UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

MESTRADO ACADÊMICO EM ENGENHARIA DE TELEINFORMÁTICA

BRUNO RICCELLI DOS SANTOS SILVA

UMA ANÁLISE COMPARATIVA DE TÉCNICAS DE SUBAMOSTRAGEM PARA PROJETOS DE SISTEMAS DE DETECÇÃO DE INTRUSÃO EM REDES DE COMPUTADORES

FORTALEZA

2020

BRUNO RICCELLI DOS SANTOS SILVA

UMA ANÁLISE COMPARATIVA DE TÉCNICAS DE SUBAMOSTRAGEM PARA PROJETOS DE SISTEMAS DE DETECÇÃO DE INTRUSÃO EM REDES DE COMPUTADORES

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia de Teleinformática do Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas

Orientador: Prof. Dr. Paulo César Cortez

FORTALEZA

2020

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S578a Silva, Bruno Riccelli dos Santos.
Uma análise comparativa de técnicas de subamostragem para projetos de sistemas de detecção de intrusão em redes de computadores / Bruno Riccelli dos Santos Silva. – 2020.
85 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2020.
Orientação: Prof. Dr. Paulo César Cortez.

1. Sistemas de detecção de intrusão. 2. Subamostragem. 3. Aprendizagem de máquina. I. Título.
CDD 621.38

BRUNO RICCELLI DOS SANTOS SILVA

UMA ANÁLISE COMPARATIVA DE TÉCNICAS DE SUBAMOSTRAGEM PARA PROJETOS DE SISTEMAS DE DETECÇÃO DE INTRUSÃO EM REDES DE COMPUTADORES

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia de Teleinformática do Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Engenharia de Teleinformática. Área de Concentração: Sinais e Sistemas

Aprovada em: 28/02/2020

BANCA EXAMINADORA

Prof. Dr. Paulo César Cortez (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Marcial Porto Fernandez
Universidade Estadual do Ceará (UECE)

Prof. Dr. Alexandre Augusto da Penha Coelho
Universidade Federal do Ceará (UFC)

Prof. Dr. João Paulo Pordeus Gomes
Universidade Federal do Ceará (UFC)

Prof. Dr. Jarbas Nunes Aryel da Silveira
Universidade Federal do Ceará (UFC)

Prof. Dr. Giovanni Cordeiro Barroso
Universidade Federal do Ceará (UFC)

Dedico este trabalho à minha família e esposa, pessoas que fizeram de tudo para que eu chegasse onde cheguei.

AGRADECIMENTOS

Agradeço primeiramente a Deus, que iluminou meu caminho durante essa jornada, me dando saúde e força para superar as dificuldades.

À minha esposa, Luéline Elias, pelo amor, paciência, dedicação e companheirismo em todos os momentos.

À minha família, por sua capacidade de acreditar e investir em mim. Mãe, sua dedicação foi o que deu, em alguns momentos, a esperança para seguir.

Ao meu orientador, Prof. Paulo César Cortez, pelo acompanhamento e estreitamento da relação professor-aluno e exemplo de profissional bem como pelo apoio, incentivo, sugestões e comentários durante a supervisão dos meus estudos.

Ao meu co-orientador, Prof. Ricardo Jardel Nunes da Silveira, pelo apoio, incentivo, sugestões e tempo dedicado para me ajudar durante meus estudos.

Ao meu amigo Manuel, pela grande ajuda, apoio e sugestões durante os estudos.

Aos meus amigos da Universidade Federal do Ceará, 8086FC e 8086Team pela amizade e pelos momentos de descontração e estudo.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

"O sonho é que leva a gente para frente. Se a gente for seguir a razão, fica aquietado, acomodado."

(Ariano Suassuna)

RESUMO

Sistemas de Detecção de Intrusão (SDIs) figuram como um das principais soluções adotadas na área de segurança em redes para evitar intrusões de rede e garantir a segurança dos dados e serviços de forma assertiva e eficiente. Técnicas de subamostragem de classes majoritárias permitem que classificadores sejam avaliados a partir de sub-bases de dados menores de forma representativa, buscando obter melhor assertividade em tempo aceitável. Esta dissertação tem por objetivo realizar uma análise comparativa de três técnicas de subamostragem (Aleatória, Cluster centroides e NearMiss) para projetos de SDIs através de cinco classificadores em duas bases de dados recentes (CICIDS 2017 e CICIDS 2018) para fins de comparação. Para esta avaliação, entre estes classificadores, são empregados as métricas de acurácia, precision, recall e eficiência que permitem escolher o(s) classificador(es) mais adequado(s) a ser(em) utilizado(s) em projetos de SDIs baseados em aprendizagem de máquina. Além disso, adotou-se o esquema de treinamento e testes baseado em técnicas de validação cruzada, seguida do teste estatístico de Wilcoxon. Os resultados indicam que a subamostragem por Cluster centroides apresenta o melhor desempenho quando aplicados em classificadores baseados em distância, podendo-se inferir que a técnica de subamostragem influencia no processo de escolha do melhor classificador no projeto de um Sistema de Detecção de Intrusão.

Palavras-chave: Sistemas de Detecção de Intrusão. Subamostragem. Aprendizagem de Máquina.

ABSTRACT

Intrusion Detection Systems (IDS) figure as one of the leading solutions adopted in the area of network security to prevent network intrusion and ensure the security of data and services. However, this type of problem requires IDS to be assertive and efficient concerning processing time. Undersampling techniques allow classifiers to be evaluated from smaller sub-databases in a representative manner, seeking better assertiveness in less processing time. Some works in the literature present this kind of solution in the IDS project, but criteria such as the adoption of a replicable methodology, are generally not respected. Three sub-sampling methodologies were selected: random selection, by Cluster centroids and Nearmiss in two recent databases (CICIDS 2017 and CICIDS 2018) and comparison purposes between the classifiers. Thus, based on the results obtained and on the criteria adopted for the choice of classifiers, in the complete CIC2017 and CIC2018 databases, the random forest classifier obtains the best results. As for the sub-base generated, from the CIC2017 database, by the random under-sampling, the KNN classifier was considered the best for its average metrics of accuracy, efficiency, and training time. In the sub-base using the Cluster centroids under-sampling technique, generated from CIC2018, the classifier Naive Bayes gets the best results. As for the subbases generated from CIC2017 and CIC2018, using the NearMiss sub-sampling technique, the best classifiers, for their average metrics of accuracy, efficiency and training time, were KNN and Naive Bayes, respectively. Also, the results indicate that the sub-sampling by Cluster centroids presents the best performance when applied to classifiers based on distance, it follows that the technique of under-sampling influences the process of choosing the best classifier in the design of an Intrusion Detection Systems.

Keywords: Intrusion Detection Systems. Undersampling. CICIDS2018. Machine Learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

DoS	Denial of Service
IoT	Internet of Things
PCA	Principal Component Análisys
SDI	Sistemas de Detecção de Intrusão

SUMÁRIO

1 INTRODUÇÃO

No início da década de sessenta, a Internet era uma ferramenta disponível apenas em universidades, os quais utilizavam computadores caros e pesados. O crescente avanço da tecnologia proporcionou a miniaturização e consequentemente a disseminação dos dispositivos de computação, tais como celulares, *notebooks*, *tablets* e etc (MEILE *et al.*, 2019).

Em paralelo com as melhorias nos dispositivos, também foi ampliada a abrangência e velocidade de comunicação de dados na *internet*, principalmente pela evolução das tecnologias de comunicação com e sem fio, tais como ADSL (*Asymmetric Digital Subscriber Line*), Wi-Fi e 4G/5G. Essa evolução, concomitante de dispositivos e tecnologias de suporte à comunicação, fez a *internet* deixar de ser uma ferramenta restrita a um público limitado para atingir todas as pessoas, de todas as idades e classes sociais. Assim hoje, as pessoas fazem o uso da *internet* em seu cotidiano para atividades, tais como, comunicar-se com amigos e familiares por mensagens instantâneas, chamadas de voz e fazer uso de aplicações *online*, o que tem sido cada vez mais comum no contexto atual na rede mundial de computadores. Além disso, o número de serviços e comodidades fornecidos aos usuários apenas aumentam seja por aplicações *web*, aplicativos de celular, ou até mesmo dispositivos Internet of Things (IoT). Com a difusão dessas tecnologias, atividades maliciosas na rede vêm surgindo em proporção cada vez maior, colocando em risco os serviços, bem como a integridade dos dados dos usuários (SILVA NETO *et al.*, 2019).

Um ataque consiste de uma atividade maliciosa que explora uma vulnerabilidade em uma rede ou em um dispositivo na rede. Dentre os ataques em redes de computadores, o Denial of Service (DoS) é o que ocasiona uma maior perturbação da qualidade de serviço, representando uma ameaça constante a cibersegurança. Um ataque dessa natureza objetiva tornar um nó na rede (geralmente servidor *web*) parcialmente ou completamente indisponível, inundando-o com falsas requisições, ocupando assim a largura de banda e/ou consumindo seus recursos computacionais. Uma evolução deste tipo de ataque trata-se do DDoS, possuindo as mesmas características do anterior, porém sendo executado de forma distribuída, ou seja, realizado por mais de um atacante ou mesmo um atacante controlando remotamente outras vítimas, com o objetivo de tornar indisponível um servidor/serviço alvo (SALIM *et al.*, 2019).

O primeiro registro de atividades DDoS aconteceu em Julho de 1999, na universidade de Minnesota - Estados Unidos, em que a rede da vítima ficou indisponível por mais que dois dias por meio do ataque conhecido por Trin00. No ano seguinte, um ataque DDoS tornou-se conhecido mundialmente, quando um grande número de sites como o *Yahoo, EBay, Amazon e*

CNN ficaram inoperantes devido a ataques dessa natureza (OSTERWEIL *et al.*, 2019).

Há outro tipo de ataque, conhecido como Mirai, que é um tipo de *botnet* que infecta dispositivos inteligentes (IoT) na rede para realizar um ataque em grande escala em um alvo. Esse ataque ocorreu em 2016, em que milhões de equipamentos IoT infectados (zumbis) geraram inúmeras requisições à provedora de serviços DNS chamada Dyn, que não suportou o montante de tráfego e os serviços que dependem do servidor ficaram indisponíveis por aproximadamente 24 horas: PayPal, Twitter, Netflix, Spotiy, dentre outros. Os prejuízos do ataque foram estimados em 8,6 milhões de dólares, chamando a atenção para a segurança em dispositivos IoT (KOLIAS *et al.*, 2017; SALIM *et al.*, 2019).

Nesse sentido, redes que não possuem nenhum mecanismo de análise de tráfego, não podem garantir segurança aos clientes que a utilizam, pois, não têm garantia de que podem operar eficientemente. Além disso, redes que não possuem sistemas de detecção de atividades maliciosas estão sujeitos a terem suas funcionalidades comprometidas, ou mesmo invadidas por ataques produzidos por um agente malicioso na rede (KHRAISAT *et al.*, 2019).

Para lidar com esses tipos de violações de segurança, Sistemas de Detecção de Intrusão (SDI)s são responsáveis pelo monitoramento e detecção de anomalias em redes de computadores (UTIMURA; COSTA, 2018; ARAUJO *et al.*, 2017). Tais mecanismos buscam antecipar/detectar atividades maliciosas por meio de classificadores, os quais podem distinguir um tráfego de duas formas: classificação binária ou multi-classes (SILVA NETO; GOMES, 2019). No primeiro caso, o tráfego de rede pode ser considerado normal ou anômalo, já com múltiplas classes, além do tráfego normal, o sistema detecta especificamente o tipo de ataque ou anomalia em questão. Os exemplos práticos de SDIs utilizados atualmente são o Snort (ROESCH *et al.*, 1999) e Bro (PAXSON, 1999).

Técnicas de aprendizagem de máquina permitem que um computador aprenda sobre um determinado comportamento e, a partir disso, consiga inferir acerca de dados que não foram utilizados na fase de aprendizagem. SDIs baseados em aprendizagem de máquina fornecem uma metodologia baseada em aprendizagem para descobrir ataques de acordo com o comportamento treinado no sistema (MISHRA *et al.*, 2018).

Os projetistas de sistemas de detecção de intrusão enfrentam diversos desafios ao adotarem abordagens baseadas em aprendizado de máquina, entre os quais, o fato de que o desempenho do SDI depende da qualidade, tamanho e generalização de bases de dados de conhecimento utilizados para treinamento e testes, bem como do classificador escolhido para o

reconhecimento dos padrões de rede (LI *et al.*, 2019). Ao encontro disso, o desbalanceamento em bases de dados é uma condição em que a proporção entre as amostras de cada classe não é igualmente distribuída e consequentemente influencia no comportamento dos classificadores. Desta forma, é necessário que, no projeto dos SDIs, sejam avaliados uma vasta quantidade de classificadores em tempo hábil. Portanto, uma abordagem que adote apenas parte das bases no processo de avaliação mantendo a representatividade e generalização, é necessária. Assim, permite-se considerar características como a quantidade de registros nas bases de dados, tempo necessário para treino/teste dos classificadores, além do custo computacional utilizados em suas avaliações (LEE; PARK, 2019).

A literatura pesquisada apresenta trabalhos com diferentes abordagens para avaliação de classificadores no projeto de SDIs. Diversos trabalhos recentes foram publicados utilizando bases de dados obsoletas, tais como NSL KDD, DARPA e CAIDA (BHATTACHARJEE *et al.*, 2017; MEENA; CHOUDHARY, 2017; ZHANG *et al.*, 2019, 2019; Thomas; Pavithran, 2018; STIAWAN *et al.*, 2017; JIAO *et al.*, 2017). Entretanto, tais bases não possuem ataques recentes, tais como SQL Injection e Heartbleed, os quais são comuns em cenários atuais.

Já SILVA NETO e Gomes (2019) avaliaram o desempenho de algoritmos de detecção de intrusão baseados em aprendizagem de máquina na base de dados CIC2017 (SHARAFALDIN *et al.*, 2018), utilizando diferentes técnicas de amostragem. Entretanto, por utilizar toda a base de dados em sua avaliação, não foi possível avaliar de forma eficiente em termos de tempo de processamento algoritmos complexos computacionalmente, tais como *Support Vector Machine* (SVM). Parsaei *et al.* (PARSAEI *et al.*, 2016), visando avaliar algoritmos complexos computacionalmente e tendo em vista o número de descritores nas bases de dados, utilizaram-se de técnicas de seleção de características tais como a Análise dos Componentes Principais (PCA). Nesta mesma direção, Utimura e Costa (2018) utilizaram 10% da base de dados ISCXIDS2012, mantendo diferentes proporções entre classes. Um estudo recente sobre as bases NSL-KDD, CIC2017 e CIC2018 foram realizados por D'hooge *et al.* (2019), porém a abordagem por subamostragem utilizada, bem como o fluxo dos experimentos realizados carecem de mais detalhes para fins de replicação do trabalho.

Além disso, nenhum desses trabalhos utiliza a subamostragem de forma sistemática, variando técnicas de subamostragem visando avaliar a generalização das bases de dados subamostradas.

1.1 Objetivos

Objetivo Geral

O objetivo geral desta dissertação é avaliar o desempenho de diferentes técnicas de subamostragem em bases de dados recentes para o projeto de sistemas de detecção de intrusão baseados em aprendizagem de máquina.

Objetivos Específicos

Além do objetivo geral, outros objetivos específicos devem ser alcançados:

- avaliar o desempenho de SDIs baseados em aprendizagem de máquina em bases de dados subamostradas, a partir de diferentes técnicas encontradas na literatura;
- comparar os classificadores a partir de teste estatístico, visando a assertividade e tempo de reconhecimento para detecção de ataques.

1.2 Contribuições

Os resultados parciais desta dissertação foram reunidos no seguinte artigo:

- Bruno Silva, Manuel da Silva Neto, Paulo Cortez e Danielo Gonçalves. "Design of Network Intrusion Detection Systems with under-sampled datasets ". CHILECON 2019, Valparaíso, Chile, Outubro 2019.

Além disso, as seguintes contribuições podem ser destacadas:

- análise exploratória das bases de dados CIC2017 e CIC2018;
- avaliação do desempenho na base CIC2018 completa; e
- criação de bases subamostradas a partir de diferentes técnicas aplicadas nas bases CIC2017 e CIC2018.

1.3 Organização da Dissertação

Esta dissertação está organizada de forma que no Capítulo 2 está descrita a fundamentação teórica sobre os temas abordados no trabalho. Os trabalhos relacionados são discutidos no Capítulo 3. O Capítulo 4 apresenta a metodologia do trabalho, descrevendo e explicando os experimentos realizados na dissertação. Os resultados e discussões são abordados no Capítulo 5. Por fim, no Capítulo 6 são sintetizadas as conclusões, contribuições, bem como as perspectivas

de trabalhos futuros.

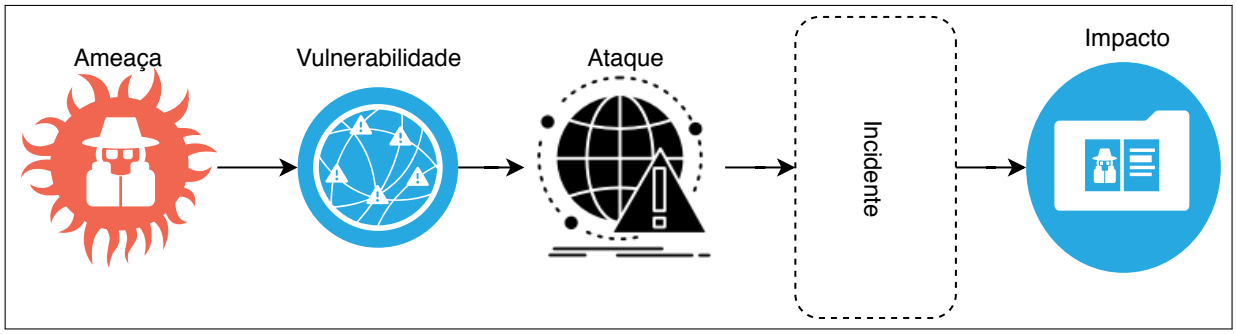
2 FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo, os conceitos para a compreensão desta dissertação são apresentados, abordando definições na área de segurança em redes tais como ataques, SDIs, aprendizagem de máquina, técnicas de re-amostragem, bem como testes estatísticos.

2.1 Segurança em redes de computadores

Em redes de computadores, uma ameaça trata-se de um potencial para violação de segurança quando há uma circunstância que pode quebrá-la, causando danos a um serviço/*host*. Exemplos de ameaças são: *malwares*, ataques de negação de serviço e envio de pacotes com falso endereço origem. Uma ameaça desta natureza explora uma vulnerabilidade no alvo para obter as informações que deseja ou mesmo tornar o serviço indisponível, ou seja, provoca a violação da segurança no alvo atacado. Um ataque pode ocasionar, por exemplo, a destruição de dados, perda da integridade, dentre outros incidentes. Um incidente, no que lhe concerne, pode causar prejuízos financeiros, na reputação do alvo, além de acarretar indisponibilidade do serviço fornecido (KUROSE; ROSS, 2010). A Figura 1 mostra uma síntese desses conceitos, na qual uma ameaça utiliza-se de uma vulnerabilidade para realizar um ataque que, produz um incidente que causa um impacto substancial no alvo, podendo ser de origens financeiras, de informações ou na disponibilidade dos serviços.

Figura 1 – Fluxo explicativo de uma ameaça



Fonte: elaborado pelo autor (2020).

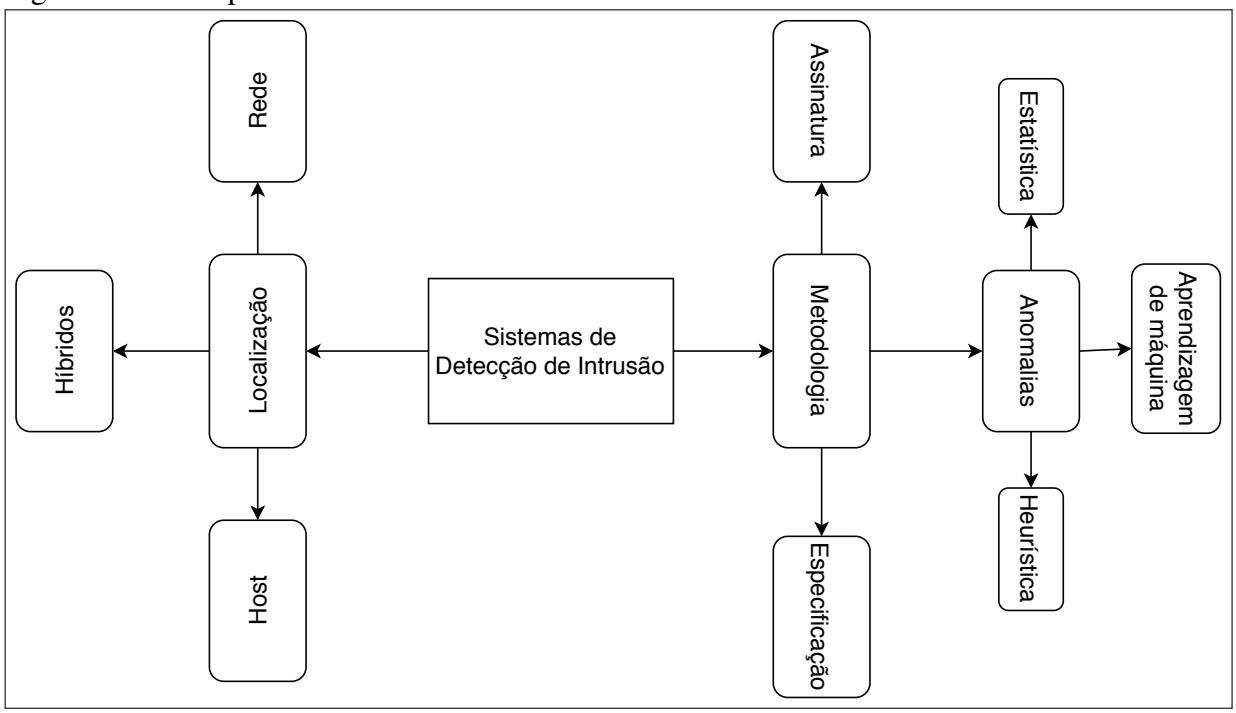
Uma intrusão trata-se de um conjunto de ações objetivando o comprometimento dos pilares da segurança da informação (KUROSE; ROSS, 2010): integridade, confidencialidade e disponibilidade. Na integridade, deve-se garantir que o conteúdo da comunicação não seja alterado por acidente ou má intenção durante a transmissão. Já a confidencialidade trata-se do conceito em que apenas o remetente e destinatário pretendidos devem poder entender o conteúdo

da mensagem transmitida. Por fim, a disponibilidade está relacionada à garantia do acesso aos dados ou serviços.

2.2 Sistemas de Detecção de Intrusão - SDIs

Para tratar esse problema são utilizados Sistemas de Detecção de Intrusão. Tratam-se de *softwares* responsáveis por detectar anomalias na rede que os *firewalls* tradicionais não conseguem lidar, como por exemplo, acessos não autorizados e tráfegos mal-intencionados. Os SDIs monitoram, em tempo real o tráfego, buscando anomalias e, em caso positivo, alertam aos administradores da rede para que estes tomem as medidas corretivas. Estas medidas podem ser, por exemplo, bloquear as portas, ou negar serviço a um IP específico que esteja enviando requisições maliciosas, ou, inundando serviços que são geralmente utilizados para ataques (KHRAISAT *et al.*, 2019).

Figura 2 – Principais subdivisões de SDIs



Fonte: elaborado pelo autor (2020).

A Figura 2 ilustra as principais SDIs, os quais subdividem-se em localização e metodologia. A localização de um SDI na rede pode impactar na detecção da ameaça, afetando a acurácia do sistema. Nesse sentido, SDIs classificam-se em três tipos: *Host-Based*, *Network-Based* e híbridos. Na primeira abordagem, um SDI pode ser instalado no dispositivo alvo para a aquisição/avaliação do tráfego. Já no caso dos SDIs *Network-Based*, a captura do tráfego é

realizada na rede, sendo possível monitorar todos os ativos na mesma. Por fim, um sistema híbrido trata-se de uma combinação dos dois tipos anteriores (HINDY *et al.*, 2018).

Já quanto à metodologia, Hindy *et al.* (2018) definem SDIs em três tipos: baseados em assinatura, anomalias e especificação. Um sistema de detecção baseado em assinaturas representa um ataque conhecido através de um padrão ou assinatura para que os ataques descritos e suas variações sejam identificados. Assim, esse sistema depende de listas atualizadas com padrões de ataques e nesse sentido, deve ser impossível detectar uma ameaça desconhecida ou atualizada. Sua principal vantagem é a alta taxa de detecção para ataques conhecidos. Entretanto, esse SDI é incapaz de detectar ameaças desconhecidas e polimórficas. No caso de sistemas baseados em anomalias, é importante que haja uma classificação da rede como normal ou anômala, além de estabelecer um comportamento benigno da rede. Tais sistemas possuem três subcategorias fundamentados no método de treinamento que são baseados em: estatística, heurística e aprendizagem de máquina. Sistemas baseados em estatística incluem análises uni-variadas, multivariadas e de séries temporais das características de rede para o processo de detecção do tráfego a ser analisado. Já os sistemas baseados em heurística utilizam máquinas de estado finitas e regras para a identificação do tráfego. Por outro lado, sistemas baseados em aprendizado de máquina compreendem técnicas tais como redes neurais, clusterização, algoritmos genéticos , etc. Por fim, um sistema de detecção baseado em especificação é responsável por monitorar os processos e caso detecte qualquer comportamento anormal, emite um alerta e deve ser mantido e atualizado sempre que houver alguma alteração. Este método, por requerer treinamento do sistema, a acurácia contra ameaças desconhecidas e polimórficas é maior, se comparado com o SDI anterior. Entretanto, a taxa de falsos positivos é alta (HINDY *et al.*, 2018).

A partir das definições já realizadas anteriormente, os projetistas de SDIs baseiam-se em alguns conceitos para realizar suas estratégias. Pode-se destacar a definição de objeto de tráfego, o qual significa um conjunto de pacotes em uma determinada janela de tempo. A partir de um objeto de tráfego, pode-se calcular métricas de avaliação da rede. Vale ressaltar que, para a obtenção de objetos de tráfego, faz-se necessário o uso de *sniffer*, um analisador de rede que captura o tráfego de entrada e saída. Desta forma, os pacotes são capturados e se calculam os parâmetros do objeto de tráfego.

2.3 Etapas de aprendizagem de máquina no projeto de SDIs

SDIs baseados em anomalia, especificamente aprendizagem de máquina, requerem que dados sejam previamente treinados para a generalização do modelo utilizado. Para este fim, métodos de amostragem tais como a divisão entre parcelas de treino e teste, bem como de validação cruzada, buscam otimizar este processo em termos de métricas de assertividade e tempo.

2.3.1 Holdout

Este método consiste em dividir os dados em dois subconjuntos diferentes: treinamento e teste. Os tamanhos de cada uma das parcelas podem ser definidos de forma igual ou diferente. Após essa divisão, a estimação do modelo é realizada nos dados de treinamento e, em seguida, os dados de teste são aplicados e as métricas de avaliação dos classificadores são computadas.

2.3.2 Validação cruzada

Trata-se de uma técnica para avaliar a capacidade de generalização de um modelo com base no conjunto de dados de entrada. Diferentemente do método anterior, nesta técnica os dados são particionados em k subconjuntos mutualmente exclusivos e iguais em tamanho. Em seguida, um dos subconjuntos é utilizado para teste e os k-1 restantes são utilizados para o treinamento do modelo. Este processo é realizado N vezes alternando de forma circular o subconjunto de teste.

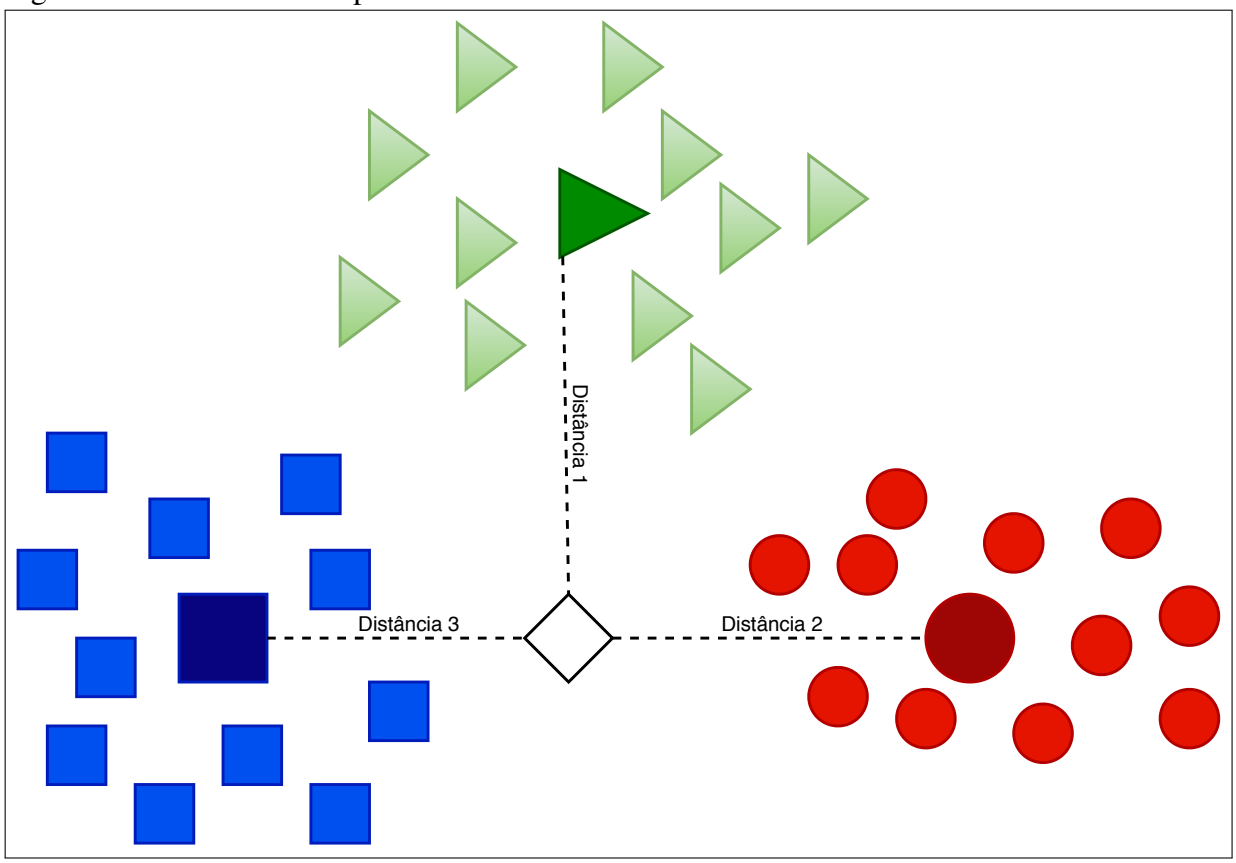
2.4 Algoritmos de aprendizagem no projeto de SDIs

Técnicas de aprendizado de máquina tem sido adotadas no projeto de SDIs ao longo dos anos e diversos algoritmos de classificação foram utilizados na detecção de intrusões e reconhecimento de padrões no tráfego da rede. Dentre os quais se destacam: Centroide mais próximo, Naive Bayes, Florestas aleatórias, K-vizinhos mais próximos e máquinas de vetores de suporte (LI *et al.*, 2019).

2.4.1 Algoritmo do centroide mais próximo (NC)

É um algoritmo que é utilizado como uma das principais referências nessa dissertação por sua simplicidade de implementação/uso, tendo em vista que o mesmo representa cada classe pelo centroide das amostras. Em sua formulação proposta por Tibshirani *et al.* (2002), tem-se que o centroide de cada classe é calculado pela média de todas as características e a predição é a classe que possui a menor distância para a amostra a ser identificada. Na Figura 3, existem as classes 1 (Azul), 2 (Verde) e 3 (Vermelho), bem como a amostra a ser classificada. As distâncias 1, 2 e 3 entre a amostra a ser classificada e o centroide de cada classe são calculadas. Assim, a predição é baseada pela classe que possui a menor distância entre a amostra a ser identificada e o centroide de cada classe. Neste caso, a amostra em análise deve pertencer à classe em azul, pois, a distância 3 corresponde a menor dentre as demais.

Figura 3 – Centroide mais próximo.



Fonte: elaborado pelo autor (2020).

A principal vantagem deste classificador é que, dado um grande número de amostras, a predição pode ser computacionalmente mais rápido que os demais. Já quanto a desvantagem tem-se que o algoritmo não lida bem com dados contendo muitos pontos fora da curva de

distribuição (LANTZ, 2015; TOMAR, 2013).

2.4.2 Naive Bayes (NB)

Classificadores baseados em métodos Bayesianos utilizam os dados de treinamento para calcular as probabilidades de cada classe extraída de informações provindas dos preditores. Segundo (LANTZ, 2015), quando o classificador é aplicado para os testes, as probabilidades observadas no treinamento são utilizadas para a predição, sendo mais apropriado o seu uso quando o número de características é alto. Além disso, os classificadores Bayesianos são utilizados para: classificação em textos tais como filtragem de e-mails, detecção de anomalias ou intrusões em redes de computadores e diagnóstico de condições médicas dado um conjunto de sintomas observados.

Ainda segundo (LANTZ, 2015), esses algoritmos são melhores aplicados a problemas em que a informação de numerosos atributos precise ser considerada simultaneamente, objetivando estimar a probabilidade total de uma classe.

Esse algoritmo, como sugere seu nome, tem como base a probabilidade condicional definida pelo teorema Bayesiano. Assim, para o conjunto de classes, a função de predição é dada pelo argumento máximo entre produtório das probabilidades condicionais entre os preditores de cada classe e o conjunto de classes a ser avaliado. O principal problema deste classificador é que o mesmo assume que todos os atributos são independentes entre si, enquanto outras aplicações tais como detecção de intrusões e medicina, os preditores são correlacionados. Entretanto, para problemas em que se pode assumir que os atributos são independentes, esse algoritmo apresenta bom desempenho em termos de falsos positivos e negativos (TOMAR, 2013).

A partir disso, o algoritmo de classificação Naive Bayes pode ser definido pela seguinte fórmula geral:

$$\hat{y} = \underset{k \in 1,\ldots,K}{\operatorname{argmax}} [p(C_k) \prod_{i=1}^{n} p(x_i|C_k)], \tag{2.1}$$

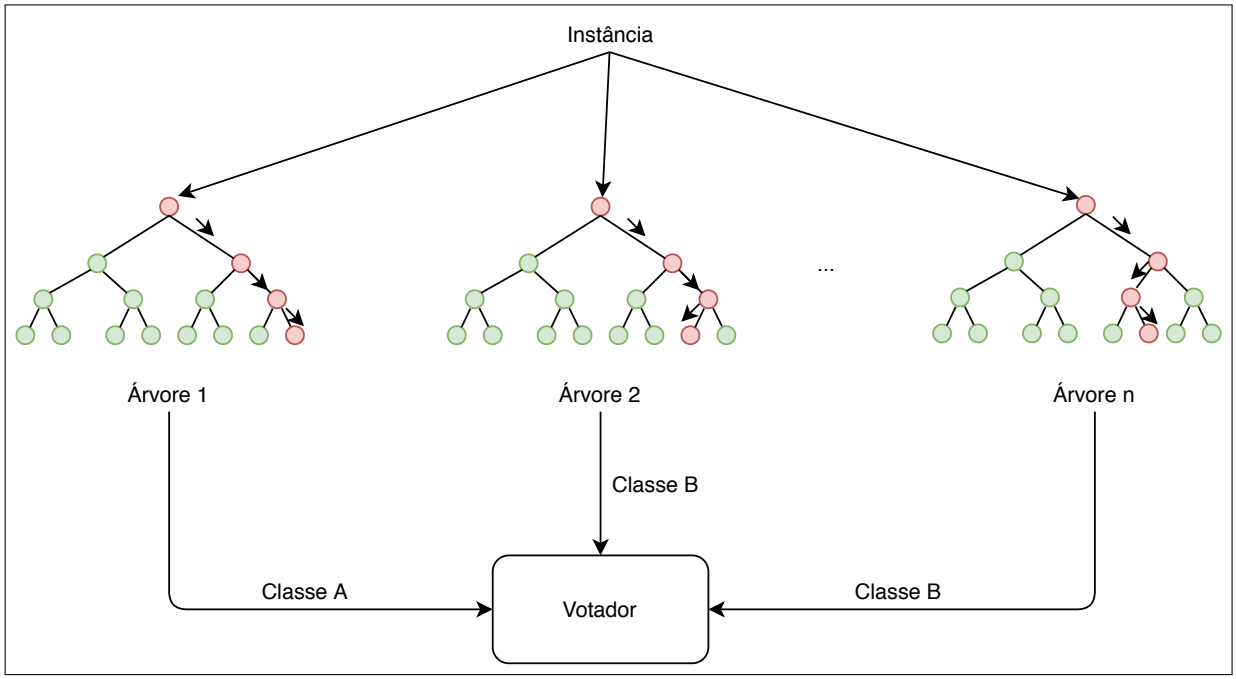
nos quais $x = (x_1, \ldots, x_n)$ representa os n preditores, C_k é o conjunto de classes e $argmax[.]$ é o maior argumento do conjunto $[.]$.

2.4.3 Florestas Aleatórias (RF)

Trata-se de um método de classificação que funciona criando uma grande quantidade de árvores de decisão para o treinamento e a saída do modelo é a moda das classes (PEDREGOSA

et al., 2011). Na Figura 3 estão mostradas as árvores 1, 2 e 3. Neste caso, o resultado produzido pelo votador é dado pela moda predita por essas árvores.

Figura 4 – Florestas Aleatórias.



Fonte: elaborado pelo autor (2020).

Segundo Lantz (2015), este classificador consegue lidar bem com dados inválidos e categóricos além de ser recomendado para bases de dados que contém muitas características. Enquanto a principal desvantagem é que se necessita de uma grande quantidade de recursos computacionais para o algoritmo, uma vez que o mesmo cria muitas instâncias de árvores para realizar a votação para a predição.

2.4.4 K-Vizinhos mais próximos (KNN)

Classificadores baseados nos vizinhos mais próximos são definidos por sua característica de classificar amostras atribuindo-as a classe similar aquela treinada no modelo. As aplicações são diversas, dentre as quais destacam-se: reconhecimento facial, predizer se uma pessoa deve gostar de uma recomendação de filmes ou músicas, bem como na identificação de padrões de dados genéticos. Como o seu nome sugere, após a escolha de um valor de k ímpares vizinhos e a fase de treinamento, para cada amostra a ser classificada, o KNN identifica as k amostras na fase de treinamento que são mais próximas. Assim, a classe predita é a que possui maioria entre os k vizinhos. Trata-se de um algorítimo simples e rápido para treinamento, entretanto requer a definição de um valor de k vizinhos, é lento para testes e características

categóricas e dados inválidos necessitam de processamento adicional antes de avaliar o algoritmo (LANTZ, 2015).

2.4.5 Máquinas de Vetores de suporte (SVM)

Trata-se de um algoritmo proposto por Vapnik (1998), o qual cria um ou múltiplos hiperplanos que buscam separar amostras de diferentes classes. Suas aplicações incluem: classificação de dados na área de bioinformática, categorização de texto, tais como identificação de linguagem utilizada em um documento e detecção de eventos tais como terremotos e outros desastres naturais.

Todavia, este algoritmo é lento para treino caso a base de dados de entrada possua um número grande de amostras ou características. Além disso, para mapear um conjunto de dados para um espaço multidimensional de características é necessário utilizar uma função núcleo. Segundo Lantz (2015), não há nenhuma regra para a escolha de um núcleo em processos de aprendizagem. Assim, a escolha do mesmo é arbitrária e a performance pode variar levemente. É um classificador robusto, lida bem com pontos não lineares e não é tão suscetível à sobreajustes. Já a principal desvantagem é que por ser complexo computacionalmente, o tempo de treinamento é maior que o normal (TOMAR, 2013).

2.4.6 Subamostragem no projeto de SDIs

Muitos algoritmos de classificação tem seu desempenho alterado ao lidar com bases de dados desbalanceadas, ou seja, quando o número de registros em determinada classe difere largamente das demais e altera demasiadamente sua representação. Uma forma de lidar com este problema é com uso de técnicas chamadas re-amostragem para diminuir os efeitos do não balanceamento e criar fronteiras de decisão mais adequadas (LEMAîTRE *et al.*, 2017).

Uma forma de avaliar o desbalanceamento em uma base de dados é pela razão de desbalanceamento (I_D), no qual consiste na razão do número de amostras da classe majoritária e os das classes minoritárias.

Quando se dispõe de dados suficientes, uma técnica simples e de fácil aplicação é a subamostragem de classes majoritárias, na qual os registros da classe dominante são removidos até que se tenha uma base balanceada ou pelo menos se diminua os efeitos do desbalanceamento (LEMAîTRE *et al.*, 2017).

Taxa de desbalanceamento (I_D)

A taxa de desbalanceamento trata-se da razão entre a classe majoritária e a minoritária. Tal medida serve para indicar o quão uma base de dados é desbalanceada.

2.5 Métodos de re-amostragem

Uma abordagem simples para o balanceamento de classes consiste na re-amostragem dos dados originais, até que as classes se tornem igualmente representadas. Visando reduzir os efeitos do desbalanceamento entre as classes, técnicas de re-amostragem que são utilizadas na literatura se dividem em duas grandes categorias: subamostragem e sobre-amostragem (MORE, 2016; SAHU *et al.*, 2014). Ambos os métodos podem ser aplicados em qualquer sistema de aprendizagem, desde que ocorra na fase de pré-processamento, permitindo o sistema receber as instâncias de treinamento no início do procedimento de classificação (GANGANWAR, 2012). Já segundo Hulse *et al.* (2007), subamostragem pode descartar dados potencialmente úteis, enquanto a sobre-amostragem aumenta artificialmente o tamanho da base de dados e consequentemente aumenta a carga computacional em algoritmos de aprendizado.

2.5.1 Sobre-amostragem

Trata-se de um método que balanceia uma base de dados pelo aumento das amostras das classes minoritárias. A vantagem desse método é que não há perda de dados, enquanto a desvantagem é que pode gerar sobre-ajustes, além de adicionar custo computacional ao sistema. As técnicas de sobre-amostragem se agrupam em duas abordagens: aleatória e instrutiva (SONAK; PATANKAR, 2015). A técnica de sobre-amostragem aleatória é uma simples e efetiva abordagem para re-amostragem. Seu funcionamento consiste em selecionar membros aleatoriamente da classe minoritária, duplicá-los e os adicionar ao novo conjunto de treinamento. Já no caso da técnica instrutiva, as amostras das classes minoritárias são geradas sinteticamente baseados em critérios pré-especificados. Vários algoritmos de sobre-amostragem podem ser encontrados na literatura, dentre os quais podem-se citar: SMOTE (CHAWLA *et al.*, 2002), MTDF (HE *et al.*, 2008) e CUBE (DEVILLE; TILLÉ, 2004).

2.5.2 Subamostragem de classe majoritária

Nesse método de re-amostragem, apenas a classe majoritária tem seu tamanho reduzido, visando diminuir os efeitos do desbalanceamento das bases de dados para melhorar a classificação. Alguns exemplos deste tipo de técnica são: Aleatória, Cluster centroides e NearMiss.

2.5.2.1 Subamostragem aleatória

Nessa abordagem, amostras são aleatoriamente removidas das classes que possuem os maiores números de instâncias, de forma que a sub-base de dados gerada seja o mais próximo possível de uma base balanceada.

2.5.2.2 Subamostragem por Cluster centroides

É uma abordagem, como sugere o nome, em que a subamostragem é realizada por meio do conceito de clusterização. Assim, para cada classe que se deseja sub-amostrar, encontra-se o centroide por meio da média de todas as características. Em seguida, as N amostras mais próximas desse centroide encontrado são as instâncias mais importantes. Vale ressaltar que N varia de acordo com o tamanho da sub-base desejada. Na Figura 5a, duas classes são mostradas, uma majoritária e outra minoritária, representando uma base de dados desbalanceada. Ao aplicar essa técnica de subamostragem, as N mais próximas amostras do centroide da classe #1 são selecionadas, sendo possível balancear a base de dados, conforme se observa na Figura 5b.

2.5.2.3 Subamostragem por NearMiss

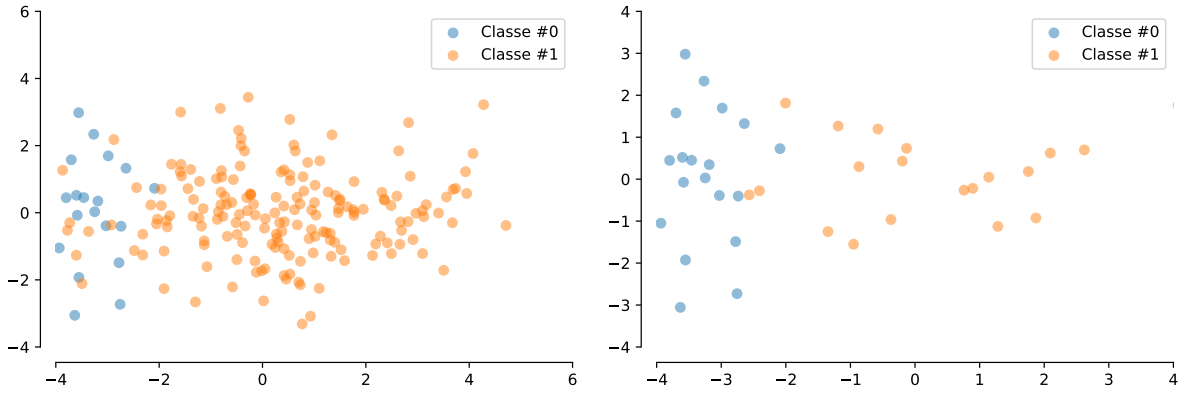
Trata-se de um método de subamostragem baseado na teoria do algoritmo KNN, proposta por (MANI; ZHANG, 2003). Tal algoritmo possui três versões: NearMiss1, NearMiss2, NearMiss3. A primeira versão, NearMiss1, seleciona as amostras da classe majoritária que estão mais próximas das demais classes para a subamostragem. Na Figura 6, tem-se duas classes em que é calculada a distância e os elementos mais próximos são selecionados.

A segunda versão, NearMiss2, seleciona as amostras com maior distância média aos N vizinhos das demais classes. Na Figura 7, tem-se duas classes em que é calculada a distância e os elementos mais distantes são selecionados.

Já na terceira versão do algoritmo, NearMiss3, inicialmente para cada amostra das

Figura 5 – Exemplo de subamostragem de duas classes hipotéticas em uma base de dados desbalanceada, apresentando elementos gerados artificialmente.

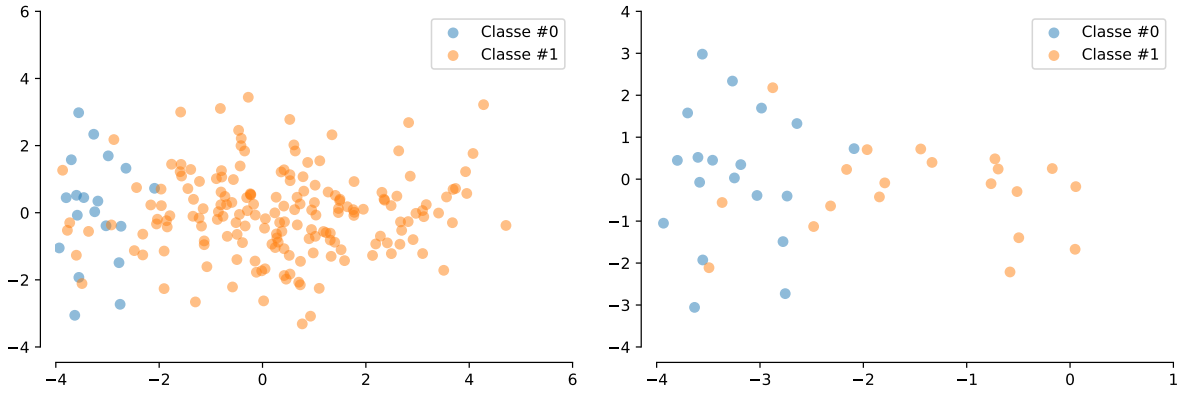
(a) Conjunto original. (b) Subamostragem por clusterização.



Fonte: Elaborado pelo autor.

Figura 6 – Exemplo de subamostragem de duas classes hipotéticas em uma base de dados desbalanceada, apresentando elementos gerados artificialmente.

(a) Conjunto original. (b) Subamostragem por NearMiss1.

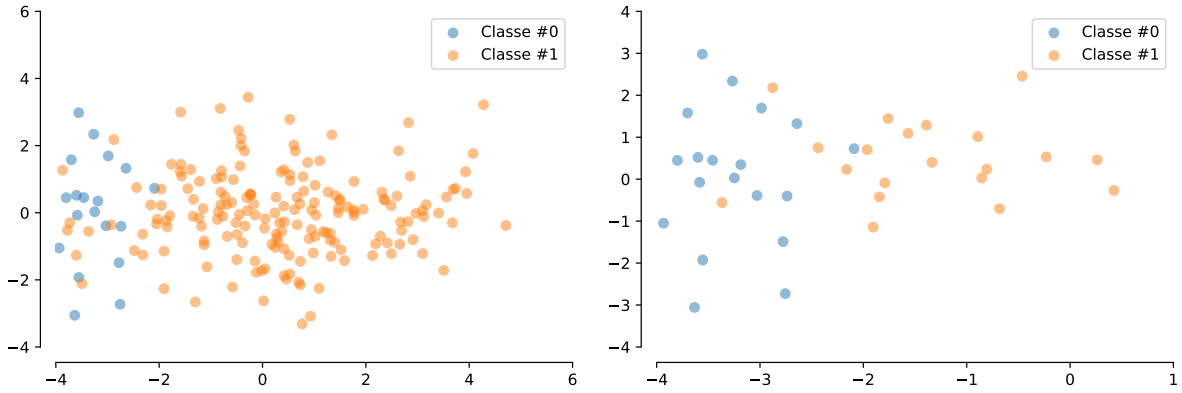


Fonte: Elaborado pelo autor.

classes minoritárias, os elementos da classe majoritária mais próximos são selecionados. Em seguida, as amostras selecionadas são aquelas para as quais a distância média até o N vizinho mais próximo é a maior. Na Figura 8, tem-se duas classes em que para cada elemento da classe minoritária, é calculada a distância e os elementos mais distantes são selecionados.

Figura 7 – Exemplo de subamostragem de duas classes hipotéticas em uma base de dados desbalanceada, apresentando elementos gerados artificialmente.

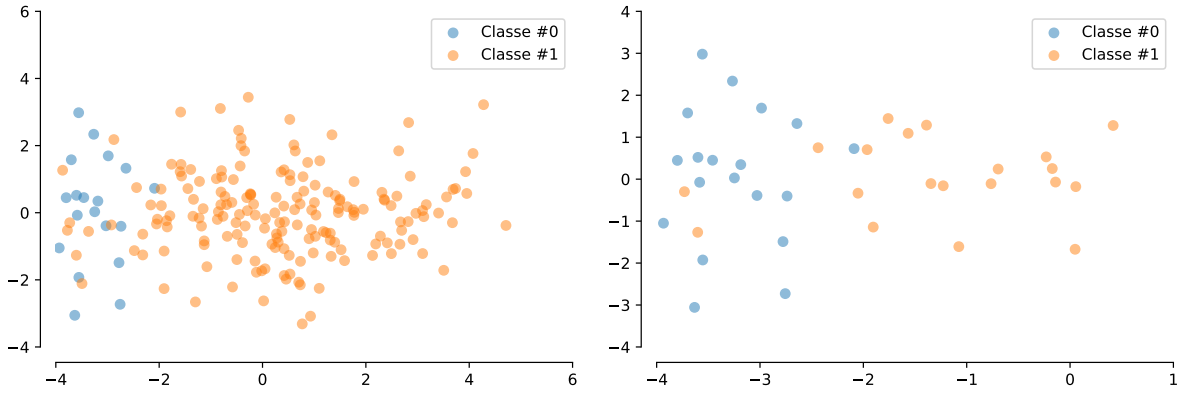
(a) Conjunto original. (b) Subamostragem por NearMiss2.



Fonte: Elaborado pelo autor.

Figura 8 – Exemplo de subamostragem de duas classes hipotéticas em uma base de dados desbalanceada, apresentando elementos gerados artificialmente.

(a) Conjunto original. (b) Subamostragem por NearMiss3.



Fonte: Elaborado pelo autor.

2.6 Testes estatísticos para comparação entre classificadores

Como visto em seções anteriores, algoritmos de aprendizado de máquina são responsáveis por reconhecer um padrão de comportamento e inferir sobre dados novos, podendo ser assertivo ou não, de acordo com a quantidade/generalização dos dados. Mas qual é a abordagem para escolher o melhor algoritmo para utilizar na prática?. Por exemplo, qual é o algoritmo utili-

zado no SDI para o reconhecimento de intrusões em uma rede de um servidor? Para responder esse tipo de pergunta, testes estatísticos são utilizados para a comparação de diferentes algorítimos/metodologias de classificação em diferentes bases de dados. Não existe procedimento padrão para a comparação entre classificadores, de forma que vários autores adotam variadas técnicas para decidirem se as diferenças entre os algoritmos de aprendizagem de máquina são reais ou aleatórias (DIETTERICH, 1998; DEMŠAR, 2006).

Os trabalhos realizados por Dietterich (1998), Demšar (2006) possuem alta relevância no tema pelo número de citações e discutem sobre os testes mais confiáveis para a comparação de algoritmos de classificação supervisionada em diferentes bases de dados. Os autores nestes trabalhos citam e comparam alguns testes relevantes, tais como: teste de McNemar, ANOVA, teste T, Teste de Wilcoxon e de Friedman. Com base nos experimentos realizados, os autores concluem que, para ter uma noção relativa do desempenho de um algoritmo de aprendizagem, podem ser utilizados 5 repetições de 2 folds em validação cruzada (5x2CV) e o teste unicaudal superior de Wilcoxon para situações em que os algoritmos de aprendizagem são eficientes o suficiente para executar 10 vezes em diferentes bases de dados.

2.6.1 Teste unicaudal superior de Wilcoxon

Trata-se de um teste não paramétrico cujo o objetivo é comparar dois conjuntos (A e B) com a assunção de que as duas amostras devem ter a mesma distribuição, apenas uma deslocada da outra (SILVA NETO; GOMES, 2019). Além disso, esse procedimento é mais aplicável devido as amostras não precisarem de tantas condições como no caso do teste T, tais como distribuição normal e homogeneidade em termos de variância. Para avaliar-se duas amostras em termos deste teste, dois valores devem ser calculados: o valor mínimo da soma dos postos positivos e negativos, T, bem como a estatística do teste, z. Em seguida, duas Hipóteses devem ser consideradas: a hipótese nula (H_0) e a Hipótese Alternativa (H_1). A primeira é o ponto de partida antes do teste: as duas amostras não possuem diferenças estatísticas, ou seja, ao aumentar o número de amostras, as duas distribuições vão possuir os mesmos valores de mediana. Entretanto, rejeitar H_0 significa assumir a hipótese alternativa, em que comprova-se a diferença entre as distribuições. No teste unicaudal a hipótese alternativa refere-se que além das amostras A e B serem diferentes, B se sobressai estatisticamente sobre A.

Esse teste envolve os seguintes passos (OTT; LONGNECKER, 2015):

- calcular as diferenças entre os pares de observações;

- remover todos os valores iguais a zero, dado que n é o numero de valores não nulos;
- listar os valores absolutos das diferenças em ordem crescente e atribuir uma posição para cada um deles.
- encontrar o valor crítico da distribuição e;
- rejeitar ou não H_0

Na descrição desse teste, algumas notações são necessárias para seus parâmetros:

- n - número de pares de observações cuja diferença é não nula;
- T_+ - soma das diferenças positivas, se não houver diferença positiva, $T_+ = 0$;
- T_- - soma das diferenças negativas, se não houver diferença negativa, $T_- = 0$;
- T - o menor entre T_+ e T_-, $\min(T_+, T_-)$

com base nestes parâmetros, o teste estatístico z é executado pela aplicação da fórmula

$$z = \frac{T - \frac{n(n+1)}{4}}{\sqrt{\frac{n(n+1)(2n+1)}{24}}}.$$

Após o cálculo do teste, o valor z calculado é comparado com uma tabela de valores de distribuições que pode ser encontrada no trabalho desenvolvido por Sheskin (2003) e os valores encontrados são referentes as hipóteses nula e alternativa, que são as seguintes:

- **Hipótese nula** ($H_0 = 0$) - não há diferença entre as amostras testadas, ou seja, os classificadores comparados possuem métricas equivalentes;
- **Hipótese alternativa** ($H_a < 0$) - segundo Sheskin (2003), as amostras do segundo par de observações são maiores que as do primeiro. No caso de uma comparação entre dois classificadores A e B, o segundo possuiria métricas maiores que as do primeiro.

2.7 Bases de dados de Intrusão

Visando a avaliação de SDIs de forma genérica, bases de dados de intrusão são utilizadas. Várias bases de intrusão podem ser encontradas na literatura (XYLOGIANNOPOULOS *et al.*, 2017), (BEIGI *et al.*, 2014),(JAZI *et al.*, 2017), (RING *et al.*, 2017b), (RING *et al.*, 2017a). Nessa dissertação são abordadas as bases de dados mais recentes disponibilizados no trabalho desenvolvido por Sharafaldin *et al.* (2018): CICIDS2017 e CICIDS2018.

2.8 CICFlowMeter

O CICFlowMeter [1] é um programa que permite analisar e gerar fluxos de dados a partir de arquivos de captura de tráfego tais como tcmpdump e pcap. Este programa é disponibilizado no formato de pacote jar e pode ser encontrado no site do Instituto Canadense de Cibersegurança.

Os arquivos de captura são carregados e convertidos em fluxos de rede no formato CSV, contendo 80 características. Destaca-se que as bases de dados CICIDS2017 e CICIDS2018 são disponibilizados nesse formato. Além disso, essa aplicação permite uniformizar os fluxos de rede, sendo possível converter qualquer arquivo de captura em bases de dados contendo as características assim como os fluxos.

Tal ferramenta busca auxiliar no processo de detecção de ataques, os quais provocam prejuízos de grande escala tanto em servidores de empresas, que armazenam dados de clientes ou que fornecem serviços aos usuários na *internet*. Vale ressaltar que a quantidade de ferramentas utilizadas para a realização desses ataques é amplamente disseminada na rede mundial de computadores, além de possuírem interface amigável e fácil de utilizar. Assim, Sistemas de Detecção de Intrusão são estudados e desenvolvidos, visando a efetiva detecção desses tipos de ameaças. Nesse contexto, diversos trabalhos podem servir como referência para obter-se uma linha de raciocínio para o desenvolvimento de SDIs.

[1] <https://www.netflowmeter.ca/>

3 TRABALHOS RELACIONADOS

No projeto de um SDI baseado em aprendizagem de máquina, a tomada de decisão pelo algoritmo de classificação é uma atividade fundamental que engloba um conjunto específico de critérios passíveis de erro (SILVA NETO; GOMES, 2019). Neste processo, os sistemas devem lidar com os dados de forma eficiente, uma vez que em um cenário ideal o tempo de treinamento, testes e a carga computacional dispensada no processamento para detecção de intrusões devem ser minimizados. Neste contexto, diferentes abordagens tem sido utilizadas para equilibrar o compromisso entre o tempo de processamento do SDI e sua assertividade. Encontram-se na literatura trabalhos que abordam o projeto de SDIs baseados em aprendizagem de máquina, em que adotam-se técnicas para otimizar o processo de seleção dos algoritmos de classificação. Alguns destes trabalhos são descritos a seguir.

Uma parcela de 20 % da base de dados NSL-KDD foi utilizada por Dhanabal e Shantharajah (2015) para avaliar 3 algoritmos de aprendizado de máquina utilizando a ferramenta WEKA nos experimentos. A métrica de avaliação utilizada foi a acurácia, sendo possível alcançar taxas de até 99.8% com redução de 41 para 6 características. Entretanto, não foi detalhado pelos autores como os 20% da base de dados foram obtidos, dificultando desta forma a replicação do trabalho.

Outra abordagem utilizando 20% do NSL-KDD pode ser encontrada no trabalho realizado por Aljawarneh *et al.* (2018). Nesse trabalho, os autores utilizam classificação binária na base de dados. Assim, os rótulos das classes referentes a ataques foram unificados representando um padrão anormal de tráfego. Além disso, um modelo de detecção baseado em um votador entre 7 algoritmos clássicos foi proposto, bem como a utilização do algoritmo Information Gain para reduzir o número de características de 41 para 8. Os resultados obtidos pelos autores indicam que o modelo proposto possui altas taxas acurácia, entre 90% e 99% e baixa taxa de falsos positivos, entre 0.003% e 0.102%. Entretanto, em cenários reais que há restrições de processamento, tal modelo mostra-se complexo computacionalmente, sendo necessário executar 7 algoritmos diferentes para realizar a votação.

O trabalho realizado por Bhaskar *et al.* (2019) propõe uma técnica de seleção de características para avaliar na base de dados NSL-KDD, buscando minimizar as taxas de falsos positivos e negativos, bem como maximizar a taxa de detecção. Entretanto, nenhuma metodologia de avaliação foi aplicada, seja por algoritmos ou métricas de avaliação. Além disso, os experimentos foram realizados utilizando uma base de dados obsoleta e não avalia Sistemas

de Detecção de Intrusão em sua metodologia.

Gao *et al.* (2019) utilizaram a base de dados NSL-KDD [1] como objeto de pesquisa para propor um algoritmo combinado de aprendizagem, semelhante ao trabalho discutido anteriormente. Foram utilizados 5 classificadores para a composição do sistema votador e os experimentos foram realizados utilizando 2-Folds em validação cruzada, obtendo-se resultados entre 73% 79% para acurácia e Sensibilidade, 80% e 84% especificidade e 69% e 80% para eficiência. Além disso, para lidar com o desbalanceamento do base de dados, a técnica de subamostragem aleatória foi utilizada. Entretanto, o número de bases subamostradas aleatoriamente não foi explicitado, podendo levar a crer que os dados não necessariamente estão bem representados por conta da aleatoriedade de apenas uma sub-base gerada.

Um algoritmo de seleção de características baseados em duas funções objetivas chamado MOEDAFS foi proposto por Maza e Touahria (2019). Os experimentos foram conduzidos por meio da criação de diferentes sub-bases de dados com número de características entre 5 e 22, usando o NSL-KDD como objeto de estudo e 7 algoritmos de aprendizagem de máquina. Os resultados apresentados em termos de acurácia variam entre 81% e 98%.

Utimura e Costa (2018) realizaram uma subamostragem de 10% na base de dados ISCX2012 [2] na avaliação de algoritmos destinados a criação de um *plugin* para o *Snort* baseado nos classificadores *Optimum-Path Forest* (OPF) e *Multi-layer Perceptron MLP*. Os autores adotaram diferentes proporções de treino/teste na parcela selecionada, mantendo as proporções entre as classes na base de dados original, sendo possível alcançar resultados de até 97% em termos de acurácia média. Entretanto, o número de bases subamostradas aleatoriamente não foi explicitado, podendo levar a crer que os dados não necessariamente estão bem representados por conta da aleatoriedade de apenas uma sub-base gerada.

Ullah e Mahmoud (2017) apresentaram um *framework* para SDIs com foco em cenários IoT aplicados a *SmartGrid*, no qual existem restrições de recursos como energia e capacidade de processamento. Os autores utilizaram uma abordagem baseada em subamostragem na avaliação dos algoritmos de aprendizado de máquina para a detecção de ataques nesse tipo de ambiente. Assim, 20% da base ISCX2012 (SHIRAVI *et al.*, 2012) foi utilizado para treino e 80% para testes, variando-se diferentes valores de *K-Fold cross-validation*. É perceptível que a abordagem não é recomendada na avaliação de algoritmos que possuam alto custo computacional em sua fase de testes.

[1] <https://www.unb.ca/cic/datasets/nsl.html>
[2] <https://www.unb.ca/cic/datasets/ids.html>

Sharafaldin *et al.* (2018) aplicaram uma técnica de seleção de características chamada *Mean Decrease Impurity* (MDI) no conjunto de dados CICIDS2017 para avaliação de sete algoritmos de aprendizagem de máquina. Esta técnica permitiu selecionar as melhores características para cada um dos 15 tipos de tráfego, reduzindo assim a quantidade de dados utilizados para treinamento e teste dos classificadores no processo de seleção dos algoritmos. Os resultados dos experimentos foram de 77% a 98% para precisão, 4% e 98% para revocação e 4% e 94% para Eficiência. No entanto, apenas a técnica de seleção das melhores características ainda resultou em uma expressiva quantidade de informações e em um tempo elevado para o treinamento e testes dos classificadores.

SILVA NETO e Gomes (2019) avaliaram a *performance* de oito algoritmos de detecção de intrusão baseados em aprendizagem de máquina na base de dados CICIDS 2017 (SHARAFALDIN *et al.*, 2018), utilizando diferentes técnicas de amostragem. Além disso, foi utilizada a técnica MDI para reduzir o número de características da base de dados, alcançando altas taxas de detecção em termos de precisão, Revocação e Eficiência, bem como tempo de processamento. Os resultados dos experimentos foram de 75% a 99% para precisão, 55% e 98% para revocação e 63% e 99% para Eficiência Entretanto, por utilizar toda a base de dados em sua avaliação, não foi possível avaliar algoritmos com alta carga computacional, tais como SVM.

Aksu e Aydin (2018), avaliaram os algoritmos SVM e *deep learning* na detecção de ataques do tipo *PortScan* na base de dados CICIDS2017. Os experimentos conduzidos pelos autores consistiram em separar 10% da base de dados contendo todos os ataques, adotando apenas tráfegos caracterizados como *PortScan* e normais. Com a parcela selecionada, foi utilizado 67% para treino e 33% para testes, obtendo-se valores médios de acurácia entre 69% e 97%. Tratando-se de dois algoritmos custosos computacionalmente, os autores abordaram apenas um tipo de ataque em sua avaliação.

Duas abordagens de redução de dimensionalidade foram utilizadas por Abdulhammed *et al.* (2019): Principal Component Análisys (PCA) e Auto-Encoder, visando avaliar algoritmos tais como Random Forest, Redes Bayesianas, Linear Discriminant Analysis (LDA) e Quadratic Discriminant Analysis (QDA) na base de dados CICIDS2017 nas abordagens binária e multi-classes. Os experimentos conduzidos permitiram reduzir o número de características de 80 para 10, mantendo altas taxas de acurácia (entre 85% e 99%). Entretanto, a PCA por transformar as características em componentes e consequentemente modificar o eixo de representação dos dados, acarreta em perda de informação original ao reduzir os componentes, tornando assim

difícil a reconstrução dos dados e uso no contexto de análise de tráfegos reais.

O trabalho realizado por D'hooge *et al.* (2019) avalia 12 algoritmos de aprendizado de máquina em 4 bases de dados de forma abrangente: NSL-KDD, ISCX2012, CIC2017 e CIC2018. O design experimental no trabalho consistiu na redução vertical (subamostragem) e horizontal (redução de características), obtendo resultados expressivos (até 99% de acurácia) utilizando 1% dos dados para treinamento e 99% para testes. Entretanto, o trabalho é de difícil replicação por alguns motivos:

- a técnica de subamostragem utilizada não foi explicitada, não se sabem quais amostras foram selecionadas pelos autores;
- a redução de características foi realizada baseada em métodos empíricos, ou seja, as características removidas foram justificadas no trabalho porque as mesmas contaminavam os resultados ou eram redundantes ou até mesmo problemáticas; (D'HOOGE *et al.*, 2019) e não foram especificados detalhes sobre esta "contaminação";
- os códigos dos experimentos, bem como os arquivos CSV das bases de dados utilizadas foram disponibilizadas para consulta, entretanto, observa-se que: a base de dados CIC2018 não está completa e técnicas não explicitadas no artigo tais como a Análise dos Componentes Principais (PCA) são utilizadas.

Uma comparação resumida entre os trabalhos relacionados e esta dissertação encontra-se na Tabela 1.

Tabela 1 – Comparação entre os trabalhos relacionados e esta dissertação

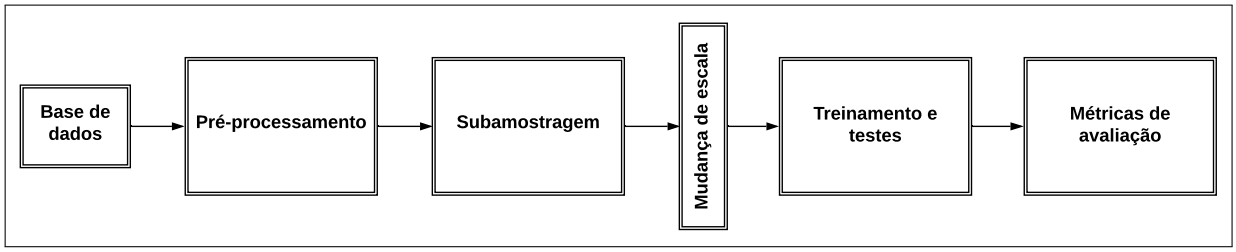
Trabalho	Base de dados	Estratégia de amostragem	Técnica de subamostragem	Algoritmos avaliados	Métricas de avaliação	Teste Estatístico
Dhanabal e Shantharajah (2015)	NSL-KDD	Não está claro	Não foi especificada (20%)	J48, SVM e Naive Bayes	Acurácia	Nenhum
Aljawarneh *et al.* (2018)	NSL-KDD	10-Fold CV	20% NSL-KDD treinamento	J48, Meta Pagging, RandomTree, REPTree, AdaBoostM1, DecisionStump e Naive Bayes	Acurácia, Falsos positivos, Falsos negativos, Verdadeiros positivos, Verdadeiros negativos	Nenhum
Bhaskar *et al.* (2019)	NSL-KDD	Nenhum	Nenhuma	Nenhum	Nenhuma	Nenhum
Gao *et al.* (2019)	NSL-KDD	2-Fold CV	Subamostragem Aleatória	DecisionTree, RandomForest, KNN, LR, SVM, DNN, AdaBoost	Acurácia, Precisão, Revocação e Eficiência	Nenhum
Maza e Touahria (2019)	NSL-KDD	80/20 Train Test Split	Nenhuma	Naive Bayes, MLP, SVM, KNN, DecisionTree	Acurácia, Precisão, Revocação e Eficiência	Nenhum
Utimura e Costa (2018)	ISCX2012	50/50 Train/Test Split, 60/40 Train/Test Split, 70/30 Train/Test Split, 80/20 Train/Test Split	Não foi especificada (10%)	OPF e MLP	Acurácia, Tempos de treinamento e teste	Nenhum
Ullah e Mahmoud (2017)	ISCX2012	80/20, 3-Fold CV, 5-Fold CV, 10-Fold CV, 15-Fold CV, 20-Fold CV	Nenhuma	J48, JRip, Naive Bayes, SVM, MLP	Precisão, Revocação e Eficiência	Nenhum
Sharafaldin *et al.* (2018)	CICIDS2017	Não está claro	Nenhuma	KNN, RamdonForest, ID3, AdaBoost, MLP, Naive-Bayes, QDA	Precisão, Revocação, Eficiência e Tempo de execução	Nenhum
SILVA NETO e Gomes (2019)	CICIDS2017	10-Fold CV, 10-Fold Stratified CV	Nenhuma	Nearest Centroid, NaiveBayes, AdaBoost, MLP, DecisionTree, KNN, Random Forest e QDA	Precisão, Revocação, Eficiência e Tempo de teste	Wilcoxon
Aksu e Aydin (2018)	CICIDS2017	67/33 Train/Test Split	Nenhuma	Deep learning e SVM	Acurácia, Precisão, Revocação e Eficiência	Nenhum
Abdulhammed *et al.* (2019)	CICIDS2017	70/30 Train/Test Split	Nenhuma	RamdonForest, Naive Bayes, LDA, QDA	False Alarm Rate, Acurácia, Detection Rate, Precisão, Revocação, Eficiência	Nenhum
D'hooge *et al.* (2019)	NSL-KDD, ISCX2012, CICIDS2017, CICIDS2018,	1/99 Stratified Train/Test Split, 10/90 Stratified Train/Test Split, 20/80 Stratified Train/Test Split, 30/70 Stratified Train/Test Split, 40/60 Stratified Train/Test Split, 50/50 Stratified Train/Test Split	Não está claro	DecisionTree, Bagging, AdaBoost, Gradient Boosted Trees, Regularized Gradient Boosting, RamdonForest, ExtraTrees, KNN, Nearest Centroid, Linear SVM, RBF SVM, Logistic Regression	Acurácia, Precisão, Revocação, Eficiência e ROC	Nenhum
Esta Dissertação	CICIDS2017, CICIDS2018	5x 2CV	Aleatória, Cluster Centroides, Near Miss	Centroide mais próximo, Naive Bayes, Ramdon Forest, KNN, SVM	Acurácia, Precisão, Revocação, Eficiência e Tempo de Treinamento	Wilcoxon

Esta dissertação, diferentemente dos demais trabalhos, avalia técnicas de subamostragem de forma sistemática e adaptável para novos cenários. Para isto, avaliaram-se algoritmos de aprendizagem de máquina em diferentes bases subamostradas de acordo com técnicas encontradas na literatura, visando a melhor escolha para cenários reais nos quais hajam restrições de bateria/processamento para lidar com uma grande massa de dados ou em casos em que deseja-se uma resposta no menor tempo possível.

4 METODOLOGIA

Neste Capítulo, a metodologia do trabalho é apresentada. Os experimentos seguem um fluxo sistemático que permite sua reprodução e extensão com posterior substituição dos itens utilizados. Os experimentos consistem no pré-processamento da base de dados, seguida de subamostragens das classes majoritárias para criação de sub-bases para treinamento/testes de acordo com a técnica utilizada, conforme mostrado no diagrama de alto nível da Figura 9. O objetivo principal é a análise da viabilidade do uso de apenas uma subamostra da base de dados no projeto dos SDIs. A análise da capacidade de generalização é feita por meio da comparação do desempenho dos classificadores nas sub-bases geradas e em suas respectivas bases de dados originais.

Figura 9 – Diagrama de alto nível da metodologia empregada



Fonte: elaborado pelo autor.

4.1 Bases de dados de Intrusão utilizadas

4.1.1 Base de dados CICIDS2017

A fim de suprir a necessidade de uma base de dados pública e atualizada, Sharafaldin *et al.* (SHARAFALDIN *et al.*, 2018) propôs a base CICSDI2017, que consiste de dados reais capturados durante 5 dias e está disponível para fins de pesquisa pelo *Canadian Institute for Cybersecurity* (CIC)[1]. A base dispõe de 78 características e contém 14 tipos de ataques atualizados, tais como DDoS, DoS, Brute Force, bem como tráfego normal. Os nomes das classes , bem como o quantitativo, porcentagens e índices de desbalanceamento (I_D) são encontrados na Tabela 2.

Ao observar essa tabela, percebe-se que se trata de uma base de dados desbalanceada, uma vez que dos 15 tipos de classe, 4 delas concentram 98,2% de todas as amostras. Além disso, o índice de desbalanceamento na base de dados alcança valores altos tais como no caso

[1] <https://www.unb.ca/cic/datasets/\gls{SDI}-2017.html>

Tabela 2 – Quantidade de registros por classe na base de dados CIC2017

Classe	Número de amostras	% da classe	I_D
Classe	Número de amostras	% da classe	I_D
Normal	2273097	80,3	1:1
DoS Hulk	231073	8,1	9:1
PortScan	158930	5,6	14:1
DDoS	128027	4,5	17:1
Dos GoldenEye	10293	0,36	220:1
FTP-Patator	7938	0,28	286:1
SSH-Patator	5897	0,20	385:1
DoS Slowloris	5796	0,20	392:1
DoS Slowhttptest	5499	0,19	413:1
Bot	1966	0,069	1156:1
Web Attack Brute Force	1507	0,053	1508:1
Web Attack XSS	652	0,0023	3486:1
Infiltration	36	0,00012	63141:1
Web Attack SQL Injection	21	0,000074	108242:1
Heartbleed	11	0,000038	206645:1
Total	2830743	-	-

Fonte: o autor.

dos ataques Heartbleed e SQL Injection. Tal fato deve ser levado em consideração no projeto de SDIs, de forma que bases de dados muito desbalanceadas podem ocasionar problemas de assertividade devido à uma baixa representatividade das classes minoritárias. Para isso, uma abordagem que selecione os principais amostras nesse cenário pode ser uma solução.

4.1.2 Base de dados CICIDS2018

Essa é a base de dados mais recente encontrada na literatura e ainda possui poucas avaliações. Tal base é resultado de um projeto entre o Communications Security Establishment (CSE) e o Canadian Institute for Cibersecurity (CIC) que utilizaram perfis de rede para a geração de 15 tráfegos, sendo um deles normal e 14 ataques. A infraestrutura de ataque consiste em 50 máquinas e a organização atacada possui 5 departamentos com 420 computadores e 30 servidores. Os dados consistem de 10 dias de monitoramento, 80 características em arquivos com valores separados por vírgulas (.CSV). Na Tabela 3 são apresentados os nomes das classes, bem como os quantitativos e as porcentagens das amostras para cada uma delas. Ao observar essa tabela, nota-se que tal como a base CICIDS2017, o tráfego normal possui 83% do número de amostras da base de dados, configurando o desbalanceamento. Além disso, o índice de desbalanceamento na base de dados alcança valores altos tais como no caso dos ataques Brute Force - XSS e SQL Injection.

Tabela 3 – Quantidade de registros por classe na base de dados CIC2018

Classe	Número de amostras	% da classe	I_D
Normal	13609917	83,20	1:1
DDOS attack-HOIC	686012	4,19	19:1
DDoS attacks-LOIC-HTTP	576191	3,52	23:1
DoS attacks-Hulk	461912	2,82	29:1
Bot	286191	1,74	47:1
FTP-BruteForce	193354	1,18	70:1
SSH-Bruteforce	187589	1,14	72:1
Infiltration	160639	0,98	84:1
DoS attacks-SlowHTTPTest	139890	0,85	97:1
DoS attacks-GoldenEye	41508	0,25	327:1
DoS attacks-Slowloris	10990	0,067	1238:1
DDOS attack-LOIC-UDP	1730	0,0105	7867:1
Brute Force -Web	611	0,0037	22274:1
Brute Force -XSS	230	0,0014	59173:1
SQL Injection	87	0,00053	156435:1
Total	16356851	100	-

Fonte: o autor.

4.2 Fluxo sistemático da metodologia empregada

Uma visão geral de alto nível da metodologia empregada é apresentada na Figura 10. A fase de pré-processamento seguiu a metodologia proposta por (SILVA NETO; GOMES, 2019) o qual resultou na concatenação de todos os arquivos das bases de dados, seguido da remoção de registros contendo dados inválidos, bem como remoção de características com média e desvio padrão zero para todas as amostras. Na fase de subamostragem, as bases de dados foram reduzidas seguindo o tamanho completo da base de dados NSL-KDD [2], a qual contém 173709 registros, uma vez que trata-se de uma base amplamente explorada na literatura. Assim, as seguintes técnicas de subamostragem de classe majoritária foram utilizadas: aleatória, Cluster centroides e NearMiss1. Salienta-se que o objetivo desta fase é de gerar sub-bases de dados o mais balanceadas possível de acordo com o número de registros escolhido. Para tal fim, os registros das classes majoritárias foram removidos, preservando as classes minoritárias.

Para a primeira subamostragem, dez sub-bases de dados aleatórias foram geradas, pois pelo teor randômico da técnica, busca-se a estabilidade nos dados selecionados. Desta forma a avaliação do desempenho nesse tipo de subamostragem é dada pela média de todas as métricas dos subconjuntos treinados/testados.

A análise do centroide é utilizada na segunda abordagem para a geração da outra base subamostrada. Assim, o centroide é calculado para todas as classes e as n amostras as quais

[2] <https://www.unb.ca/cic/datasets/nsl.html>

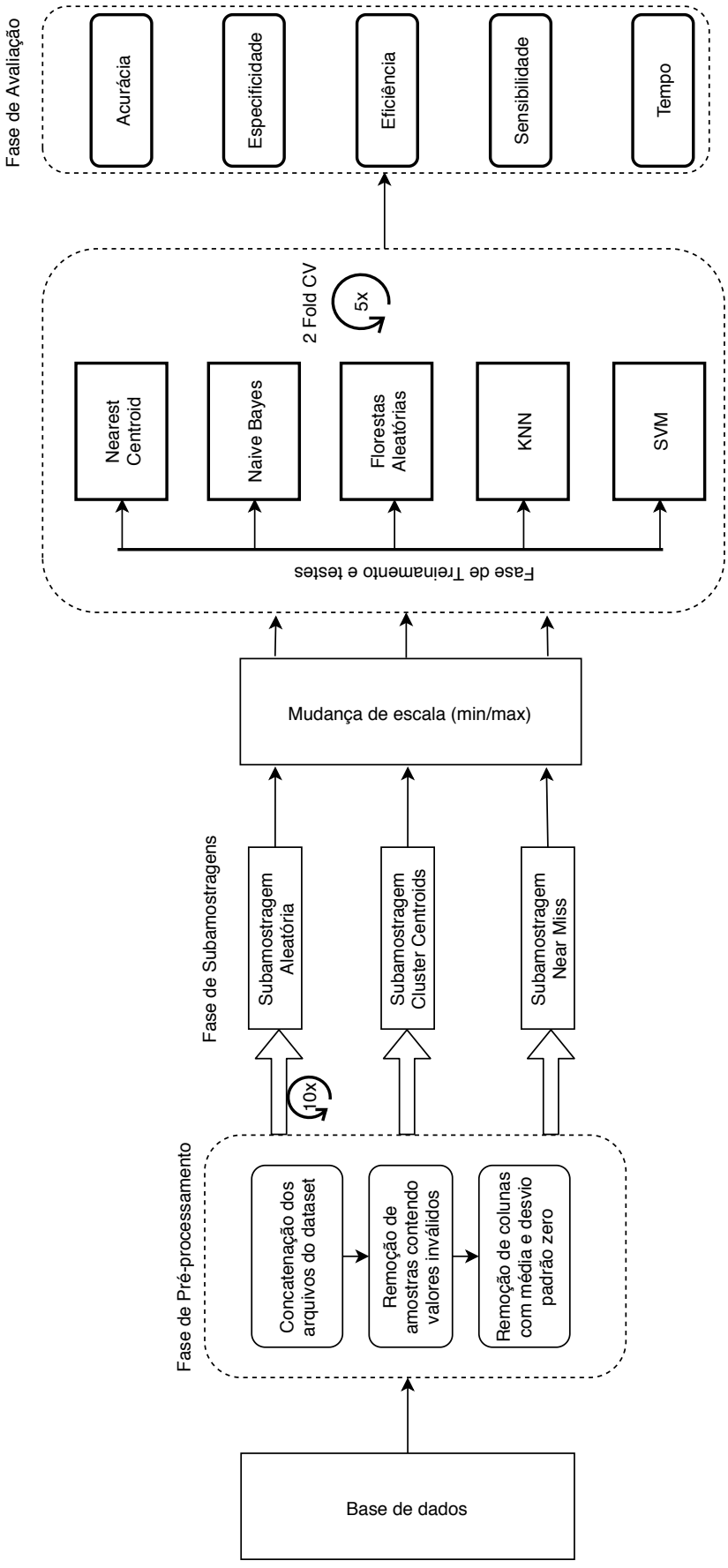


Figura 10 – Fluxograma da metodologia empregada na dissertação.

possuem as menores distâncias para o centroide foram selecionadas. Vale ressaltar que n é o tamanho da base desejada, nesse caso o mesmo tamanho da base anterior.

Na terceira abordagem, a redução das amostras nas classes majoritárias ocorre de forma que as k instâncias escolhidas são as que possuem menor distância para as amostras das classes vizinhas.

Na fase de mudança de escala, a técnica chamada de Min-max scaler proposta por Pedregosa *et al.* (2011) é utilizada baseada na seguinte fórmula: $X_{MinMax} = \frac{X - X_{min}}{X_{max} - X_{min}}$, nos quais X_{MinMax} é o vetor com escala entre 0 (zero) e 1 (um), X é o vetor original, X_min e X_max são os menores e maiores elementos do vetor X, respectivamente.

Em seguida, é avaliada a performance dos classificadores Naive-Bayes (NB), KNN, Nearest-Centroid (NC), Random Forests (RF) e Support Vector Machine (SVM). A escolha dos classificadores foi balizada nos seguintes princípios: três algoritmos baseados em distância - NC, linear, enquanto o SVM e KKN não-lineares, Naive Bayes que é baseado em probabilidade e o Florestas Aleatórias que lida bem com bases desbalanceadas, desfavorecendo subamostragens mais balanceadas. Na fase de classificação, cada algoritmo é treinado e testado 10 (dez) vezes por meio da técnica de validação cruzada, em que os dados foram separados em dois subconjuntos mutualmente exclusivos (*folds*) repetindo por 5 vezes: uma para treinamento e outra para testes. Tal técnica foi adotada de acordo com Dietterich (1998) o qual recomenda experimentos com cinco repetições de duas *folds* de validação cruzada, visando testes estatísticos tais como Wilcoxon e Friedman. Os parâmetros do sistemas avaliados foram definidos de acordo com Pedregosa *et al.* (2011) e são mostrados na Tabela 4.

Tabela 4 – Tabela de parâmetros utilizados nos experimentos

Algoritmo	Parâmetros	% Valor
NC	- Medida de distância	- Euclidiana
RF	- Número de estimadores - Critério	- 100 - Índice de Gini
KNN	- Número de vizinhos - Métrica de distância	- 3 - Euclidiana
SVM	- C - Kernel	- 1.0 - RBF

Fonte: o autor.

4.2.1 Ambiente de desenvolvimento

Os experimentos são realizados em computador com sistema operacional Linux, distribuição Ubuntu 16.04 LTS, processador Core Intel i7-6700-K (8 Núcleos) e 32GB de RAM. Utilizou-se a linguagem de programação Python3 e o pacote scikit-learn (PEDREGOSA *et al.*, 2011) para implementação dos classificadores. O pacote inbalanced-learn (LEMAîTRE *et al.*, 2017) foi utilizado para as técnicas de subamostragem.

4.2.2 Métricas de avaliação

Nessa seção, as métricas utilizadas no trabalho são apresentadas. Vale ressaltar que nesse trabalho as bases de dados contém múltiplos tráfegos (15 ataques e 1 normal) de forma desbalanceada, conforme discutido anteriormente. Portanto, é necessária uma avaliação que leve em consideração a assertividade entre todos os tipos de fluxo de rede de forma ponderada. Segundo (VLUYMANS, 2019) e (FLACH; KULL, 2015), o desbalanceamento entre as classes pode "mascarar"os resultados e consequentemente levar a conclusões precipitadas. Desta forma as métricas ponderadas a seguir são apresentadas

4.2.2.1 Acurácia (AC)

Segundo (VLUYMANS, 2019), essa é a métrica mais intuitiva dentre as formas de avaliação de um modelo, pois representa uma medida da proporção de predições corretas, sem levar em consideração falsos positivos ou negativos. Dado X como um conjunto de elementos de teste para cada rótulo das classes que foram preditas e $corr(.)$, uma função que conta o número de predições corretas, a acurácia pode ser definida por: $Acc(X) = \frac{corr(X)}{|X|}$

4.2.2.2 Recall

Trata-se da proporção dos casos positivos identificados corretamente. A notação das formulas a seguir faz-se necessária

- y é o conjunto de amostras preditas pelo modelo
- $\hat{y}$ é o conjunto de rótulos de teste
- L é o conjunto de rótulos
- y_l é o subconjunto de y com rótulo l

$$R_{ponderada}(y,\hat{y}_l) = \frac{1}{\sum_{l\in L}|\hat{y}_l|}\sum_{l\in L}^{max}|\hat{y}_l|R(y_l,\hat{y}_l), \tag{4.1}$$

em que $R(y_l,\hat{y}_l) = \frac{|y_l \cap \hat{y}_l|}{|\hat{y}_l|}$

4.2.2.3 Precision

Esta métrica representa a fração de predições positivas que foram corretas. Trata-se de uma métrica complementar a anterior. A Equação 4.2 apresenta a sua formulação:

$$Pr_{ponderada}(y,\hat{y}_l) = \frac{1}{\sum_{l\in L}|\hat{y}_l|}\sum_{l\in L}^{max}|\hat{y}_l|P(y_l,\hat{y}_l) \tag{4.2}$$

em que $P(y_l,\hat{y}_l) = \frac{|y_l \cap \hat{y}_l|}{|\hat{y}_l|}$.

4.2.2.4 F1

Para avaliar o compromisso entre as duas métricas citadas anteriormente, a medição eficiência, também chamada de F_β é utilizada

$$F_{\beta ponderada}(y,\hat{y}_l) = \frac{1}{\sum_{l\in L}|\hat{y}_l|}\sum_{l\in L}^{max}|\hat{y}_l|F_\beta(y_l,\hat{y}_l), \tag{4.3}$$

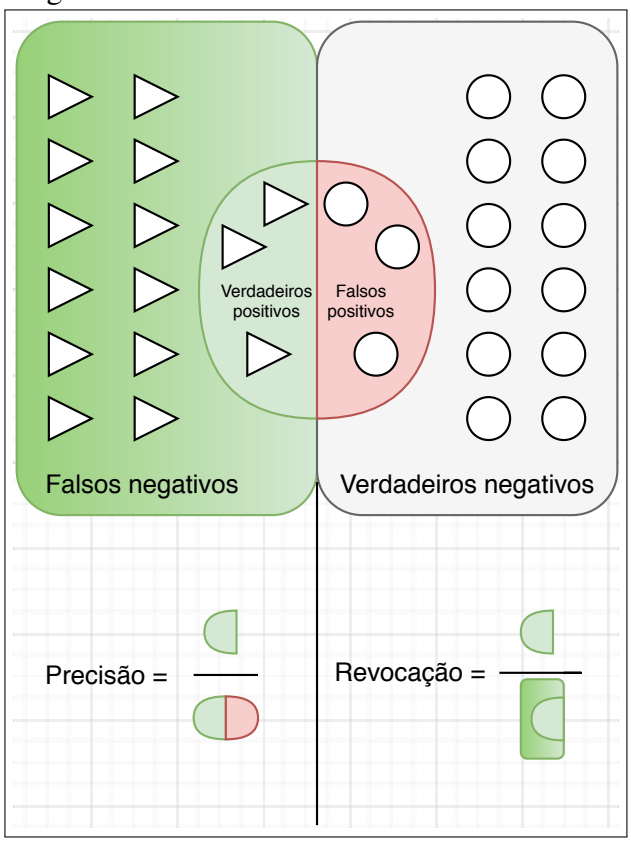
em que

$$F_\beta = (1+\beta^2)\frac{P(A,B)XR(A,B)}{\beta P(A,B)+R(A,B)} \tag{4.4}$$

Segundo (VLUYMANS, 2019), o parâmetro β pode ser qualquer valor real positivo, mas geralmente é definido como 1. Nesse caso, a métrica é simplificada e é dada por $F = \frac{2.Pr.R}{Pr+R}$, que é a média harmônica entre precision e recall.

Como pode-se observar, cada métrica, exceto a acurácia, tem seu peso calculado baseado no número de amostras. Além disso, precision e recall são métricas complementares. A Figura 11 ilustra esse comportamento, em que percebe-se que a recall é calculada pela razão entre o numero de verdadeiros positivos e o total de triângulos, enquanto a precision leva em consideração a quantidade de triângulo com relação a soma de verdadeiros e falsos positivos.

Figura 11 – Precision x Recall



Fonte: elaborado pelo autor.

Assim, a precision trata-se da razão entre a quantidade de elementos de uma classe preditos corretamente pelo classificador (verdadeiros positivos), pela soma das quantidades de elementos de uma classe preditos corretamente e a quantidade de elementos de outras classes preditos erroneamente como elementos da mesma (falsos positivos). Já a recall trata-se da razão entre a quantidade de elementos de uma classe preditos corretamente pelo classificador (verdadeiros positivos) e a soma entre a taxa de verdadeiros positivos e a quantidade de elementos da mesma classe preditos erroneamente (Falsos negativos). Desta forma, a precision pode ser utilizada em situações em que os Falsos positivos são considerados mais prejudiciais que os falsos negativos. No caso da recall, a mesma é utilizada quando os falsos negativos são mais prejudiciais que os falsos positivos. Esta dissertação utiliza as métricas de forma ponderada de acordo com o número de amostras de cada classe conforme as Equações 4.1, 4.2 e 4.3

5 RESULTADOS E DISCUSSÕES

Neste Capítulo, os resultados encontrados pela aplicação da metodologia descrita no Capítulo 4 são apresentados e discutidos. Inicialmente, uma análise exploratória das bases de dados é realizada, de modo a compreender os possíveis fatos observados na obtenção dos resultados. Em seguida, os resultados dos experimentos referentes à aplicação da metodologia descrita no capítulo anterior são mostrados. Os experimentos estão divididos com respeito às bases de dados empregadas na obtenção dos resultados, sendo adotado de Experimentos A e B para as bases CICDS2017 e CICDS2018, respectivamente. Assim, para cada um destes Experimentos, são obtidos os resultados referentes a cada uma das sub-bases resultante da aplicação das técnicas de subamostragem: aleatória, Cluster centroides e NearMiss1. Já a comparação de desempenho entre os classificadores é feita por meio do teste estatístico de *Wilcoxon*, visando encontrar diferenças estatísticas entre os pares de amostras das métricas descritas no Capítulo 4. Além disso, apresentam-se discussões específicas para cada resultado e também um comentário crítico mais geral sobre as características dos classificadores, das bases e das sub-bases geradas.

5.1 Análise exploratória das bases de dados

Visando um melhor entendimento do comportamento dos dados, bem como da interpretação dos resultados dos experimentos, uma análise exploratória dos dados é necessária. Assim, foram realizadas análises uni-variadas e multivariadas de ambas as bases de dados utilizadas nesta dissertação.

CICIDS2017

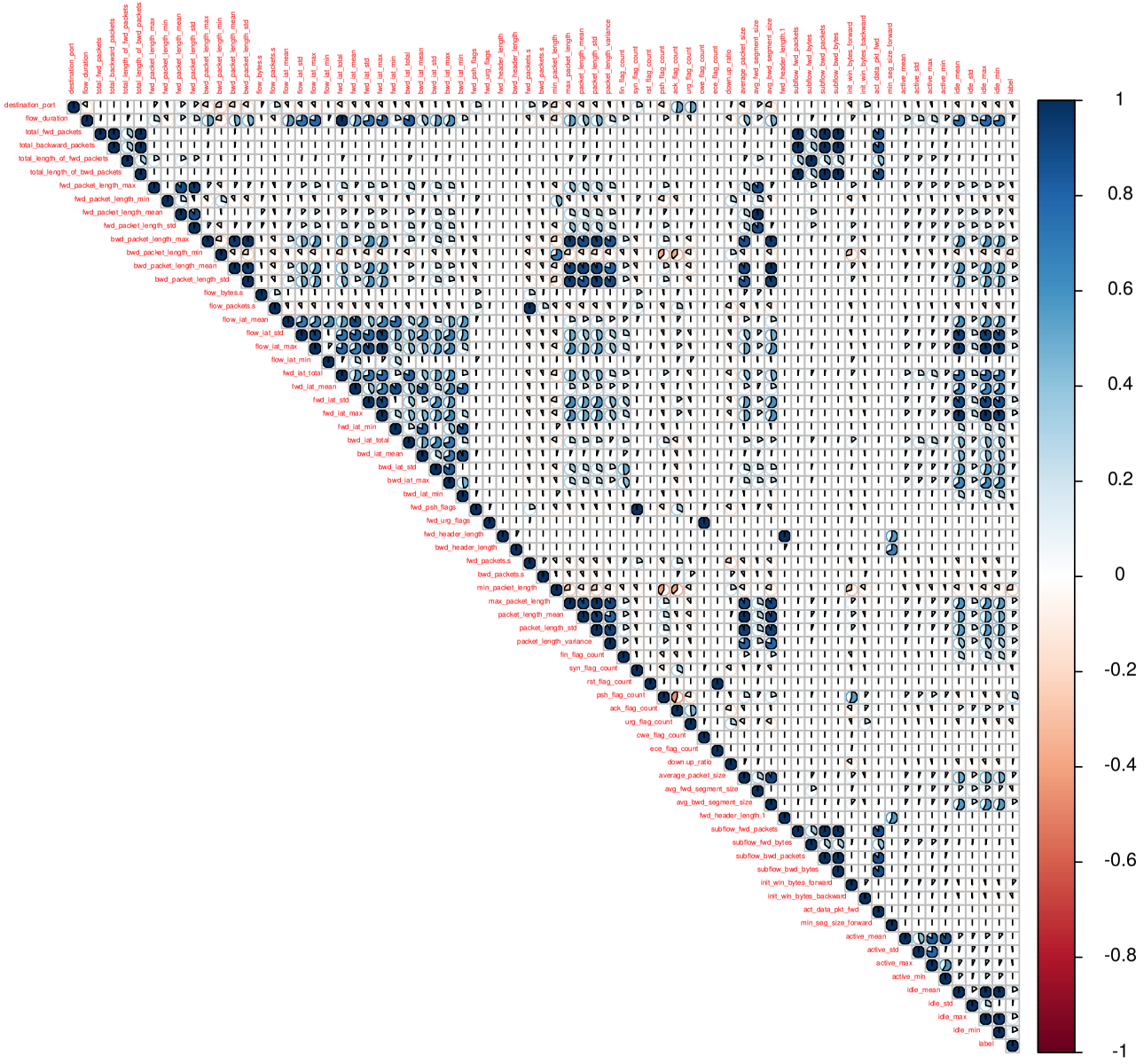
Ao analisar a base de dados, nota-se que para o tráfego normal, 53788 portas diferentes foram utilizadas, enquanto para os ataques, 1686, nas quais as portas 53, 443 (DNS e HTTPS) e 80,21 (HTTP e SSH) são as mais frequentes para tráfegos normais e ataques, respectivamente. Na Tabela 21 no Apêndice A são mostrados a média, desvio padrão, mínimo e máximo de cada característica da base CIC2017. Observa-se que algumas características tais como *flow_duration* e *flow_bytes/s* possuem grande variedade de valores ao longo das amostras, evidenciando a necessidade da troca de escala nas características. Além disso, tem-se que algumas características, tais como *bwd_psh_flags*, *bwd_urg_flags* e *fwd_avg_bytes/bulk* são

repletas de zeros, indicando que não há variação entre as mesmas ao longo dos diferentes tipos de tráfego. Outra análise na Tabela permite avaliar a simetria dos dados na base por meio da coluna Obliquidade. Desta forma, observa-se que a maioria das características possui assimetria deslocada para direita, indicando que a distribuição dos preditores é deslocada à direita da média. Tal comportamento pode interferir em classificadores baseados em probabilidade, tais como Naive Bayes.

Os nomes das classes , bem como o quantitativo, porcentagens e índices de desbalanceamento (I_D) da base de dados CIC2017 subamostrada são encontrados na Tabela 23 do Apêndice A. Observa-se que as amostras das classes majoritárias foram reduzidas, diminuindo assim o índice de desbalanceamento I_D

Uma análise multivariada por meio da técnica *corrplot* consta na Figura 12, na qual é possível visualizar a matriz correlação entre as características no formato de gráfico de pizza, em que quanto mais preenchido o círculo, mais próximo de 1 em módulo é a correlação. Os tons de vermelho e azul indicam correlação negativa e positiva, respectivamente. Assim, percebe-se que algumas características são fortemente correlacionadas positivamente, o que pode em alguns casos implicar no desempenho de alguns algoritmos de aprendizagem de máquina devido a semelhança entre tais características.

Figura 12 – Correlação entre as características CICIDS2017.



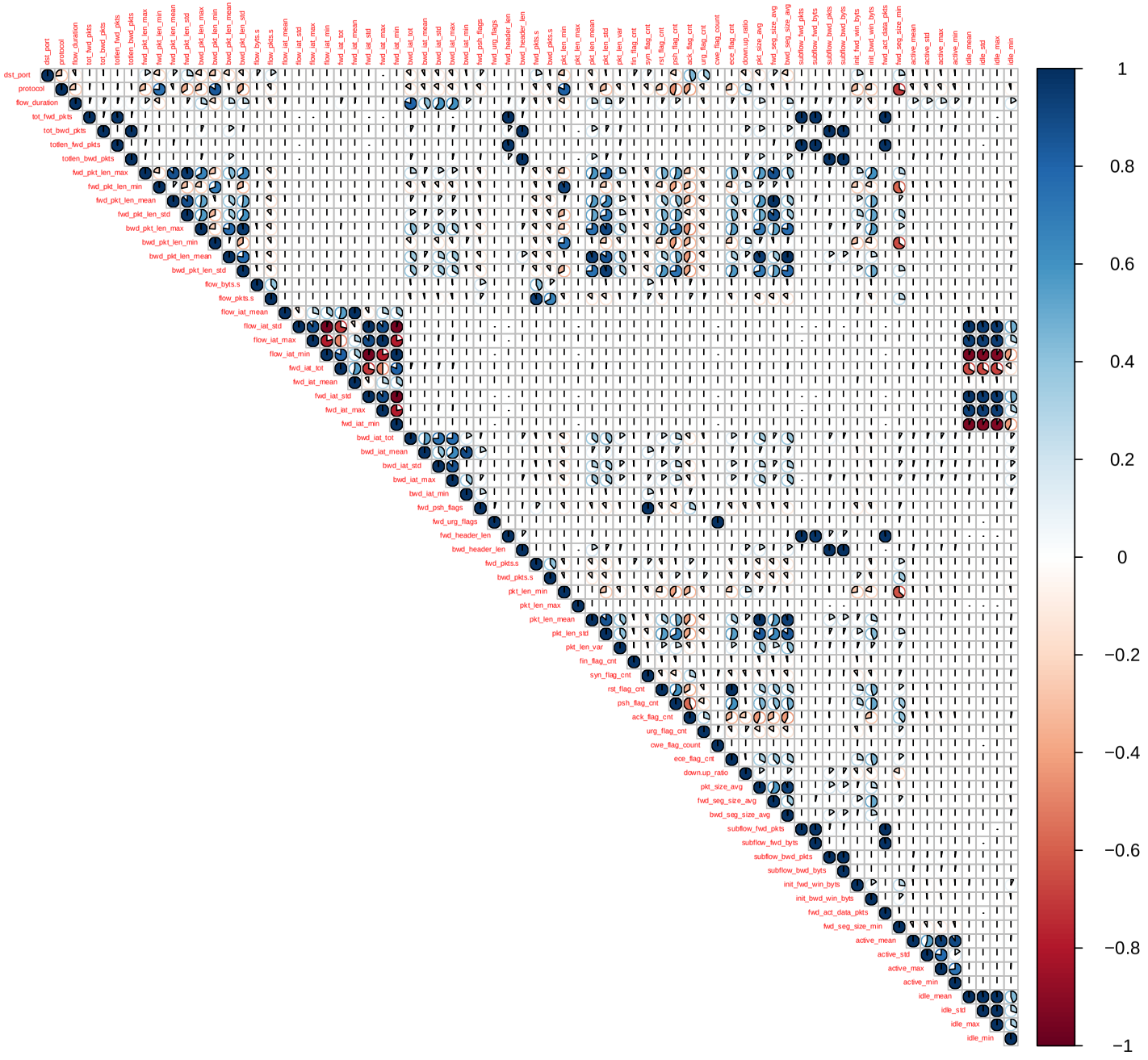
Fonte: elaborado pelo autor .

CICIDS2018

Na Tabela 22 no apêndice A são mostrados a média, desvio padrão, mínimo e máximo de cada característica da base CICIDS2018. Observa-se na tabela que as mesmas características com média e desvio padrão zero mencionadas na seção anterior possuem comportamento similar para a base de dados CICIDS2018, evidenciando a necessidade da remoção de tais preditores. Quanto à obliquidade, observa-se que as características em sua maioria também possuem assimetria deslocada para a direita, podendo ser possível inferir que para estas duas bases de dados, os fluxos de rede (ataques e normais) possuem este comportamento. A análise multivariada também foi realizada por meio do *corplot* na Figura 13. Observa-se que algumas características possuem correlação fortemente positiva, tais como tot_fwd_pkts e fwd_header_len e outras possuem correlação fortemente negativa, tais como fwd_iat_std e fwd_iat_min, fwd_iat_min e idle_mean e flow_iat_std e fwd_iat_min.

Os nomes das classes , bem como o quantitativo, porcentagens e índices de desbalanceamento (I_D) da base de dados CIC2018 subamostrada são encontrados na Tabela 24 do Apêndice A. Observa-se que a maioria das classes teve seu tamanho reduzido, diminuindo assim o índice de desbalanceamento I_D. Salienta-se que o uso da subamostragem nas bases de dados visa manter a maior quantidade possível de amostras nas classes minoritárias, consequentemente preservando o número de registros nas classes minoritárias.

Figura 13 – Correlação entre as características CICIDS2018.



Fonte: elaborado pelo autor .

5.2 Resultados obtidos para a Base CIC2017

Os resultados dos experimentos descritos na metodologia quanto à base de dados CIC2017 são mostrados nas Tabelas de 7 a 10, sendo que a Tabela 5 se refere a base completa e as demais tabelas às sub-bases oriundas da base completa.

Tabela 5 – Resultados obtidos com Base de Dados CIC2017 completa para os classificadores avaliados

Algoritmo	Acurácia		Precision		Recall		F1		Tempo (seg.)	
	média/desvio		média/desvio		média/desvio		média/desvio		média/desvio	
NC	0,555	1,30E-03	0,857	5,19E-04	0,555	8,23E-04	0,641	1,30E-03	4,559	3,33E-02
NB	0,448	7,64E-02	0,968	1,28E-03	0,448	7,64E-02	0,572	8,46E-02	5,154	6,10E-02
KNN	**0,988**	**5,56E-04**	**0,988**	**5,55E-04**	**0,988**	**5,56E-04**	**0,988**	**5,54E-04**	**1625,54**	**8,30E+01**
RF	0,886*	1,87E-04*	0,829*	3,13E-04*	0,886*	1,87E-04*	0,850*	2,56E-04*	64,03*	1,60E-01*

Fonte: o autor.

Os resultados dos algoritmos avaliados na base CIC2017, completa, ou seja, sem a aplicação das técnicas de subamostragem, encontram-se dispostos na Tabela 5. Observa-se que o algorítimo KNN possui as melhores métricas de assertividade, entretanto, o tempo de treinamento é alto em relação aos demais.

As sub-bases geradas produzem resultados que são mostrados a seguir. Destaca-se que o classificador SVM não é avaliado na base de dados devido ao seu custo computacional ser proporcional ao número de registros, quando comparado aos demais algoritmos.

Modelos fortemente baseados em distância tais como Nearest centroid (NC) e KNN apresentam comportamentos distintos, cuja diferença é de até 55% quanto as métricas, podendo indicar que as classes não são representadas pelos centroides. Desta forma, a geometria dos dados nas classes é esparsa, dificultando a representatividade de cada classe pelo seu centroide.

Quanto a acurácia e Recall do algoritmo Naive Bayes, os valores alcançados justificam-se pela obliquidade na base de dados completa, uma vez que o classificador utiliza a probabilidade como base para reconhecimento do modelo e predição.

Já o desempenho do classificador Florestas aleatórias é intermediária, uma vez que este classificador é condicionado à quantidade de amostras de treinamento.

Salienta-se a relação entre tempo de treinamento do modelo e assertividade nos classificadores avaliados. Neste cenário, o classificador mais adequado seria o de Florestas aleatórias, que possui uma assertividade menor do que o KNN, mas em menor tempo.

Resultados obtidos para a sub-base aleatória

Os resultados dos experimentos utilizando a abordagem de subamostragem aleatória, utilizando a metodologia descrita no Capítulo 4 são mostrados na Tabela 6. O classificador NC apresenta um valor menor de assertividade, indicando que a sub-base gerada ainda possui os dados esparsos, dificultando a representatividade com base nos centroides de cada classe. Entretanto, o algoritmo NB apresentou melhores métricas de assertividade com relação ao cenário anterior e foi possível avaliar o classificador SVM nessa abordagem. O algoritmo Florestas aleatórias obteve menor assertividade com relação ao cenário anterior devido ao menor número de amostras na base subamostrada.

Tabela 6 – Resultados obtidos para a sub-base aleatória

Algoritmo	Acurácia média/desvio		Precision média/desvio		Recall média/desvio		F1 média/desvio		Tempo (seg.) média/desvio	
NC	0,577	1,08E-02	0,648	5,29E-03	0,577	1,08E-02	0,563	1,51E-02	0,367	4,59E-03
NB	0,846	5,08E-03	0,946	4,18E-03	0,846	5,08E-03	0,882	4,40E-03	0,420	5,03E-03
KNN	**0,987**	**3,26E-04**	**0,987**	**3,17E-04**	**0,987**	**3,26E-04**	**0,987**	**3,25E-04**	**3,210**	**2,66E-01**
RF	0,703	6,13E-03	0,553	2,28E-03	0,703	6,13E-03	0,614	5,08E-03	3,046	2,73E-02
SVM	0,880	1,82E-02	0,863	1,77E-02	0,880	1,82E-02	0,867	1,82E-02	198,761	1,69E+01

Fonte: o autor.

Resultados obtidos para a sub-base Cluster centroides

As métricas dos mesmos algoritmos avaliados anteriormente, diferindo na técnica de subamostragem (Cluster centroides) são apresentados na Tabela 7. Os classificadores avaliados apresentaram maior assertividade em relação aos cenários anteriores, indicando que a sub-base produzida possui melhor representatividade, Vale salientar que os algorítimos NC, NB e SVM apresentaram aumento de até 26%, 52% e 5% respectivamente. Outro fato importante é que algoritmos baseados em distância foram beneficiados na análise em termos das métricas avaliadas.

Tabela 7 – Resultados obtidos para a sub-base Cluster centroides

Algoritmo	Acurácia média/desvio		Precision média/desvio		Recall média/desvio		F1 média/desvio		Tempo (seg.) média/desvio	
NC	0,810	8,47E-03	0,858	8,78E-03	0,810	8,47E-03	0,827	8,36E-03	0,288	5,14E-03
NB	0,961*	5,83E-04*	0,971*	7,41E-04*	0,961*	5,83E-04*	0,964*	6,54E-04*	0,317*	2,43E-03*
KNN	**0,994**	**1,09E-04**	**0,994**	**1,23E-04**	**0,994**	**1,09E-04**	**0,994**	**1,09E-04**	**2,855**	**1,20E+00**
RF	0,762	2,37E-03	0,604	1,53E-02	0,762	2,37E-03	0,670	4,96E-03	3,237	1,36E-02
SVM	0,938	4,81E-03	0,933	4,68E-03	0,938	4,81E-03	0,928	5,82E-03	92,604	1,45E+00

Fonte: o autor.

Resultados obtidos para a sub-base NearMiss1

Os resultados dos experimentos utilizando a abordagem de subamostragem por NearMiss1 são mostrados na Tabela 8. O desempenho em geral dos classificadores é superior do que no cenário de subamostragem aleatória e inferior com relação ao cenário de subamostragem por Cluster centroides exceto para os classificadores KNN e SVM, os quais possuem métricas de assertividade e tempo semelhantes.

Tabela 8 – Resultados obtidos para a sub-base NearMiss1

Algoritmo	Acurácia média/desvio		Precision média/desvio		Recall média/desvio		F1 média/desvio		Tempo (seg.) média/desvio	
NC	0,71	9,42E-03	0,78	1,62E-02	0,718	9,42E-03	0,70	1,27E-02	0,28	3,71E-03
NB	0,89	2,01E-03	0,95	1,83E-03	0,896	2,01E-03	0,91	2,19E-03	0,32	1,87E-02
KNN	**0,99**	**1,50E-04**	**0,99**	**1,65E-04**	**0,99**	**1,50E-04**	**0,99**	**1,47E-04**	**3,41**	**9,04E-01**
RF	0,78	3,59E-03	0,65	1,47E-02	0,78	3,59E-03	0,70	7,05E-03	3,07	6,43E-03
SVM	0,93	1,45E-03	0,93	1,19E-03	0,93	1,45E-03	0,92	1,31E-03	77,04	1,25E+00

Fonte: o autor.

Comparação entre classificadores sob diferentes subamostragens

Nesta seção a comparação entre os classificadores é realizada. Os resultados dos experimentos são apresentados adotando *boxplots* em pares para fins comparativos para uma melhor visualização das métricas resultantes. Foram escolhidas duas métricas para comparação: acurácia, que representa uma visão geral do classificador e F1, a qual é a média harmônica entre Recall e Precision.

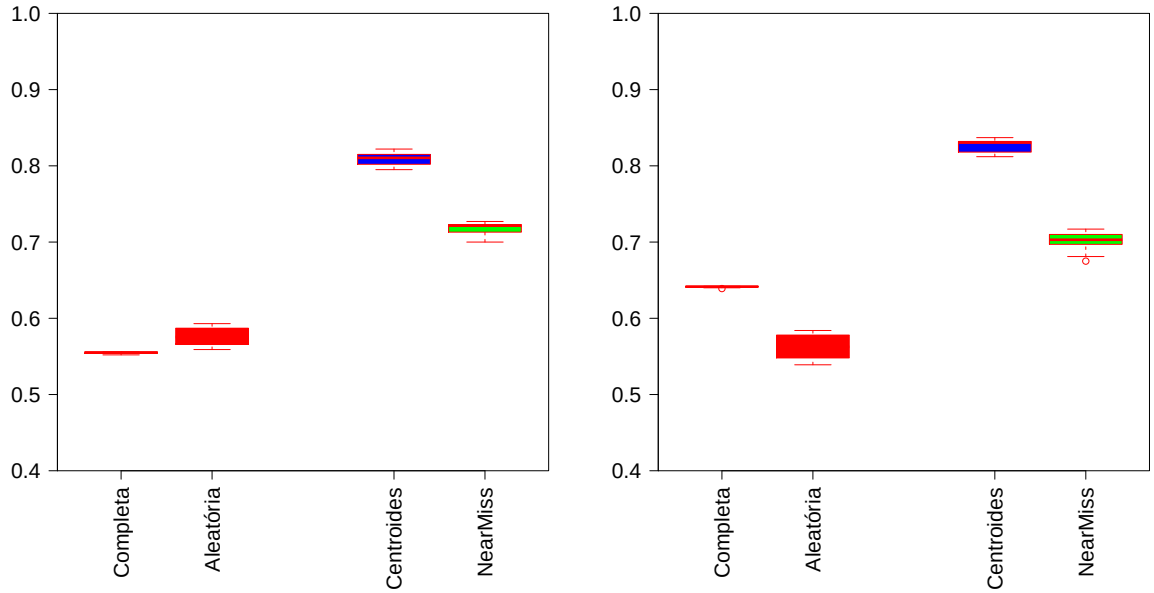
A comparação entre as bases subamostradas e a completa para o classificador NC em termos de acurácia é mostrada na Figura 14a. Nota-se que a subamostragem aleatória possui maior variação em termos dessa métrica, uma vez que para a geração dessa sub-base, 10 (dez) repetições foram utilizadas e os classificadores foram avaliados em cada sub-base. Assim, diferentes amostras sorteadas, neste caso, produzem resultados mais distantes da média, uma vez que esta técnica, por natureza, não busca representatividade na subamostragem. A subamostragem por Cluster centroides apresentou a melhor métrica para este classificador, seguida daquela por NearMiss1, indicando que estas técnicas de subamostragem aplicadas deram maior representatividade à base de dados. Observa-se que a subamostragem aleatória possui resultados semelhantes à base de dados completa para este classificador, indicando que a sub-base neste caso não é bem representada pelo centroide.

Quanto a métrica de F1, o classificador NC foi comparado na Figura 14b. A

subamostragem por Cluster centroides é a mais adequada para este classificador, o qual possui F1 acima de 80%, enquanto a subamostragem aleatória apresenta os piores valores de F1, visto que o classificador produziu maior taxa de falsos positivos do que na base completa, representada pela métrica de Precision encontrada na Tabela 6.

Figura 14 – Comparação entre as métricas de diferentes subamostragens para o classificador NC.

(a) Acurácia. (b) F1.



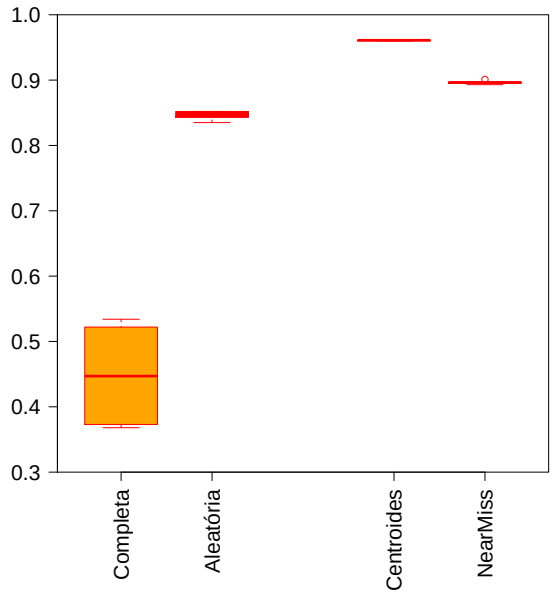
Fonte: Elaborado pelo autor.

As diferentes técnicas de subamostragem, bem como a base de dados completa para o classificador Naive Bayes estão mostradas na Figura 15a. O desempenho do classificador na base de dados completa é bem abaixo que nas sub-bases, além de produzir a maior variação em torno da média. Esse fato ocorre devido aos fatores de obliquidade e desbalanceamento da base de dados, uma vez que o classificador em questão é baseado em probabilidade e uma grande quantidade de elementos de uma classe majoritária faz com que o modelo seja impreciso em sua predição. Ao encontro disso, a abordagem subamostrada tende a mitigar este efeito, uma vez que a subamostragem ocorre nas classes majoritárias, fazendo com que o modelo calcule melhor as probabilidades e consequentemente seja mais assertivo. As abordagens por Cluster centroides e NearMiss1 se sobressaem com relação a aleatória devido a seleção das amostras nas subamostragens ser mais criteriosa que na última. O mesmo comportamento ocorre na Figura

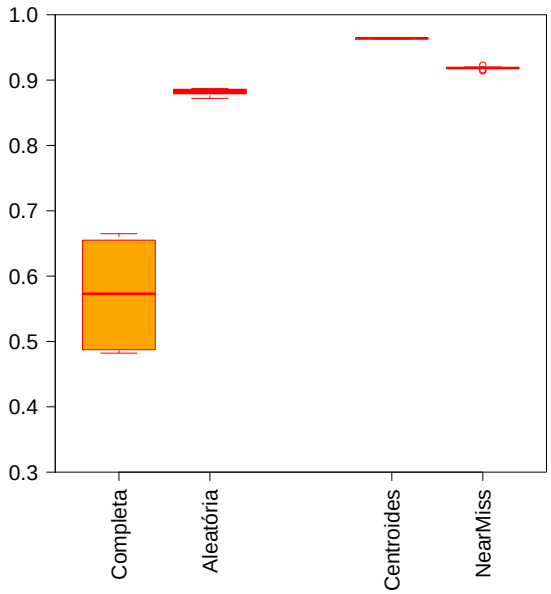
15b, na qual o classificador não lida bem com a grande massa de dados da base completa e a solução por subamostragem apresenta diferenças significativas quanto a acurácia e Recall.

Figura 15 – Comparação entre as métricas de diferentes subamostragens para o classificador NB.

(a) Acurácia.



(b) F1.



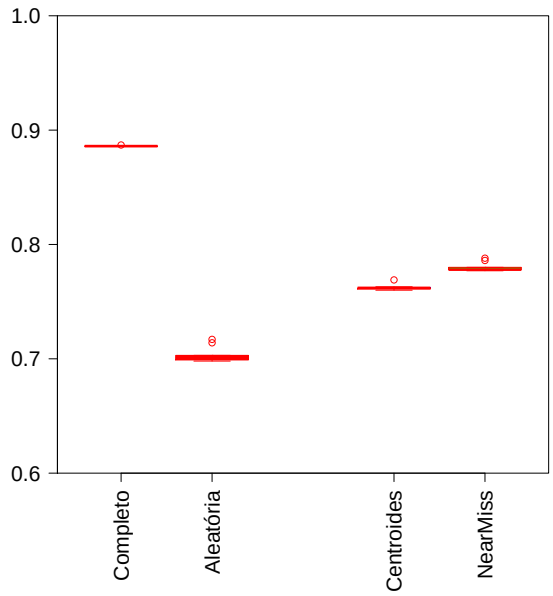
Fonte: Elaborado pelo autor.

A acurácia do classificador Florestas aleatórias sob as diferentes técnicas de subamostragem, além da base CIC2017 completa é mostrada na Figura 16a. O desempenho do classificador na base de dados completa é superior às demais devido à natureza do algoritmo de classificação, visto que para uma melhor classificação, mais amostras são necessárias para a construção do modelo. Ao realizar a comparação entre subamostragens, tem-se que o classificador RF avaliado em subamostragem aleatória possui a menor acurácia média e quanto as sub-bases por Cluster centroides e NearMiss1, o algoritmo atingiu assertividades semelhantes. No caso da F1, ilustrado na Figura 16b o classificador RF manteve o mesmo comportamento, devido a alta taxa de falsos positivos encontrada nas sub-bases.

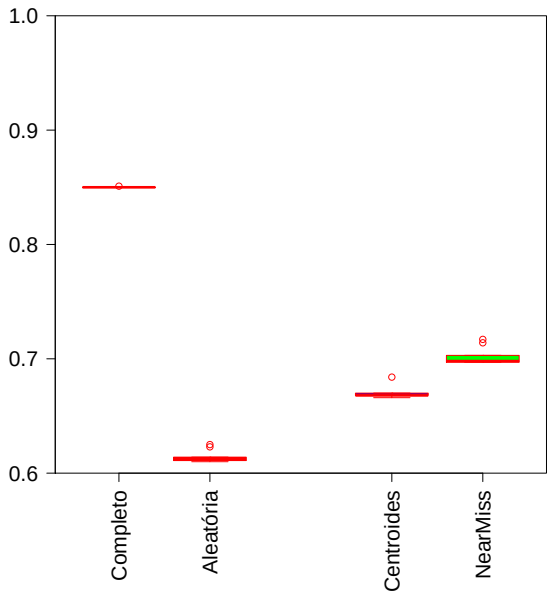
A acurácia do classificador KNN com relação as sub-bases e a base CIC2017 original é exibida na Figura 17a, na qual apresenta acurácias semelhantes nos diferentes cenários. Entretanto, como argumentado anteriormente, o tempo para reconhecimento é bem maior no caso da base de dados completa, fato que evidencia a importância da geração das bases subamostradas.

Figura 16 – Comparação entre as métricas de diferentes subamostragens para o classificador RF.

(a) Acurácia.



(b) F1.



Fonte: Elaborado pelo autor.

O mesmo comportamento ocorre quanto à F1, ilustrada na Figura 17b.

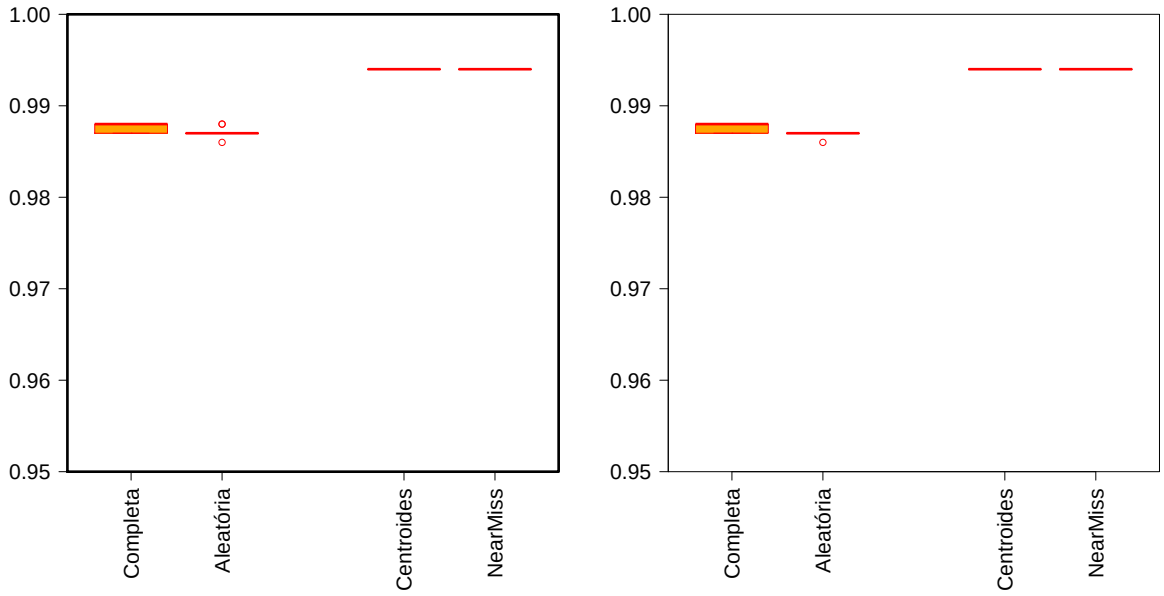
A comparação entre a acurácia do classificador SVM com relação as bases subamostradas é mostrada na Figura 18a. As subamostragens por Cluster centroides e NearMiss1 apresentam resultados acima de 90% de acurácia para o SVM, enquanto na sub-base aleatória, a acurácia é menor. O mesmo comportamento ocorre quanto à F1, ilustrada na Figura 18b.

Comparação entre classificadores por meio do teste de Wilcoxon

A comparação entre classificadores por meio do teste de Wilcoxon estatístico visa avaliar se há diferenças estatísticas entre os pares de métricas um mesmo classificador em cada subamostragem. Os valores de T e z, representam o valor mínimo da soma dos postos utilizados e o valor do teste, respectivamente. Caso o valor T seja menor que o T crítico de acordo com o nível de significância escolhido e o valor absoluto de z for maior que o valor crítico tabelado, a hipótese nula pode ser desconsiderada e consequentemente a hipótese alternativa torna-se válida. Neste último caso, as amostras são estatisticamente diferentes e pode-se inferir que as métricas de um dos classificadores possuem medianas maiores que as do outro. A Tabela 9 ilustra os valores encontrados no teste estatístico para os classificadores avaliados nas sub-bases aleatória e por

Figura 17 – Comparação entre as métricas de diferentes subamostragens para o classificador KNN.

(a) Acurácia. (b) F1.



Fonte: Elaborado pelo autor.

Cluster centroides, bem como para as sub-bases aleatória e por NearMiss1. Nestas comparações, os valores de T e z absolutos encontrados são, respectivamente, menores e maiores para todos os classificadores em todas as métricas com significância de 99%, evidenciando estatisticamente a viabilidade do uso das subamostragens por Cluster centroides e NearMiss1 comparadas com a abordagem aleatória.

Tabela 9 – Comparação entre subamostragens Aleatória e por Cluster centroides | subamostragens Aleatória e por NearMiss1

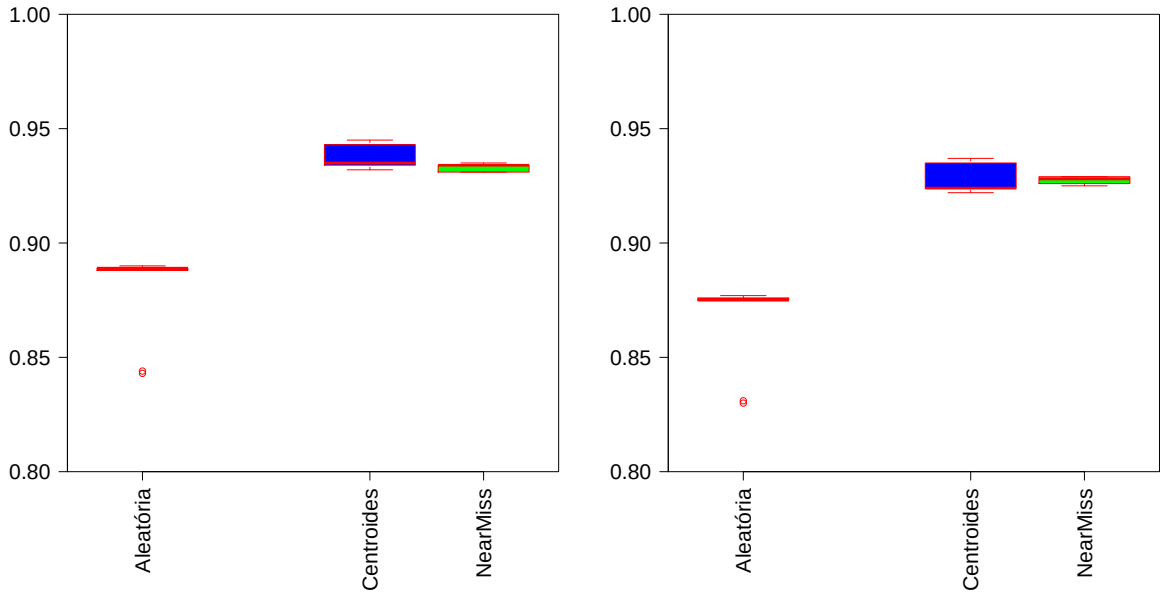
Valores de T e |z| para n = 10, T crítico = 5, z crít. 2,33 e signif. de 0,01								
Algoritmo	Acurácia T / |z|		Precision T / |z|		Recall T / |z|		F1 T / |z|	
NC	0	2,8	0	2,8	0	2,8	0	2,8
NB	0	2,8	0	2,8	0	2,8	0	2,8
KNN	0	2,8	0	2,8	0	2,8	0	2,8
RF	0	2,8	0	2,8	0	2,8	0	2,8
SVM	0	2,8	0	2,8	0	2,8	0	2,8

Fonte: o autor.

Os resultados para o teste de Wilcoxon na comparação entre as subamostragens por

Figura 18 – Comparação entre as métricas de diferentes subamostragens para o classificador SVM.

(a) Acurácia. (b) F1.



Fonte: Elaborado pelo autor.

NearMiss1 e Cluster centroides são ilustrados na Tabela 10. Por meio do teste é possível verificar que as amostras de métricas para a sub-base por Cluster centroides são diferentes estatisticamente que as provenientes da sub-base NearMiss1, uma vez que os valores de T e z correspondem com a rejeição da hipótese nula e adoção da hipótese alternativa, a qual indica que a mediana das métricas dos classificadores NC, NB são maiores que as da outra subamostragem. Entretanto para o classificador KNN, não foram encontradas diferenças numéricas ou estatísticas para o número de casas decimais iguais a três, consequentemente não sendo possível rejeitar a hipótese nula. Além disso, as métricas de Precision e F1 no classificador SVM são estatisticamente iguais para ambas sub-bases.

Como visto anteriormente, o classificador RF avaliado na sub-base CIC2017 completa tem suas métricas maiores que ao ser avaliado nas sub-bases. As comparações entre a base completa e sub-bases por Cluster centroides, aleatória e por NearMiss1 são mostradas nas Tabelas 11, 12 e 13. De fato, o teste confirma que o classificador avaliado na base completa possui métricas estatisticamente superiores em mediana do que ao ser avaliado em sub-bases. O teste estatístico confirma que as métricas nos classificadores NC,NB e KNN são superiores em

Tabela 10 – Comparação entre Cluster centroides e NearMiss1 por meio do teste de Wilcoxon

Valores de T e |z| para n = 10, T crítico = 5, z crít. 2,33 e signif. de 0,01								
Algoritmo	Acurácia T / |z|		Precision T / |z|		Recall T / |z|		F1 T / |z|
NC	0	2,8	0	2,8	0	2,8	0	2,8
NB	0	2,8	0	2,8	0	2,8	0	2,8
RF	0	2,8	0	2,8	0	2,8	0	2,8
SVM	0	2,5	10	1,78	0	2,5	21	0,6625

Fonte: o autor.

um cenário de subamostragem por Cluster centroides com relação à base completa. O mesmo comportamento ocorre para os classificadores NC e NB em um cenário de subamostragem aleatória. Quanto ao classificador KNN, não foi possível realizar o teste estatístico devido a diferença entre as amostras das métricas ser zero para Precision de três casas decimais, fato que inviabiliza o cálculo do teste, consequentemente a hipótese nula deve ser considerada nesse caso. A Precision do classificador Naive Bayes para a base completa é estatisticamente maior que no cenário de subamostragem por NearMiss1, entretanto, ambas abordagens possuem valores aproximados, cuja diferença é de apenas 1%.

Tabela 11 – Comparação entre a base completa e subamostragem por Cluster centroides

Valores de T e |z| para n = 10, T crítico = 5, z crít. 2,33 e signif. de 0,01								
Algoritmo	Acurácia T / |z|		Precision T / |z|		Recall T / |z|		F1 T / |z|
NC	0	2,8	0	2,8	0	2,8	0	2,8
NB	0	2,8	0	2,8	0	2,8	0	2,8
KNN	0	2,8	0	2,8	0	2,8	0	2,8
RF	0	2,8	0	2,8	0	2,8	0	2,8

Fonte: o autor.

Tabela 12 – Comparação entre a base completa e subamostragem por Cluster centroides

Valores de T e |z| para n = 10, T crítico = 5, z crít. 2,33 e signif. de 0,01								
Algoritmo	Acurácia T / |z|		Precision T / |z|		Recall T / |z|		F1 T / |z|
NC	0	2,8	0	2,8	0	2,8	0	2,8
NB	0	2,8	0	2,8	0	2,8	0	2,8
RF	0	2,8	0	2,8	0	2,8	0	2,8

Fonte: o autor.

Tabela 13 – Comparação entre a base completa e subamostragem por NearMiss1

Valores de T e |z| para n = 10, T crítico = 5, z crít. 2,33 e signif. de 0,01								
Algoritmo	Acurácia T / |z|		Precision T / |z|		Recall T / |z|		F1 T / |z|	
NC	0	2,8	0	2,8	0	2,8	0	2,8
NB	0	2,8	0	2,8	0	2,8	0	2,8
KNN	0	2,8	0	2,8	0	2,8	0	2,8
RF	0	2,8	0	2,8	0	2,8	0	2,8

Fonte: o autor.

5.3 Resultados obtidos para a Base CIC2018

Os resultados dos experimentos na base de dados CIC2018 completa são mostrados na Tabela 14, em que o classificador base empregado nos experimentos, Nearest centroid, apresenta baixos valores de acurácia/Recall, indicando que as classes não são representadas pelos centroides e a geometria dos dados é esparsa, dificultando a predição deste classificador. Além disso, outro fato que colabora com o esse comportamento é que a base de dados é 5 vezes maior que a CIC2017 e o desbalanceamento é mais evidente na CIC2018. Nesse sentido, as métricas para o classificador Naive Bayes são superiores ao anterior. Isto ocorre devido ao alto grau de desbalanceamento da base CIC2018, a qual possui uma grande quantidade de registros de tráfego normal. Assim, o classificador Naive Bayes tem nesse caso os pesos mais significativos com relação a esta classe e na porção de teste, as amostras possivelmente ocorrem mais vezes devido à grande quantidade de registros. O desempenho do classificador Florestas aleatórias é mediana, uma vez que este classificador é condicionado à quantidade de amostras de treinamento. Neste cenário, diferentemente do anterior, o classificador mais adequado é o Naive Bayes, que possui aceitável assertividade em menor tempo. Além disso, os classificadores SVM e KNN não foram avaliados nesse cenário devido à elevada complexidade computacional, uma vez que tal base é cinco vezes maior que a CIC2017.

Tabela 14 – Resultados obtidos com Base de Dados CIC2018 completa para os classificadores avaliados

Algoritmo	Acurácia média/desvio		Precision média/desvio		Recall média/desvio		F1 média/desvio		Tempo (seg.) média/desvio	
NC	0,26	7,74E-03	0,90	6,20E-03	0,26	7,74E-03	0,35	6,72E-03	3,75	1,35E+00
NB	**0,80**	**8,54E-05**	**0,86**	**1,68E-04**	**0,80**	**8,54E-05**	**0,81**	**1,30E-04**	**5,86**	**5,61E-01**
RF	0,83	1,33E-04	0,69	2,21E-04	0,83	1,33E-04	0,75	1,87E-04	410,86	7,21E+01

Fonte: o autor.

Resultados obtidos para a sub-base aleatória

Os resultados dos experimentos utilizando a abordagem de subamostragem aleatória são mostrados na Tabela 15. O classificador NC apresenta baixa assertividade, apesar de haver um aumento de até 46% em suas métricas, indicando que a sub-base gerada ainda possui os dados esparsos, dificultando a representatividade com base nos centroides de cada classe. Já o algoritmo NB apresenta menores resultados em termos das métricas avaliadas devido ao fato que esta subamostragem revela um comportamento na base completa e desbalanceada: as métricas são ponderadas pela quantidade de elementos da classe. Assim o classificador que é assertivo para uma determinada classe majoritária, obtém resultados aceitáveis em termos de assertividade. No caso a subamostragem realizada na base CIC2018 completa reduziu bastante o índice de desbalanceamento, sendo possível inferir a partir dos resultados que as alterações nas métricas dos classificadores são provenientes também do fator de desbalanceamento reduzido. O algoritmo Florestas aleatórias obteve menor assertividade com relação ao cenário anterior, semelhante ao ocorrido na base CIC2017 e os classificadores KNN e SVM obtiveram resultados semelhantes em termos de suas métricas, cuja diferença é de apenas 2%.

Tabela 15 – Resultados obtidos para a sub-base aleatória

Algoritmo	Acurácia média/desvio		Precision média/desvio		Recall média/desvio		F1 média/desvio		Tempo (seg.) média/desvio	
NC	0,47	5,64E-03	0,54	2,23E-02	0,479	5,64E-03	0,454	6,58E-03	0,39	3,52E-02
NB	0,76	1,52E-03	0,85	5,08E-03	0,76	1,52E-03	0,76	4,35E-03	0,39	3,15E-02
KNN	**0,85**	**2,35E-03**	**0,83**	**8,40E-03**	**0,85**	**2,35E-03**	**0,82**	**5,75E-03**	**6,87**	**1,20E+00**
RF	0,74	4,15E-03	0,73	1,02E-02	0,74	4,15E-03	0,68	6,44E-03	2,58	8,15E-02
SVM	0,81	6,25E-04	0,84	7,55E-04	0,81	6,25E-04	0,79	8,13E-04	151,50	1,44E+00

Fonte: o autor.

Resultados obtidos para a sub-base Cluster centroides

As métricas dos mesmos algoritmos avaliados anteriormente, diferindo na técnica de subamostragem (Cluster centroides) são apresentados na Tabela 16. Os classificadores avaliados apresentam maior assertividade em relação aos cenários anteriores, indicando que a sub-base produzida possui melhor representatividade. Vale salientar que os algorítimos NC, NB e SVM apresentaram aumento de até 62%, 12% e 16% respectivamente com relação aos demais cenários apresentados anteriormente. Outro fato importante é que algoritmos baseados em distância foram beneficiados na análise em termos das métricas avaliadas: NC, KNN e SVM, fato que ocorre na base de dados discutida anteriormente.

Tabela 16 – Resultados obtidos para a sub-base Cluster centroides

Algoritmo	Acurácia média/desvio		Precision média/desvio		Recall média/desvio		F1 média/desvio		Tempo (seg.) média/desvio	
NC	0,884	3,71E-03	0,891	4,42E-03	0,884	3,71E-03	0,878	3,00E-03	0,344	2,41E-02
NB	0,98*	5,37E-03*	0,98*	4,44E-03*	0,98*	5,37E-03*	0,98*	5,08E-03*	0,36*	2,41E-03*
KNN	**0,99**	**1,67E-04**	**0,99**	**1,65E-04**	**0,99**	**1,67E-04**	**0,99**	**1,66E-04**	**6,70**	**3,26E+00**
RF	0,850	5,11E-03	0,828	2,06E-02	0,850	5,11E-03	0,806	8,02E-03	2,873	2,97E-01
SVM	0,971	6,95E-04	0,971	7,77E-04	0,971	6,95E-04	0,969	7,61E-04	68,988	7,10E+00

Fonte: o autor.

Resultados obtidos para a sub-base NearMiss1

Os resultados dos experimentos utilizando a abordagem de subamostragem por NearMiss1 são mostrados na Tabela 17. O desempenho em geral dos classificadores é superior do que no cenário de subamostragem aleatória e inferior com relação ao cenário de subamostragem por Cluster centroides, indicando que a técnica de subamostragem por Cluster centroides é mais representativa com relação as demais utilizadas, facilitando os classificadores a reconhecerem os diferentes tipos de tráfego.

Tabela 17 – Resultados obtidos para a sub-base NearMiss1

Algoritmo	Acurácia média/desvio		Precision média/desvio		Recall média/desvio		F1 média/desvio		Tempo (seg.) média/desvio	
NC	0,867	8,13E-04	0,889	6,44E-04	0,867	8,13E-04	0,868	8,12E-04	0,376	6,19E-03
NB	0,91*	3,86E-03*	0,92*	3,23E-03*	0,91*	3,86E-03*	0,90*	3,90E-03*	0,444*	2,35E-02*
KNN	**0,92**	**1,64E-02**	**0,92**	**3,32E-02**	**0,92**	**1,64E-02**	**0,90**	**2,95E-02**	**9,17**	**3,73E+00**
RF	0,843	8,24E-03	0,855	8,91E-03	0,843	8,24E-03	0,813	8,17E-03	1,839	5,77E-02
SVM	0,848	6,66E-04	0,817	7,77E-04	0,848	6,66E-04	0,818	7,88E-04	73,241	6,85E-01

Fonte: o autor.

Comparação entre classificadores sob diferentes subamostragens

Nesta seção a comparação entre os classificadores é realizada por meio do *boxplot* dos principais resultados referentes às métricas de avaliação empregadas nos experimentos. Foram escolhidas duas métricas para comparação: acurácia, que representa uma visão geral do classificador e F1, a qual é a média harmônica entre Recall e Precision.

A comparação entre as bases subamostradas e a completa para o classificador NC em termos de acurácia é mostrada na Figura 19a. Conforme abordado anteriormente, a abordagem com a base de dados completa possui métricas muito baixas relacionadas às abordagens por subamostragem. Além disso, o tempo de reconhecimento do modelo é maior, haja vista que as bases subamostradas possuem 1% do tamanho da base original. A subamostragem por Cluster centroides apresentou a melhor métrica para este classificador, seguida daquela por NearMiss1, indicando que estas técnicas de subamostragem aplicadas deram maior representatividade à base

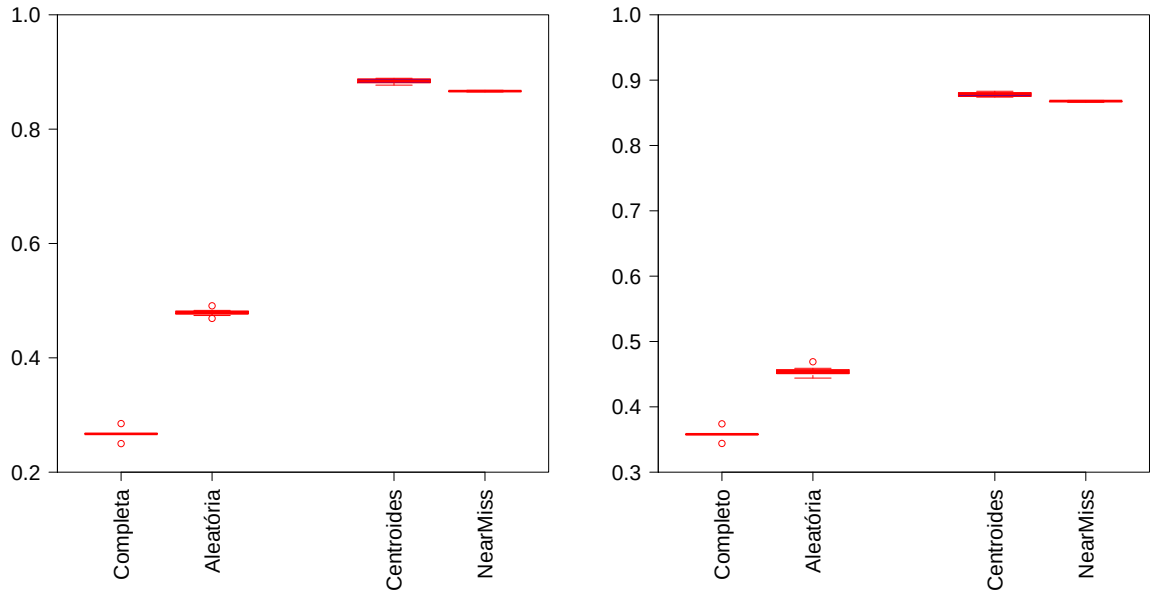
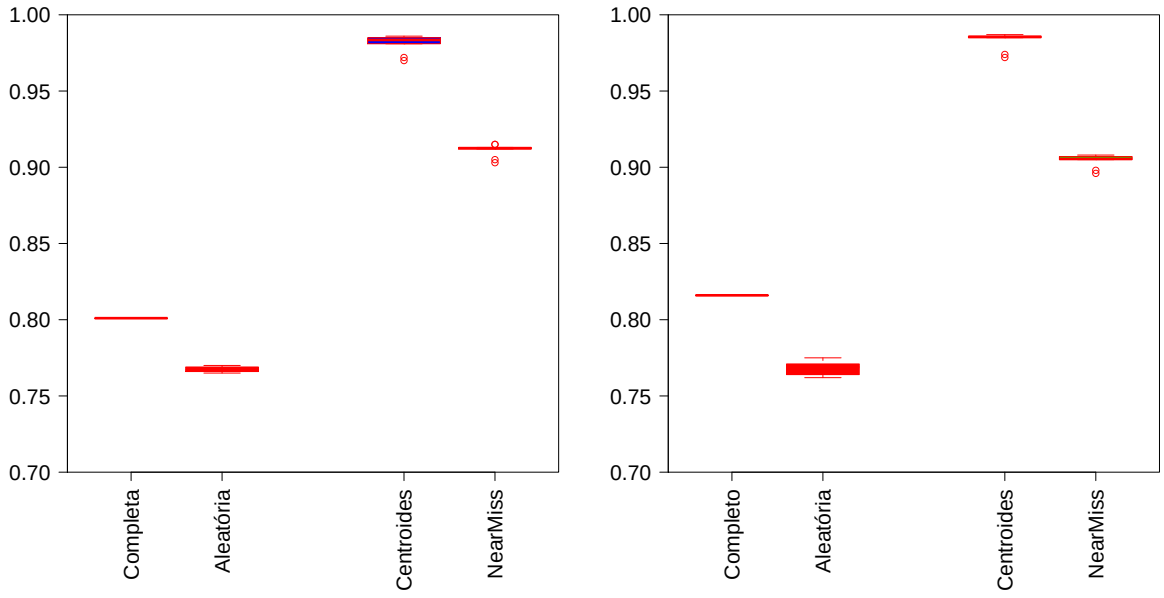


Figura 19 – Comparação entre métricas de diferentes subamostragens para o classificador NC: a) Acurácia e b) F1

de dados. Entretanto, a subamostragem aleatória tem comportamento distante das demais por seu teor randômico. Quanto a métrica de F1, o classificador NC foi comparado na Figura 19b. A subamostragem por Cluster centroides é a mais adequada para este classificador, o qual possui F1 acima de 85%, enquanto a abordagem completa apresenta os piores valores de F1, visto que o classificador produziu maior taxa de falsos positivos do que nos demais cenários, representada pela métrica de Precision encontrada na Tabela 15.

As diferentes técnicas de subamostragem, bem como a base de dados completa para o classificador Naive Bayes estão mostradas na Figura 20a. O desempenho do classificador na sub-base aleatória é menor que nos demais cenários comparados. Este fato ocorre pelos diferentes índices de balanceamento na base completa e nas sub-bases, fazendo com que o classificador gere grande quantidade de falsos positivos. Desta forma, como ocorreu na base CIC2017, as abordagens por Cluster centroides e NearMiss1 se sobressaem com relação a aleatória devido a seleção das amostras nas subamostragens ser mais criteriosa que na última. O mesmo comportamento ocorre na Figura 20b, na qual o classificador não lida bem com a grande massa de dados da base completa e a solução por subamostragem apresenta diferenças significativas quanto a acurácia/Recall.

A acurácia do classificador Florestas aleatórias sob as diferentes técnicas de subamostragem, além da base CIC2018 completa é mostrada na Figura 21a. A performance do classificador na base de dados completa é superior à aleatória devido a dois motivos: o classi-



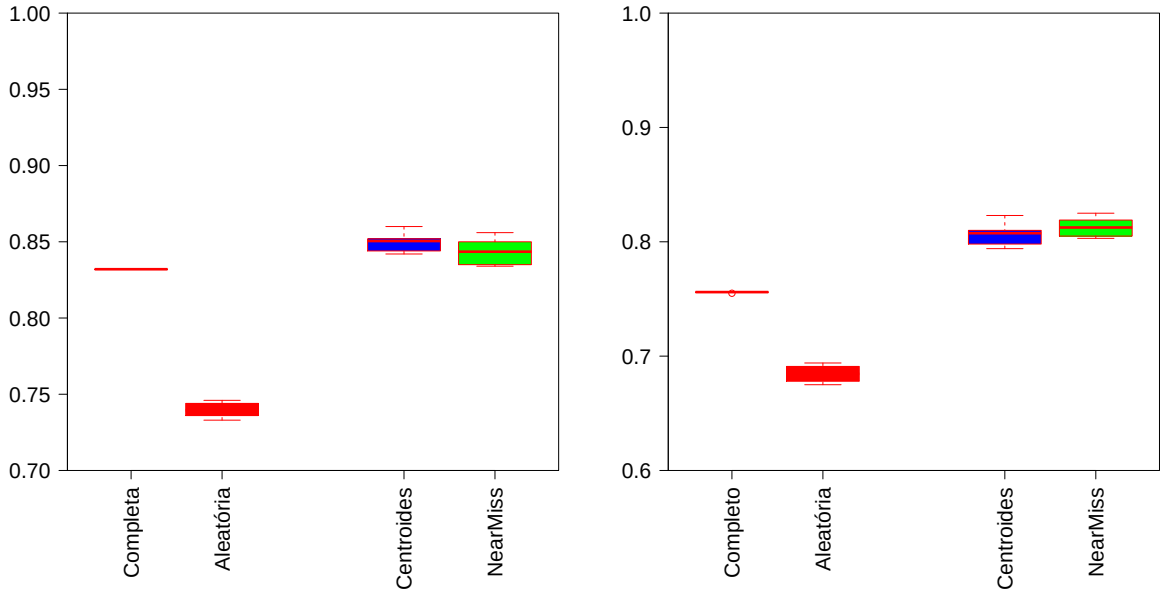
(a) (b)

Figura 20 – Comparação entre métricas de diferentes subamostragens para o classificador NB: a) Acurácia e b) F1

ficador ao lidar com dados desbalanceados favorece a classe majoritária para a predição e nos testes a classe majoritária aparece mais vezes na avaliação do algoritmo e a sub-base de dados aleatória não possui critérios para seleção das amostras, dificultando na representatividade da base completa. Ao realizar a comparação entre as demais subamostragens, tem-se que para as sub-bases por Cluster centroides e NearMiss1, o algoritmo atingiu assertividades acima de 90%, sendo que a primeira técnica apresenta maiores métricas para esse classificador. No caso da F1, ilustrado na Figura 21b o classificador RF manteve o mesmo comportamento, devido a representatividade das técnicas de subamostragem.

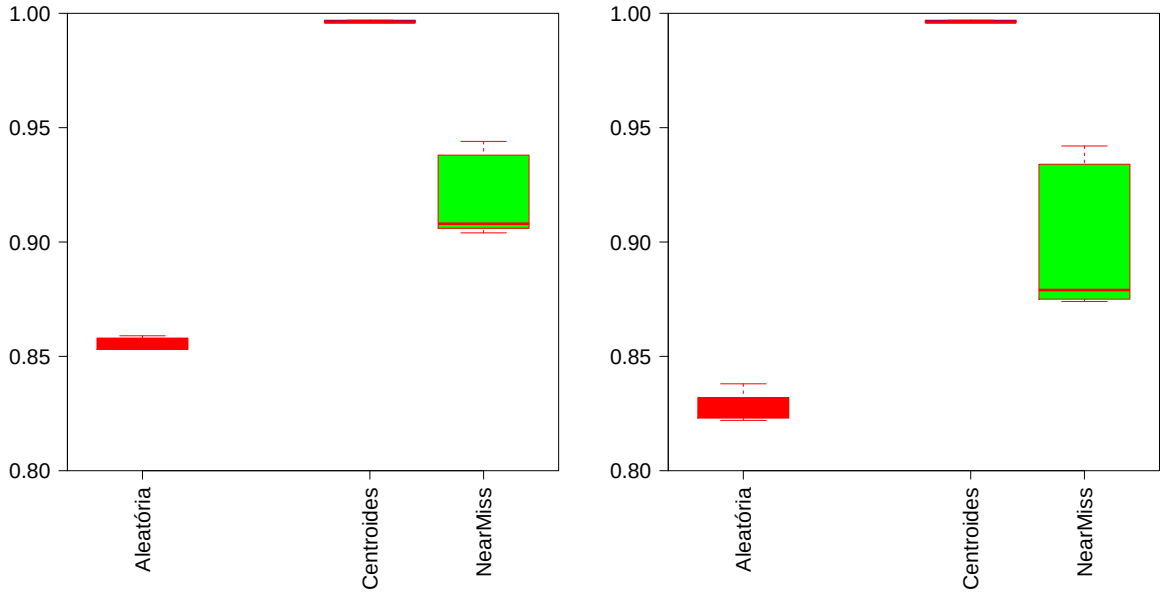
A acurácia do classificador KNN com relação as sub-bases é exibida na Figura 22a, na qual a abordagem por Cluster centroides apresenta acurácia maior que as demais, reafirmando capacidade de seleção das amostras mais importantes para a base de dados CIC2018. Além disso, vale ressaltar que o tempo de reconhecimento é viável para um projeto de SDIs, enquando na abordagem da base completa, o tempo de reconhecimento é alto o suficiente para considerar inviável em um contexto real. O mesmo comportamento ocorre quanto à F1, ilustrada na Figura 22b.

A comparação entre a acurácia do classificador SVM com relação as bases subamostradas é mostrada na Figura 23a. As subamostragens por Cluster centroides e NearMiss1 apresentam resultados acima de 85% de acurácia para o SVM, enquanto na sub-base aleatória, a acurácia é menor. A sub-base gerada por Cluster centroides possui a melhor representatividade,



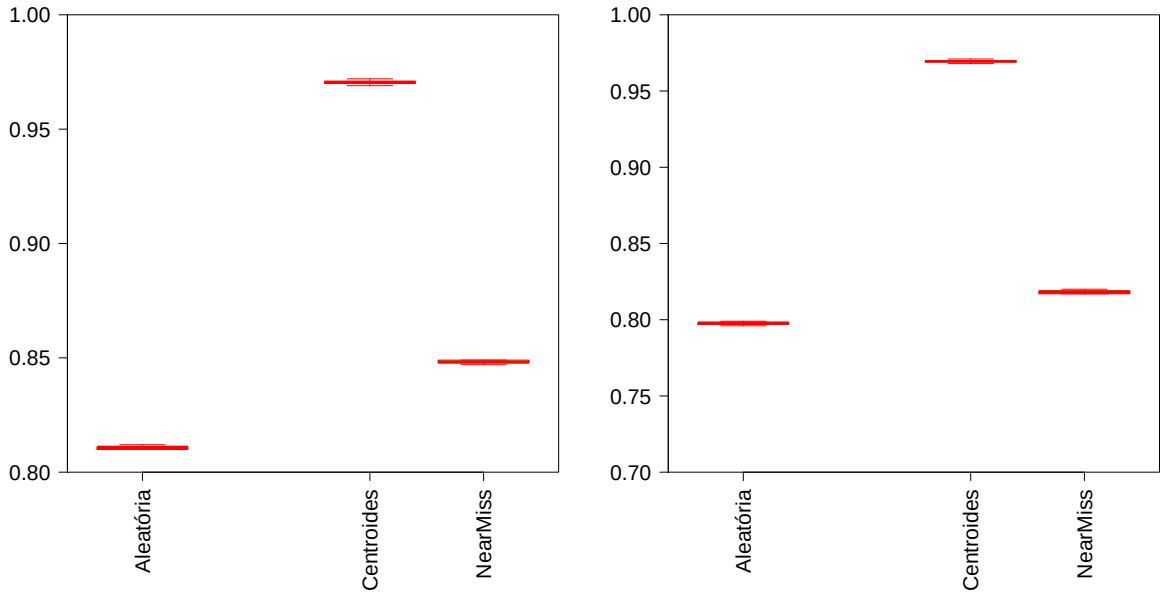
(a) (b)

Figura 21 – Comparação entre métricas de diferentes subamostragens para o classificador RF: a) Acurácia e b) F1



(a) (b)

Figura 22 – Comparação entre métricas de diferentes subamostragens para o classificador KNN: a) Acurácia e b) F1



(a) (b)

Figura 23 – Comparação entre métricas de diferentes subamostragens para o classificador SVM: a) Acurácia e b) F1

sendo possível alcançar taxas de até 97%. O mesmo comportamento ocorre quanto à F1, ilustrada na Figura 23b.

Comparação entre classificadores por meio do teste de Wilcoxon

As comparações entre a base completa e sub-bases por Cluster centroides, aleatória e por NearMiss1, bem como a base CIC2018 completa são mostradas nas tabelas 18 e 19 . O teste estatístico confirma que as métricas nos classificadores NC,NB e KNN, RF e SVM são superiores em um cenário de subamostragem por Cluster centroides com relação às demais subamostragens. Ao comparar com a base de dados completa, o mesmo ocorre para os classificadores NC, NB e RF, exceto quanto a Precision no primeiro algoritmo. O mesmo comportamento ocorre para os classificadores NC e NB em um cenário de subamostragem aleatória em relação a base de dados completa. A Precision do classificador Nearest centroid para a base completa é estatisticamente maior que no cenário das subamostragens.

Tabela 18 – Comparação entre base completa e por Cluster centroides, bem como por NearMiss1 e Cluster Centroides e base aleatória e por NearMiss1

Valores de T e |z| para n = 10, T crítico = 5, z crít. 2,33 e signif. de 0,01								
Algoritmo	Acurácia T / |z|		Precision T / |z|		Recall T / |z|		F1 T / |z|	
NC	0	2,8	0	2,8	0	2,8	0	2,8
NB	0	2,8	0	2,8	0	2,8	0	2,8
KNN	0	2,8	0	2,8	0	2,8	0	2,8
RF	0	2,8	0	2,8	0	2,8	0	2,8
SVM	0	2,8	0	2,8	0	2,8	0	2,8

Fonte: o autor.

Tabela 19 – Comparação entre base completa e por Cluster centroides, NearMiss1 e aleatória

Valores de T e |z| para n = 10, T crítico = 5, z crít. 2,33 e signif. de 0,01								
Algoritmo	Acurácia T / |z|		Precision T / |z|		Recall T / |z|		F1 T / |z|	
NC	0	2,8	**0**	**2,65**	0	2,8	0	2,8
NB	0	2,8	0	2,8	0	2,8	0	2,8
RF	0	2,8	0	2,8	0	2,8	0	2,8

Fonte: o autor.

5.4 Comparação com os trabalhos na literatura

O uso de aprendizado de máquina na detecção de intrusões é um tema amplamente abordado na literatura, de forma que a escolha do melhor algoritmo é uma tarefa de extrema relevância no projeto de SDI (SILVA NETO; GOMES, 2019). O processo de subamostragem tem sido utilizado ao longo dos anos (UTIMURA; COSTA, 2018; GAO *et al.*, 2019; DHANABAL; SHANTHARAJAH, 2015; D'HOOGE *et al.*, 2019), mas infelizmente sua adoção não tem recebido a atenção devida. Esta dissertação apresenta uma abordagem dirigida à redução do número de amostras, visando à representatividade em bases de dados recentes a partir de classificadores encontrados na literatura.

Com base nos resultados obtidos, observa-se que o uso de técnicas de subamostragem representativas buscam lidar com bases de dados desbalanceadas, visando aumentar a assertividade em classificadores para o projeto de Sistemas de Detecção de Intrusão. Dentre os classificadores avaliados, o KNN destacou-se com as melhores métricas, entretanto observa-se o compromisso com o tempo

Trabalhos na literatura (UTIMURA; COSTA, 2018; SILVA NETO; GOMES, 2019; D'HOOGE *et al.*, 2019) exploraram a base de dados CIC2017, obtendo resultados semelhantes

Tabela 20 – Principais resultados dos trabalhos encontrados na literatura

Autor	Base de dados	Algoritmos	Métricas	Médias
SILVA NETO e Gomes (2019)	CIC2017	NC	Precision / Recall / F1	0,83 / 0,49 / 0,61
SILVA NETO e Gomes (2019)	CIC2017	NB	Precision / Recall / F1	0,90 / 0,58 / 0,70
SILVA NETO e Gomes (2019)	CIC2017	RF	Precision / Recall / F1	0,84 / 0,88 / 0,86
SILVA NETO e Gomes (2019)	CIC2017	KNN	Precision / Recall / F1	0,84 / 0,88 / 0,85
Sharafaldin *et al.* (2018)	CIC2017	KNN	Precision / Recall / F1	0,96 / 0,96 / 0,96
Sharafaldin *et al.* (2018)	CIC2017	NB	Precision / Recall / F1	0,88 / 0,04 / 0,04
D'hooge *et al.* (2019)	CIC2017, CIC2018	KNN	Precision / Recall	0,99 / 0,99

Fonte: o autor.

aos encontrados para a base de dados completa. A Tabela 20 sintetiza esses principais resultados, nos quais percebem-se semelhanças entre os valores encontrados nesta dissertação para as bases completas CIC2017 e CIC2018.

Salienta-se que uma das contribuições desta dissertação é o tratamento de características categóricas para o melhor reconhecimento do modelo e consequentemente predição de novos valores. Além disso, o uso de tais técnicas de subamostragem viabilizam o uso em cenários com poucos recursos computacionais ou em tempo real.

Três técnicas de subamostragem foram utilizadas para realizar a avaliação dos classificadores: aleatória, Cluster Centroides e NearMiss1. A escolhas técnicas de subamostragem foi balizada nos seguintes motivos: a subamostragem aleatória é a abordagem padrão amplamente utilizada na literatura, sem o devido detalhamento; a subamostragem por cluster centroides utiliza o conceito de seleção de amostras baseadas no centroide de cada classe, buscando um agrupamento entre as amostras selecionadas; a subamostragem por NearMiss1 emprega o conceito de fronteira entre as classes para a seleção das amostras mais representativas. Tais técnicas são utilizadas na literatura em contextos diferentes tais como imagens (ALTBACH *et al.*, 2019), voz (KATHIRESAN *et al.*, 2019), financeiro (TSAI *et al.*, 2019), dentre outros. O estudo desenvolvido por Peng *et al.* (2019) avaliou as técnicas Cluster centroides e NearMiss1 em diferentes contextos de bases de dados, obtendo resultados que permitem concluir que a primeira técnica é mais representativa que a segunda, obtendo métricas de assertividade melhores na comparação. Os resultados inicialmente buscam a comparação entre as subamostragens e a base completa e posteriormente comparam-se as técnicas de subamostragem.

Os resultados obtidos indicam que a abordagem por Cluster centroides favoreceu em todos os cenários os classificadores baseados em distância, bem como o classificador Naive Bayes, aumentando de forma relevante as métricas de assertividade e diminuindo o tempo de reconhecimento do modelo. Os resultados evidenciaram ainda que, por meio de teste estatístico, as métricas obtidas pelos classificadores em sua maioria são superiores em classificadores avaliados sob a sub-base gerada a partir desta abordagem. Neste cenário, os resultados desta

dissertação apontam para o fato de que o uso da subamostragem baseada em Cluster centroides é recomendada ao avaliar algoritmos baseados em distância. Foram utilizadas 5 métricas de avaliação dos classificadores para técnica de subamostragem e a escolha do melhor classificador é uma decisão de projeto que pode utilizar uma das métricas ou uma combinação das mesmas. Nesta dissertação, apesar de utilizar-se de 5 métricas para a avaliação dos algoritmos de aprendizagem de máquina, são adotadas 2 métricas de assertividade, são elas - 1. acurácia e 2. F1, como valores gerais e utilizou-se o tempo de treinamento como um critério adicional para de viabilidade para o projeto de SDIs.

Nas bases de dados completas e subamostradas, o classificador KNN obteve as maiores métricas de assertividade. Entretanto observa-se o compromisso entre assertividade e tempo de treinamento, conforme discutido nos capítulos anteriores e reforçado no trabalho de SILVA NETO e Gomes (2019), os quais obtém resultados competitivos para o classificador KNN, entretanto o tempo de testes foi decisivo para a escolha de outros classificadores para o projeto de SDIs.

Com base nos critérios adotados para a escolha dos classificadores, quanto às bases de dados CIC2017 e CIC2018 completas, o classificador Florestas aleatórias obteve os melhores resultados, com métricas médias de acurácia, F1 e tempo de 88,6%, 85,0%, 64 segundos e 83,2%, 75,6% e 410 segundos respectivamente em cada base de dados.

Quanto às subamostragens aleatórias, o KNN foi considerado o melhor classificador por suas métricas médias de acurácia, F1 e tempo de treinamento, com valores iguais a 98,7%, 98,7%, 3,2 segundos na base subamostrada a partir do CIC2017 e 85,5%, 82,8%, 6,8 segundos para a base subamostrada a partir da CIC2018. Embora o classificador Naive Bayes tenha obtido métricas de assertividade 14% e 9% menores respectivamente em cada base subamostrada, observa-se que é 8 (oito) vezes mais rápido para a subamostragem a partir da CIC2017 e 12 (doze) vezes mais rápido para a subamostragem a partir da CIC2018 em termos de tempo de treinamento.

Já nas sub-bases por Cluster centroides, o classificador Naive Bayes obteve os melhores resultados, com métricas médias de acurácia, F1 e tempo de treinamento de 98,7%, 98,7% e 0,3 segundos na base subamostrada a partir do CIC2017 e 98,1%, 98,3% e 0,3 segundos para a base subamostrada a partir da CIC2018.

Quanto às sub-bases por NearMiss1, é observado que para a base subamostrada a partir do CIC2017, o classificador KNN é considerado o melhor por suas métricas médias

de acurácia, F1 e tempo de treinamento, com valores iguais a 99,4%, 99,4% e 3,4 segundos. Embora o classificador Naive Bayes tenha obtido métricas de assertividade 10% menor, o tempo de treinamento é 6 (seis) vezes mais rápido. Já para a base subamostrada a partir da CIC2018, o classificador Naive Bayes é considerado o melhor por suas métricas médias de acurácia, F1 e tempo de treinamento, com valores iguais a 91,1%, 90,5% e 0,4 segundos.

6 CONCLUSÕES, CONTRIBUIÇÕES E TRABALHOS FUTUROS

Esta dissertação apresenta uma avaliação de três tipos de subamostragem em bases de dados recentes, CIC2017 e CIC2018 (SHARAFALDIN *et al.*, 2018) que contém registros de dados de intrusão. Foi avaliada a performance dos algoritmos Nearest Centroid, Naive Bayes, Florestas Aleatórias, K-Vizinhos mais próximos (KNN) e Máquinas de vetores de suporte (SVM) nas duas bases de dados completas, assim como em sub-bases geradas pelas técnicas de seleção aleatória, Cluster centroides e NearMiss. Destaca-se que a avaliação visa selecionar os melhores classificador e técnica de subamostragem para projetos de SIDs. Assim, a partir dos resultados obtidos o classificador KNN obteve as melhores métricas nas bases/sub-bases nas quais foi avaliado. Entretanto, com base nos critérios adotados para a escolha dos melhores classificadores no projeto de SDIs, quanto às bases de dados CIC2017 e CIC2018 completas, o classificador Florestas aleatórias obtém os melhores resultados. Quanto à sub-base gerada, a partir da base CIC2017, pela subamostragem aleatória, o KNN foi considerado o melhor classificador por suas métricas médias de acurácia, eficiência e tempo de treinamento. Já na sub-base usando a técnica de subamostragem Cluster centroides, gerada a partir da CIC2018, o classificador Naive Bayes obtém os melhores resultados. Quanto às sub-bases geradas a partir das bases CIC2017 e CIC2018, empregando-se a técnica de subamostragem NearMiss, os melhores classificadores, por suas métricas médias de acurácia, eficiência e tempo de treinamento, foram o KNN e Naive Bayes, respectivamente.

O processo de subamostragem apresenta condições adequadas para permitir a avaliação de diferentes classificadores, inclusive aqueles que possuem tempo elevado de processamento. Além disso, o uso dessas técnicas permite melhorar o desbalanceamento nas bases e consequentemente diminuir a obliquidade nas mesmas.

A partir dos resultados obtidos, pode-se concluir que a subamostragem por Cluster centroides apresenta o melhor desempenho quando aplicados em classificadores baseados em distância. Conclui-se ainda que a técnica de subamostragem influencia no processo de escolha do melhor classificador no projeto de um Sistema de Detecção de Intrusão.

As principais contribuições dessa dissertação, não encontradas na literatura pesquisada, são: o emprego de técnicas de subamostragem em bases de dados de intrusão recentes, visando a representatividade por meio da correção do desbalanceamento nesses tipos de dados; análise exploratória das bases de dados CIC2017 e CIC2018; avaliação do desempenho de classificadores na base de dados CIC2018 completa; criação de bases por subamostragem, a

partir de diferentes técnicas aplicadas nas bases CIC2017 e CIC2018.

Esta dissertação não exauriu as possibilidades de pesquisas de técnicas de subamostragem, bem como na escolha de classificadores no projeto de SDIs. Para isto, como trabalhos futuros, recomenda-se o uso de técnicas de clusterização tais como o K-médias para obter uma melhor análise exploratória de bases de dados desbalanceadas, bem como a aplicação de técnicas de redução de dimensionalidade tais como Mean Decrease Impurity, visando diminuir a complexidade computacional do problema. Outra abordagem que pode ser adotada é a avaliação dos classificadores considerando classificação binária, em que todos os ataques representam um padrão anormal de comportamento, a fim de reduzir os efeitos do desbalanceamento. A geração de ataques reais, bem como a abordagem *cross-dataset* entre as bases de dados estudadas também pode ser considerado, com o objetivo de validação dos estudos na área de detecção de intrusões.

REFERÊNCIAS

ABDULHAMMED, R.; MUSAFER, H.; ALESSA, A.; FAEZIPOUR, M.; ABUZNEID, A. Features dimensionality reduction approaches for machine learning based network intrusion detection. **Electronics**, Multidisciplinary Digital Publishing Institute, v. 8, n. 3, p. 322, 2019.

AKSU, D.; AYDIN, M. A. Detecting port scan attempts with comparative analysis of deep learning and support vector machine algorithms. In: INTERNATIONAL CONGRESS ON BIG DATA, DEEP LEARNING AND FIGHTING CYBER TERRORISM (IBIGDELFT). **Proceedings [...]**. Ankara, Turkey: IEEE, 2018. p. 77–80.

ALJAWARNEH, S.; ALDWAIRI, M.; YASSEIN, M. B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. **Journal of Computational Science**, Elsevier, v. 25, p. 152–160, 2018.

ALTBACH, M.; BILGIN, A.; HUANG, C.; GRAFF, C. **System and method for image processing with highly undersampled imaging data**. [S.l.]: Google Patents, 2019. US Patent 10,393,839.

ARAUJO, T. E. de S.; MATOS, F. M.; MOREIRA, J. A. Intrusion detection systems' performance for distributed denial-of-service attack. In: CONFERENCE ON ELECTRICAL, ELECTRONICS ENGINEERING, INFORMATION AND COMMUNICATION TECHNOLOGIES (CHILECON). **Proceedings [...]**. Pucon, Chile: IEEE, 2017. p. 1–6.

BEIGI, E. B.; JAZI, H. H.; STAKHANOVA, N.; GHORBANI, A. A. Towards effective feature selection in machine learning-based botnet detection approaches. In: IEEE CONFERENCE ON COMMUNICATIONS AND NETWORK SECURITY. **Proceedings [...]**. San Francisco, CA,USA: IEEE, 2014. p. 247–255.

BHASKAR, T.; HIWARKAR, T.; RAMANJANEYULU, K. Adaptive jaya optimization technique for feature selection in nsl-kdd data set of intrusion detection system. In: INTERNATIONAL CONFERENCE ON COMMUNICATION AND INFORMATION PROCESSING (ICCIP). **Proceedings [...]**. Chongqing, China: SSRN, 2019. p. 53–59.

BHATTACHARJEE, P. S.; FUJAIL, A. K. M.; BEGUM, S. A. A comparison of intrusion detection by k-means and fuzzy c-means clustering algorithm over the nsl-kdd dataset. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND COMPUTING RESEARCH (ICCIC). **Proceedings [...]**. Coimbatore, India: IEEE, 2017. p. 1–6. ISSN 2471-7851.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. Smote: synthetic minority over-sampling technique. **Journal of artificial intelligence research**, v. 16, p. 321–357, 2002.

DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. **Journal of Machine learning research**, v. 7, n. Jan, p. 1–30, 2006.

DEVILLE, J.-C.; TILLÉ, Y. Efficient balanced sampling: the cube method. **Biometrika**, Oxford University Press, v. 91, n. 4, p. 893–912, 2004.

DHANABAL, L.; SHANTHARAJAH, S. A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. **International Journal of Advanced Research in Computer and Communication Engineering**, v. 4, n. 6, p. 446–452, 2015.

DIETTERICH, T. G. Approximate statistical tests for comparing supervised classification learning algorithms. **Neural computation**, MIT Press, v. 10, n. 7, p. 1895–1923, 1998.

D'HOOGE, L.; WAUTERS, T.; VOLCKAERT, B.; TURCK, F. D. Classification hardness for supervised learners on 20 years of intrusion detection data. **IEEE Access**, v. 7, p. 167455–167469, 2019. ISSN 2169-3536.

FLACH, P.; KULL, M. Precision-recall-gain curves: Pr analysis done right. In: CORTES, C.; LAWRENCE, N. D.; LEE, D. D.; SUGIYAMA, M.; GARNETT, R. (Ed.). **Advances in Neural Information Processing Systems 28**. Curran Associates, Inc., 2015. p. 838–846. Disponível em: <http://papers.nips.cc/paper/5867-precision-recall-gain-curves-pr-analysis-done-right.pdf>.

GANGANWAR, V. An overview of classification algorithms for imbalanced datasets. **International Journal of Emerging Technology and Advanced Engineering**, Citeseer, v. 2, n. 4, p. 42–47, 2012.

GAO, X.; SHAN, C.; HU, C.; NIU, Z.; LIU, Z. An adaptive ensemble machine learning model for intrusion detection. **IEEE Access**, IEEE, v. 7, p. 82512–82521, 2019.

HE, H.; BAI, Y.; GARCIA, E. A.; LI, S. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE). **Proceedings [...]**. Hong Kong: IEEE, 2008. p. 1322–1328.

HINDY, H.; BROSSET, D.; BAYNE, E.; SEEAM, A.; TACHTATZIS, C.; ATKINSON, R.; BELLEKENS, X. A taxonomy and survey of intrusion detection system design techniques, network threats and datasets. **CoRR**, abs/1806.03517, p. 1–35, 2018. Disponível em: <http://arxiv.org/abs/1806.03517>.

HULSE, J. V.; KHOSHGOFTAAR, T. M.; NAPOLITANO, A. Experimental perspectives on learning from imbalanced data. In: 24TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING. **Proceedings [...]**. Corvalis, Oregon, USA: ACM, 2007. p. 935–942.

JAZI, H. H.; GONZALEZ, H.; STAKHANOVA, N.; GHORBANI, A. A. Detecting http-based application layer dos attacks on web servers in the presence of sampling. **Computer Networks**, Elsevier, v. 121, p. 25–36, 2017.

JIAO, J.; YE, B.; ZHAO, Y.; STONES, R. J.; WANG, G.; LIU, X.; WANG, S.; XIE, G. Detecting tcp-based ddos attacks in baidu cloud computing data centers. In: 36TH SYMPOSIUM ON RELIABLE DISTRIBUTED SYSTEMS (SRDS). **Proceedings [...]**. Hong Kong: IEEE, 2017. p. 256–258.

KATHIRESAN, T.; MAURER, D.; DELLWO, V. Highly spectrally undersampled vowels can be classified by machines without supervision. **The Journal of the Acoustical Society of America**, Acoustical Society of America, v. 146, n. 1, p. EL1–EL7, 2019.

KHRAISAT, A.; GONDAL, I.; VAMPLEW, P.; KAMRUZZAMAN, J. Survey of intrusion detection systems: techniques, datasets and challenges. **Cybersecurity**, Springer, v. 2, n. 1, p. 20, 2019.

KOLIAS, C.; KAMBOURAKIS, G.; STAVROU, A.; VOAS, J. Ddos in the iot: Mirai and other botnets. **Computer**, v. 50, n. 7, p. 80–84, 2017. ISSN 1558-0814.

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet: Uma abordagem top-down**. Trad. 5 ed. São Paulo: Pearson, 2010.

LANTZ, B. **Machine learning with R**. [S.l.]: Packt Publishing Ltd, 2015.

LEE, J.; PARK, K. Gan-based imbalanced data intrusion detection system. **Personal and Ubiquitous Computing**, Springer, p. 1–8, 2019.

LEMAîTRE, G.; NOGUEIRA, F.; ARIDAS, C. K. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. **Journal of Machine Learning Research**, v. 18, n. 17, p. 1–5, 2017. Disponível em: <http://jmlr.org/papers/v18/16-365>.

LI, J.; QU, Y.; CHAO, F.; SHUM, H. P. H.; HO, E. S. L.; YANG, L. Machine learning algorithms for network intrusion detection. In: ____. **AI in Cybersecurity**. Cham: Springer International Publishing, 2019. p. 151–179. ISBN 978-3-319-98842-9. Disponível em: <https://doi.org/10.1007/978-3-319-98842-9_6>.

MANI, I.; ZHANG, I. knn approach to unbalanced data distributions: a case study involving information extraction. In: WORKSHOP ON LEARNING FROM IMBALANCED DATASETS. **Proceedings [...]**. Washington, DC, USA, 2003. v. 126.

MAZA, S.; TOUAHRIA, M. Feature selection for intrusion detection using new multi-objective estimation of distribution algorithms. **Applied Intelligence**, Springer, v. 1, p. 1–21, 2019.

MEENA, G.; CHOUDHARY, R. R. A review paper on ids classification using kdd 99 and nsl kdd dataset in weka. In: 2017 INTERNATIONAL CONFERENCE ON COMPUTER, COMMUNICATIONS AND ELECTRONICS (COMPTELIX). **Proceedings [...]**. Jaipur, India: IEEE, 2017. p. 553–558. ISSN null.

MEILE, L.; ULRICH, A.; MAGNO, M. Wireless power transmission powering miniaturized low power iot devices: A review. In: 8TH INTERNATIONAL WORKSHOP ON ADVANCES IN SENSORS AND INTERFACES (IWASI). **Proceedings [...]**. Otranto, Italy: IEEE, 2019. p. 312–317. ISSN null.

MISHRA, P.; VARADHARAJAN, V.; TUPAKULA, U.; PILLI, E. S. A detailed investigation and analysis of using machine learning techniques for intrusion detection. **IEEE Communications Surveys & Tutorials**, IEEE, v. 21, n. 1, p. 686–728, 2018.

MORE, A. Survey of resampling techniques for improving classification performance in unbalanced datasets. **CoRR**, abs/1608.06048, p. 1–7, 2016. Disponível em: <http://arxiv.org/abs/1608.06048>.

OSTERWEIL, E.; STAVROU, A.; ZHANG, L. 20 years of ddos: a call to action. **CoRR**, abs/1904.02739, n. 1, p. 1–11, 2019. Disponível em: <http://arxiv.org/abs/1904.02739>.

OTT, R. L.; LONGNECKER, M. T. **An introduction to statistical methods and data analysis**. [S.l.]: Nelson Education, 2015.

PARSAEI, M. R.; ROSTAMI, S. M.; JAVIDAN, R. A hybrid data mining approach for intrusion detection on imbalanced nsl-kdd dataset. **International Journal of Advanced Computer Science and Applications**, The Science and Information Organization, v. 7, n. 6, 2016. Disponível em: <http://dx.doi.org/10.14569/IJACSA.2016.070603>.

PAXSON, V. Bro: a system for detecting network intruders in real-time. **Computer networks**, Elsevier, v. 31, n. 23-24, p. 2435–2463, 1999.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PENG, M.; ZHANG, Q.; XING, X.; GUI, T.; HUANG, X.; JIANG, Y.-G.; DING, K.; CHEN, Z. Trainable undersampling for class-imbalance learning. In: AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE. **Proceedings [...]**. Honolulu, Hawaii, 2019. v. 33, p. 4707–4714.

RING, M.; WUNDERLICH, S.; GRÜDL, D.; LANDES, D.; HOTHO, A. Creation of flow-based data sets for intrusion detection. **Journal of Information Warfare**, JSTOR, v. 16, n. 4, p. 41–54, 2017.

RING, M.; WUNDERLICH, S.; GRÜDL, D.; LANDES, D.; HOTHO, A. Flow-based benchmark data sets for intrusion detection. In: 16TH EUROPEAN CONFERENCE ON CYBER WARFARE AND SECURITY. ACPI. **Proceedings [...]**. Dublin, Ireland, 2017. p. 361–369.

ROESCH, M. *et al.* Snort: Lightweight intrusion detection for networks. In: 13TH SYSTEMS ADMINISTRATION CONFERENCE. **Proceedings [...]**. Seattle, Washington, USA: USENIX, 1999. v. 99, n. 1, p. 229–238.

SAHU, S. K.; SARANGI, S.; JENA, S. K. A detail analysis on intrusion detection datasets. In: 2014 IEEE INTERNATIONAL ADVANCE COMPUTING CONFERENCE (IACC). **Proceedings [...]**. Gurgaon, India: IEEE, 2014. p. 1348–1353.

SALIM, M. M.; RATHORE, S.; PARK, J. H. Distributed denial of service attacks and its defenses in iot: a survey. **The Journal of Supercomputing**, Springer, p. 1–44, 2019.

SHARAFALDIN, I.; LASHKARI, A. H.; GHORBANI, A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: INSTICC. **Proceedings [...]**. Ilha da madeira, Portugal: SciTePress, 2018. p. 108–116. ISBN 978-989-758-282-0.

SHESKIN, D. J. **Handbook of parametric and nonparametric statistical procedures**. [S.l.]: Chapman and Hall/CRC, 2003.

SHIRAVI, A.; SHIRAVI, H.; TAVALLAEE, M.; GHORBANI, A. A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. **Computers & Security**, v. 31, n. 3, p. 357 – 374, 2012. ISSN 0167-4048. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167404811001672>.

SILVA NETO, M. G.; GOMES, D. G. Network intrusion detection systems design: A machine learning approach. In: XXXVI SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS. **Anais [...]**. Porto Alegre, RS, Brasil: SBC, 2019. p. 932–945. ISSN 2177-9384. Disponível em: <https://sol.sbc.org.br/index.php/sbrc/article/view/7413>.

SILVA NETO, M. G.; GOMES, D. G.; SOARES, J. M. Credibility on crowdsensing data acquisition. **Journal of Communication and Information Systems**, v. 34, n. 1, p. 248–269, 2019.

SONAK, A.; PATANKAR, R. A survey on methods to handle imbalance dataset. **Int. J. Comput. Sci. Mobile Comput**, v. 4, n. 11, p. 338–343, 2015.

STIAWAN, D.; SANDRA, S.; ALZAHRANI, E.; BUDIARTO, R. Comparative analysis of k-means method and naïve bayes method for brute force attack visualization. In: 2ND INTERNATIONAL CONFERENCE ON ANTI-CYBER CRIMES (ICACC). **Proceedings [...]**. Abha, Saudi Arabia, 2017. p. 177–182. ISSN null.

Thomas, R.; Pavithran, D. A survey of intrusion detection models based on nsl-kdd data set. In: 2018 FIFTH HCT INFORMATION TECHNOLOGY TRENDS (ITT). **Proceedings [...]**. [S.l.], 2018. p. 286–291. ISSN null.

TIBSHIRANI, R.; HASTIE, T.; NARASIMHAN, B.; CHU, G. Diagnosis of multiple cancer types by shrunken centroids of gene expression. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 99, n. 10, p. 6567–6572, 2002.

TOMAR, D. A survey on data mining approaches for healthcare. **International Journal of Bio - Science and Bio - Technology**, v. 5, p. 241–266, 10 2013.

TSAI, C.-F.; LIN, W.-C.; HU, Y.-H.; YAO, G.-T. Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. **Information Sciences**, v. 477, p. 47 – 54, 2019. ISSN 0020-0255. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025518308478>.

ULLAH, I.; MAHMOUD, Q. H. An intrusion detection framework for the smart grid. In: 30TH CANADIAN CONFERENCE ON ELECTRICAL AND COMPUTER ENGINEERING (CCECE). **Proceedings [...]**. Windsor, Canada: IEEE, 2017. p. 1–5.

UTIMURA, L. N.; COSTA, K. A. Aplicação e análise comparativa do desempenho de classificadores de padrões para o sistema de detecção de intrusão snort. In: XXXVI SIMPóSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUíDOS. **Anais [...]**. Porto Alegre, RS, Brasil: SBC, 2018. ISSN 2177-9384. Disponível em: <https://sol.sbc.org.br/index.php/sbrc/article/view/2426>.

VAPNIK, V. The support vector method of function estimation. In: **Nonlinear Modeling**. [S.l.]: Springer, 1998. p. 55–85.

VLUYMANS, S. **Dealing with imbalanced and weakly labelled data in machine learning using fuzzy and rough set methods**. [S.l.]: Springer, 2019. v. 807.

XYLOGIANNOPOULOS, K. F.; KARAMPELAS, P.; ALHAJJ, R. Advanced network data analytics for large-scale ddos attack detection. **International Journal of Cyber Warfare and Terrorism (IJCWT)**, IGI Global, v. 7, n. 3, p. 44–54, 2017.

ZHANG, C.; RUAN, F.; YIN, L.; CHEN, X.; ZHAI, L.; LIU, F. A deep learning approach for network intrusion detection based on nsl-kdd dataset. In: 13TH INTERNATIONAL CONFERENCE ON ANTI-COUNTERFEITING, SECURITY, AND IDENTIFICATION (ASID). **Proceedings [...]**. Xiamen, China: IEEE, 2019. p. 41–45. ISSN 2163-5048.

APÊNDICE A – TABELAS UTILIZADAS NA DISSERTAÇÃO

Tabela 21 – Análise exploratória CIC2017

Característica	Média	Desvio	Min	Max	Skewness
destination_port	8061,534	18274,320	0	65535	2,074
flow_duration	14800654,15	33667504,66	-13	119999998	2,154
total_fwd_packets	9,369	750,053	1	219759	244,257
total_backward_packets	10,404	997,894	0	291922	244,556
total_length_of_fwd_packets	549,852	9998,639	0	12900000	805,166
total_length_of_bwd_packets	16179,027	2264234,902	0	655453030	244,207
fwd_packet_length_max	207,804	717,518	0	24820	9,845
fwd_packet_length_min	18,729	60,355	0	2325	20,129
fwd_packet_length_mean	58,256	186,173	0	5940,857143	9,132
fwd_packet_length_std	68,978	281,321	0	7125,596846	10,526
bwd_packet_length_max	871,730	1947,157	0	19530	2,936
bwd_packet_length_min	41,089	68,881	0	2896	4,835
bwd_packet_length_mean	306,257	605,486	0	5800,5	2,518
bwd_packet_length_std	335,666	840,051	0	8194,660487	3,412
flow_bytes/s	1491719,064	25940155,67	-261000000	2071000000	46,392
flow_packets/s	70854,233	254415,438	-2000000	4000000	5,517
flow_iat_mean	1299765,122	4510039,065	-13	120000000	8,966
flow_iat_std	2922230,578	8049410,126	0	84800261,57	3,609
flow_iat_max	9191784,833	24470186,67	-13	120000000	2,900
flow_iat_min	162544,188	2951772,415	-14	120000000	23,776
fwd_iat_total	14497645,07	33589658,92	0	120000000	2,175
fwd_iat_mean	2612839,279	9530187,265	0	120000000	7,063
fwd_iat_std	3270269,378	9643378,777	0	84602929,28	3,409
fwd_iat_max	9052106,608	24539897,65	0	120000000	2,908
fwd_iat_min	1022928,936	8595728,741	-12	120000000	9,708
bwd_iat_total	9903861,063	28749449,93	0	120000000	2,882
bwd_iat_mean	1807614,534	8891514,923	0	120000000	8,246
bwd_iat_std	1487479,879	6281472,052	0	84418013,78	5,790
bwd_iat_max	4689441,934	17168998,43	0	120000000	4,433
bwd_iat_min	968242,0143	8313136,887	0	120000000	9,919
fwd_psh_flags	0,0464	0,210	0	1	4,313
bwd_psh_flags	0	0	0	0	0,000
fwd_urg_flags	0,000111	0,0106	0	1	94,733
bwd_urg_flags	0	0	0	0	0,00000000
fwd_header_length	-26023,795	21063527,22	-32212234632	4644908	-1324,403
bwd_header_length	-2275,588	1452944,902	-1073741320	5838440	-716,668
fwd_packets/s	63930,095	247654,222	0	3000000	5,682
bwd_packets/s	7002,284	38170,382	0	2000000	21,479
min_packet_length	16,449	25,246	0	1448	10,192
max_packet_length	951,359	2029,034	0	24820	2,813
packet_length_mean	172,114	305,599	0	3337,142857	2,394
packet_length_std	295,272	632,051	0	4731,522394	2,881
packet_length_variance	486646,832	1648252,022	0	22400000	5,391

fin_flag_count	0,035	0,184	0	1	5,038
syn_flag_count	0,046	0,210	0	1	4,313
rst_flag_count	0,000	0,016	0	1	64,182
psh_flag_count	0,298	0,458	0	1	0,882
ack_flag_count	0,315	0,465	0	1	0,795
urg_flag_count	0,095	0,293	0	1	2,766
cwe_flag_count	0,000	0,011	0	1	94,733
ece_flag_count	0,000	0,016	0	1	64,042
down/up_ratio	0,684	0,681	0	156	12,011
average_packet_size	192,171	331,973	0	3893,333333	2,489
avg_fwd_segment_size	58,256	186,173	0	5940,857143	9,132
avg_bwd_segment_size	306,257	605,486	0	5800,5	2,518
fwd_header_length,1	-26023,795	21063527,220	-32212234632	4644908	-1324,403
fwd_avg_bytes/bulk	0	0	0	0	0,000
fwd_avg_packets/bulk	0	0	0	0	0,000
fwd_avg_bulk_rate	0	0	0	0	0,000
bwd_avg_bytes/bulk	0	0	0	0	0,000
bwd_avg_packets/bulk	0	0	0	0	0,000
bwd_avg_bulk_rate	0	0	0	0	0,000
subflow_fwd_packets	9,369	750,053	1	219759	244,257
subflow_fwd_bytes	549,842	9985,113	0	12870338	803,195
subflow_bwd_packets	10,404	997,894	0	291922	244,556
subflow_bwd_bytes	16178,686	2264204,118	0	655453030	244,212
init_win_bytes_forward	6992,389	14340,221	-1	65535	2,562
init_win_bytes_backward	1988,290	8454,537	-1	65535	5,254
act_data_pkt_fwd	5,424	636,748	0	213557	284,451
min_seg_size_forward	-2744,494	1085539,241	-536870661	138	-474,295
active_mean	81634,001	648923,440	0	110000000	38,215
active_std	41175,820	393578,695	0	74200000	40,452
active_max	153337,823	1026333,251	0	110000000	24,340
active_min	58354,920	577381,759	0	110000000	47,660
idle_mean	8324467,717	23640569,691	0	120000000	3,063
idle_std	504354,764	4605289,253	0	76900000	10,488
idle_max	8704568,043	24377663,276	0	120000000	2,949
idle_min	7928060,620	23373897,540	0	120000000	3,182

Fonte: elaborado pelo autor.

Tabela 22 – Análise exploratória CIC2018

Característica	Média	Desvio	Min	Max	Skewness
dst_port	8926,16	18673,50	0,0	65535,0	1,915
protocol	8,73	4,91	0,0	17,0	1,013
flow_duration	12418159,42	31091098,28	-2101872896,0	1553149696,0	2,581
tot_fwd_pkts	23,48	1515,17	1,0	309629,0	88,899
tot_bwd_pkts	6,48	169,48	0,0	123118,0	177,843
totlen_fwd_pkts	939,43	49315,41	0,0	9908128,0	90,119

totlen_bwd_pkts	4701,77	243366,76	0,0	156360432,0	169,398
fwd_pkt_len_max	196,69	286,23	0,0	64440,0	4,497
fwd_pkt_len_min	10,98	22,36	0,0	1460,0	11,283
fwd_pkt_len_mean	51,89	61,57	0,0	16529,31	7,008
fwd_pkt_len_std	70,09	107,72	0,0	18401,58	2,404
bwd_pkt_len_max	353,72	463,67	0,0	65160,0	1,606
bwd_pkt_len_min	26,28	47,96	0,0	1460,0	2,125
bwd_pkt_len_mean	116,18	152,93	0,0	33879,28	4,319
bwd_pkt_len_std	132,79	202,89	0,0	22448,41	1,457
flow_byts/s	252728,62	3637267,25	0,0	1806642816,0	77,077
flow_pkts/s	49403,91	256616,59	-0,0088	6000000,0	6,986
flow_iat_mean	3478534,75	221735808,0	-828219981824,0	120000000,0	-3389,329
flow_iat_std	1229537,5	336811616,0	0,0	474354483200,0	1213,074
flow_iat_max	6642085,5	656801792,0	-828219981824,0	979781025792,0	1016,257
flow_iat_min	2530806,0	744669888,0	-947404996608,0	120000000,0	-1174,288
fwd_iat_tot	11565587,0	491237184,0	-919011000320,0	120012736,0	-1440,167
fwd_iat_mean	3817706,5	221736032,0	-828219981824,0	120000000,0	-3388,805
fwd_iat_std	1363001,625	336811808,0	0,0	474354483200,0	1213,004
fwd_iat_max	6468145,0	656801664,0	-828219981824,0	979781025792,0	1016,268
fwd_iat_min	2616091,75	744670016,0	-947404996608,0	120000000,0	-1174,274
bwd_iat_tot	7455999,5	25074738,0	0,0	120000000,0	3,624
bwd_iat_mean	805745,125	4257750,5	0,0	120000000,0	12,432
bwd_iat_std	845622,93	3280177,25	0,0	84837192,0	6,269
bwd_iat_max	2527378,75	9945390,0	0,0	120000000,0	5,194
bwd_iat_min	287155,15	3785660,5	0,0	120000000,0	17,075
fwd_psh_flags	0,041	0,19	0,0	1,0	4,615
fwd_urg_flags	8,07e-05	0,0089	0,0	1,0	111,262
fwd_header_len	258,95	12252,65	0,0	2477032,0	88,074
bwd_header_len	136,60	3383,61	0,0	2462372,0	178,115
fwd_pkts/s	35182,78	208501,26	0,0	6000000,0	8,836
bwd_pkts/s	14542,63	89520,94	0,0	2000000,0	9,447
pkt_len_min	10,85	20,67	0,0	1460,0	6,618
pkt_len_max	37953244,0	5298762752,0	0,0	1095216660480,0	165,720
pkt_len_mean	77,47	98,50	0,0	17344,98	4,370
pkt_len_std	119,17	150,28	0,0	22788,28	1,698
pkt_len_var	41309,97	208787,81	0,0	519000000,0	1866,908
fin_flag_cnt	0,0043	0,065	0,0	1,0	15,100
syn_flag_cnt	0,041	0,19	0,0	1,0	4,615
rst_flag_cnt	0,18	0,38	0,0	1,0	1,608
psh_flag_cnt	0,39	0,48	0,0	1,0	0,422
ack_flag_cnt	0,32	0,46	0,0	1,0	0,730
urg_flag_cnt	0,041	0,19	0,0	1,0	4,589
cwe_flag_count	8,076e-05	0,0089	0,0	1,0	111,262
ece_flag_cnt	0,18	0,38	0,0	1,0	1,608
down/up_ratio	0,49	0,93	0,0	311,0	127,462
pkt_size_avg	89,97	101,06	0,0	17478,40	3,897
fwd_seg_size_avg	51,89	61,57	0,0	16529,31	7,008

bwd_seg_size_avg	116,18	152,93	0,0	33879,28	4,319
subflow_fwd_pkts	23,48	1515,17	1,0	309629,0	88,899
subflow_fwd_byts	960,54	49322,07	0,0	9908128,0	90,119
subflow_bwd_pkts	6,48	169,48	0,0	123118,0	177,843
subflow_bwd_byts	4913,43	243432,44	0,0	156360426,0	169,332
init_fwd_win_byts	8915,10	16327,52	-1,0	65535,0	2,570
init_bwd_win_byts	9119,72	21098,75	-1,0	65535,0	2,050
fwd_act_data_pkts	19,78	1513,82	0,0	309628,0	89,121
fwd_seg_size_min	17,97	7,66	0,0	56,0	0,252
active_mean	168209,85	2461798,75	0,0	114000000,0	25,169
active_std	84643,96	1491686,87	0,0	75232416,0	26,235
active_max	256515,71	3258213,75	0,0	114000000,0	20,528
active_min	112211,98	2077558,62	0,0	114000000,0	34,790
idle_mean	5106355,5	261877344,0	0,0	395571429376,0	1266,307
idle_std	282043,18	168538800,0	0,0	262247858176,0	1312,927
idle_max	5512851,5	622805376,0	0,0	979781025792,0	1336,620
idle_min	4784782,0	63566656,0	0,0	239933997056,0	3335,049

Fonte: elaborado pelo autor.

Tabela 23 – Quantidade de registros por classe nas bases subamostradas a partir da CICIDS2017

Classe	Número de amostras	% da classe	I_D
Normal	33526	19	1:1
DoS Hulk	33526	19	1:1
PortScan	33526	19	1:1
DDoS	33526	19	1:1
Dos GoldenEye	10293	5,9	1:3
FTP-Patator	7938	4,5	1:4
SSH-Patator	5897	3,3	1:5
DoS Slowloris	5796	3,3	1:5
DoS Slowhttptest	5499	3,3	1:6
Bot	1966	1,1	1:17
Web Attack Brute Force	1507	0,8	1:22
Web Attack XSS	652	0,3	1:52
Infiltration	36	0,2	1:931
Web Attack SQL Injection	21	0,1	1:1596
HeartBleed	11	0,06	1:3047
Total	173707	-	-

Fonte: o autor.

Tabela 24 – Quantidade de registros por classe nas bases subamostradas a partir da CICIDS2018

Classe	Número de amostras	% da classe	I_D
Normal	16006	9,2	1:1
DDOS attack-HOIC	16006	9,2	1:1
DDoS attacks-LOIC-HTTP	16006	9,2	1:1
DoS attacks-Hulk	16006	9,2	1:1
Bot	16006	9,2	1:1
FTP-BruteForce	16006	9,2	1:1
SSH-Bruteforce	16006	9,2	1:1
Infiltration	16006	9,2	1:1
DoS attacks-SlowHTTPTest	16006	9,2	1:1
DoS attacks-GoldenEye	16006	9,2	1:1
DoS attacks-Slowloris	10990	6,3	1:1
DDOS attack-LOIC-UDP	1730	0,99	1:9
Brute Force -Web	611	0,35	1:26
Brute Force -XSS	230	0,13	1:69
SQL Injection	87	0,05	1:183
Total	173708	100	-

Fonte: o autor.