security and Applications*, 49:102388.

Wang, W., Guyet, T., Quiniou, R., Cordier, M.-O., Masseglia, F., and Zhang, X. (2014). Autonomic intrusion detection: Adaptively detecting anomalies over unlabeled audit data streams in computer networks. *Knowledge-Based Systems*, 70:103–117.

Yuan, X., Wang, R., Zhuang, Y., Zhu, K., and Hao, J. (2018). A concept drift based ensemble incremental learning approach for intrusion detection. In *2018 IEEE International Conference on Internet of things (IThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 350–357. IEEE.