UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Vinícius Alves Martins

Uma Arquitetura de Microsserviços para Detecção de Intrusões baseada em Técnicas de Aprendizado Contínuo e Rede de Conselhos

Uberlândia, Brasil

2024

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Vinícius Alves Martins

Uma Arquitetura de Microsserviços para Detecção de Intrusões baseada em Técnicas de Aprendizado Contínuo e Rede de Conselhos

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Rodrigo Sanches Miani

Coorientador: Prof. Dr. Silvio E. Quincozes

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2024

Vinícius Alves Martins

Uma Arquitetura de Microsserviços para Detecção de Intrusões baseada em Técnicas de Aprendizado Contínuo e Rede de Conselhos

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Prof. Dr. Rodrigo Sanches Miani
Orientador

Prof. Dr. Silvio E. Quincozes
Coorientador

Professor

Uberlândia, Brasil
2024

Agradecimentos

Primeiramente, quero agradecer a Deus, por sempre guiar a minha vida e me proporcionar tudo que pude viver na minha jornada. Foram muitos momentos bons e outros desafiadores, nos quais vi que, por minhas próprias forças, não teria conseguido enfrentar. Mas, com Ele, consegui enfrentar cada desafio e chegar até aqui.

Agradeço também à minha família e amigos, que sempre estiveram ao meu lado, me apoiando, aconselhando e dando suporte em todas as áreas, permitindo que eu atingisse meus objetivos. Ao meu pai, Luciano, e minha irmã, Isabela, agradeço por serem minha base, presentes em todos os momentos e participando ativamente da minha trajetória. Sem eles comigo, seria impossível chegar onde estou hoje. À minha mãe, Luciana (*in memorian*), tenho orgulho de tudo que vivemos juntos e por poder dizer que o que sou hoje devo também à ela, por sempre ser meu porto seguro e pelo amor que me acompanha até hoje.

À minha namorada, Amanda, agradeço profundamente por todo amor, incentivo e palavras de apoio que foram fundamentais em momentos de apreensão, incerteza ou desânimo durante esse tempo, pois ter alguém para te ouvir e te acompanhar nessa fase faz toda a diferença. Ainda nesse momento de gratidão, não posso deixar de mencionar meus amigos do curso, que sabem melhor do que ninguém as lutas e dificuldades que enfrentamos ao longo da graduação. Sem dúvidas, a formação é um grande desafio, mas tê-los ao lado tornou o caminho mais leve. Juntos compartilhamos momentos de alegria e de angústia, mas sempre motivando uns aos outros e contribuindo para essa caminhada.

Gostaria de expressar minha gratidão ao meu coorientador Silvio Quincozes, que compartilhou comigo seus conhecimentos e projetos, me orientando e instruindo em uma área que eu não tinha familiaridade, mas que hoje entrego essa pesquisa tão importante. Agradeço também ao meu orientador Rodrigo Miani, por ter sido tão empático e aceitado me orientar nessa fase final de graduação e pela passagem de conhecimento que agregou tanto à minha formação.

Por último, e não menos importante, agradeço a todos os professores que tive o prazer de aprender nesse período. Obrigado por cada ensinamento e experiência passada, que contribuíram ativamente para minha formação acadêmica, e que guardo com muito carinho.

Atenciosamente, Vinícius Alves Martins.

Resumo

As Redes de Conselhos consistem em uma estrutura de Sistemas de Detecção de Intrusões (IDSs) conectados uns aos outros. Tradicionalmente, tal conexão se dá através do paradigma de comunicação requisição e resposta, onde cada nó da rede precisa enviar requisições aos conselheiros individualmente. Este trabalho de conclusão de curso tem como objetivo apresentar uma arquitetura baseada em microsserviços para otimizar o fluxo de compartilhamento de conhecimento dentro de uma Rede de Conselhos. Esses microsserviços são desacoplados, e a comunicação entre eles ocorre através do conceito de publicação e assinatura, adicionando um *broker* de mensagens intermediário, implementado através do Apache Kafka. Como principais contribuições, este trabalho viabiliza o aprendizado contínuo de amostras maliciosas por IDSs de maneira otimizada por meio da arquitetura proposta. Os resultados mostram uma redução média de 90% na quantidade de conflitos encontrados após o aprendizado, além de reduzir pela metade o tempo médio de execução em relação à abordagem tradicional. Portanto, observou-se maior escalabilidade, manutenibilidade do sistema e uma estrutura desacoplada, na qual o sistema processa mensagens ininterruptamente enquanto aguarda mensagens de conselhos que são recebidas assincronamente. Por fim, a arquitetura proposta permite que trabalhos futuros explorem o aprendizado passivo, onde IDSs aprendem com conselhos compartilhados entre os demais nós da rede.

Palavras-chave: Rede de Conselhos, IDS, Microsserviços, Mensageria, Classificação de amostras, Comunicação, Aprendizado contínuo.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

IDS	Intrusion Detection System
NIDS	Network Based Intrusion Detection System
ML	Machine Learning
DMS	Data Monitoring Server
IOT	Internet of Things
VP	Verdadeiro Positivo
VN	Verdadeiro Negativo
FP	Falso Positivo
FN	Falso Negativo

Sumário

1 Introdução

A cada dia, a tecnologia se torna mais importante para a sociedade. Sua utilização está em constante crescimento por pessoas e empresas em todo o mundo, seja em grandes ou pequenas escalas. Embora esse processo tenha trazido uma série de benefícios, inúmeras vulnerabilidades têm surgido. A segurança cibernética de pessoas e dispositivos tem sido colocada em risco à medida que as atividades maliciosas têm ocorrido com cada vez mais frequência (LABS, 2024). Isso pode gerar diversos prejuízos, tais como o comprometimento de informações sigilosas relevantes para o contexto social, sequestro de dados e ataques à disponibilidade dos sistemas computacionais, entre outros. Segundo destacado pelo canal de notícias CNN Brasil, uma pesquisa realizada pela empresa Fortinet mostrou que o número de tentativas de ataques cibernéticos a empresas foi de aproximadamente 31,5 bilhões no primeiro semestre de 2022, o que é praticamente o dobro do registrado nesse mesmo período de 2021 (BRASIL, 2022). A Fortinet ainda informou que esses ataques têm se tornado cada vez mais sofisticados, e que quando os invasores tem sucesso na tentativa de ataque, costumam pedir pagamentos elevados em troca da devolução dos dados (OLIVEIRA, 2022). Em 2024, a CNN exibiu uma nova pesquisa, realizada pela empresa ESET, que mostra que o Brasil, no primeiro semestre de 2024, foi o 4º país da América Latina com mais ameaças cibernéticas detectadas, totalizando uma média de 201 mil, e esses números tendem a crescer ainda mais (BRASIL, 2024).

Dado o crescimento constante no número de ataques cibernéticos, surge a necessidade do uso de técnicas para garantir a segurança das redes e sistemas computacionais (DocuSign, 2023). Nesse contexto, os Sistemas de Detecção de Intrusões, do inglês, *Intrusion Detection Systems* (IDSs) possuem a função de analisar e identificar as atividades maliciosas nos sistemas de computadores e redes computacionais. Existem diferentes categorias de IDSs (QUINCOZES et al., 2021). Em particular, os IDSs baseados em aprendizado supervisionado utilizam um conjunto de dados de treinamento, que permite a construção de modelos para a classificação de amostras desconhecidas através do uso de algoritmos classificadores. Tais algoritmos são um tipo de algoritmo de aprendizado de máquina, do inglês, *Machine Learning*, e atuam investigando diferentes fontes de dados para obter maior assertividade na detecção de intrusões em redes e sistemas computacionais. Ao encontrar uma atividade suspeita, são tomadas as medidas necessárias a fim de reduzir ou evitar os danos aos alvos de ataques cibernéticos (DUA et al., 2019).

Apesar da relevância de IDSs para a segurança de redes e aplicações, ainda existem alguns desafios que podem prejudicar sua efetividade. O desempenho dos classificadores pode variar de acordo com os padrões de ataque. Isso significa que para contextos distintos, existem diferentes algoritmos que podem alcançar uma performance adequada na

detecção de intrusões. Dessa forma, não existe um único classificador capaz de detectar todo o tipo de padrão com perfeição. Uma abordagem para contornar esse problema é o desenvolvimento de IDSs baseados em múltiplos classificadores. No entanto, existem casos onde os diferentes classificadores de tais IDSs obtêm resultados conflitantes entre si, dificultando a tomada de decisão final. Além disso, o mesmo problema pode ocorrer quando o conjunto de dados usado para a construção do modelo de predição é insuficiente ou pouco representativo para determinados tipos de ataques (QUINCOZES et al., 2021).

Uma alternativa para contornar este problema é o uso das chamadas Redes de Conselhos (QUINCOZES et al., 2021). As Redes de Conselhos consistem em um grupo de IDSs interconectados que se comunicam e colaboram para solucionar conflitos de divergência de resultados de classificadores ou falta de conhecimento de um dos IDSs acerca de determinado padrão de ataque. Através desse método, quando um IDS analisa uma atividade e não consegue tomar uma decisão, seja por conflito entre classificadores precisos/confiáveis ou pela falta de precisão dos classificadores para aquele padrão, o mesmo pode consultar a rede de conselheiros. Assim, um IDS com maior nível de confiança na classificação do respectivo padrão transmite um conselho baseado em seus classificadores experientes. Como resultado, mais fontes de informações são utilizadas para a tomada de decisões mais assertivas. Contudo, apesar do aumento na quantidade de informação disponível, bem como o auxílio de IDSs terceiros, a atual abordagem empregada na Rede de Conselhos tem duas principais limitações: i) comunicação entre pares através do paradigma requisição e resposta, que gera dependência do IDS conhecer seu par; ii) ausência de mecanismos para o aprendizado passivo de IDSs a partir de conselhos trocados entre seus pares.

1.1 Objetivos

O objetivo desse trabalho consiste em desenvolver uma arquitetura distribuída para detecção de intrusões através da Rede de Conselhos de técnicas de aprendizado contínuo, utilizando de estratégias que aprimorem a comunicação de diversos IDSs e permitem que eles comuniquem entre si de forma mais segura e eficaz, sem comprometer sua análise e desempenho. Com isso, a partir da atualização do conjunto de amostras conhecidas e com base nas experiências compartilhadas com outros IDSs, pretende-se ajustar seus modelos de detecção conforme novos padrões de ataque são encontrados e compartilhados. Os objetivos específicos deste trabalho são listados a seguir:

- Implementar uma Rede de Conselhos através de uma arquitetura orientada à microsserviços para a troca de conselhos entre IDSs distribuídos.
- Estabelecer uma arquitetura base para o aprendizado passivo entre IDSs, usando o modelo publicação/assinatura (pub/sub) para configurar diferentes tópicos de

interesse em assinaturas de ataques.

- Avaliar a eficiência da Rede de Conselhos em IDSs configurados com bases de conhecimento que envolvem padrões de ataques distintos.

1.2 Justificativa

Mesmo com a implementação das Redes de Conselhos para solucionar o problema de resultados conflitantes entre os classificadores de IDSs, além da possível insuficiência ou baixa representatividade do conjunto de dados utilizado na construção do modelo de predição, ainda há a necessidade de se aprimorar ainda mais esse processo. A Rede de Conselhos, em sua arquitetura tradicional, opera de forma monolítica. Nessa estrutura distribuída, as atividades realizadas pelos IDSs ocorrem de forma dependente e sequencial, o que gera limitações à escalabilidade e, sobretudo, à agilidade do fluxo como um todo. Com a implementação de uma arquitetura baseada em microsserviços, cada IDS se torna independente um do outro, e realiza suas funções de forma assíncrona, tornando o sistema mais rápido e eficaz.

Para que a comunicação entre esses IDS, agora independentes, ocorra de forma eficaz e responsiva, a utilização de uma estrutura de mensageria se faz fundamental. Ela garante que a solicitação e envio de conselhos seja feita de forma fluida, tendo em vista que as mensagens são publicadas em tópicos de acordo com sua categoria, e os assinantes recebem de forma quase instantânea, podendo processá-la e realizar as ações necessárias de acordo com a lógica do fluxo. Além disso, os conselhos recebidos se tornam parte do conjunto de dados de treinamento do IDS que os solicitou, promovendo seu aprendizado contínuo e aprimorando a assertividade de classificação e detecção de intrusões.

2 Referencial Teórico

Este capítulo tem como objetivo apresentar alguns dos conceitos que são fundamentais para o entendimento desse trabalho. Além disso, também serão apresentados os trabalhos da literatura que, de alguma forma, foram relevantes para este desenvolvimento.

2.1 Conceitos Básicos

Para entender melhor sobre a abordagem desse trabalho, é preciso conhecer melhor sobre os conceitos de Rede de Conselhos, tais como sua origem, seu propósito, entre outros. Além disso, para implementar a aprendizagem contínua no IDS por meio da Rede de Conselhos, também é necessário conhecer quais são as melhores estratégias para isso, uma vez que a abordagem de aprendizado contínuo visa fazer com que o IDS tenha um desempenho de alto nível, por meio do reconhecimento de padrões de conexão de rede para conhecer os tipos de intrusão e retreinar seus métodos. Portanto, também é necessário analisar essas técnicas sob o contexto de IDSs, para observar suas vantagens e desvantagens.

Sendo assim, este capítulo irá discorrer sobre a definição e composição da Rede de Conselhos, na Seção 2.1.1, e as técnicas de aprendizado de máquina, na Seção 2.1.2.

2.1.1 Rede de Conselhos

De acordo com a proposta apresentada pelo artigo *Counselors network for intrusion detection* (QUINCOZES et al., 2021), a rede de conselhos consiste em um grupo de IDSs que se comunicam e colaboram para solucionar conflitos de divergência de resultados de seus múltiplos classificadores. Através desse método, quando um IDS analisa uma atividade e não consegue distinguir se é suspeita ou não, seja por conflito de saídas de classificadores precisos, ou saídas duvidosas de classificadores imprecisos, pode consultar a rede de conselheiros, onde receberá aconselhamento de classificadores experientes que o ajudarão a tomar uma decisão final. Como resultado, a gama de dados analisados aumenta e o processo de verificação e detecção de intrusão se torna mais preciso.

Cada conselheiro tem a função de atender às solicitações de aconselhamento na rede, através da sua experiência adquiria da sua base de conhecimento. Para que a precisão do conselho seja maior, os detectores conselheiros utilizam de fontes de dados heterogêneas que investigam a mesma classe de ataques, inclusive a precisão obtida pelo conselheiro na detecção de amostras semelhantes é fundamental para se definir o quão confiável ele é.

Como resultado disso, cada solicitação de aconselhamento deve ter como resposta o histórico de precisão do conselheiro consultado e o resultado obtido por ele. Isso permite que o detector solicitante tenha o direito de aceitar ou ignorar o conselho recebido caso considere que o conselheiro que respondeu não possua um bom histórico de precisão sobre amostras semelhantes. Caso ele opte por ignorar o conselho, ele pode consultar outros conselheiros ou tomar outras medidas para evitar que sejam disparados alarmes falsos.

Com esta abordagem, o principal objetivo é combinar a utilização de várias fontes de dados, com a integração de múltiplos classificadores, implementando a autoaprendizagem destes classificadores e minimizando a intervenção humana, visto que a autoaprendizagem ocorre através da atualização do conjunto de dados de treino com base no conselhos recebidos de detectores especializados sobre vários tipos de ataques.

2.1.2 Tipos de Aprendizado de Máquina

A abordagem de aprendizado contínuo visa fazer com que o IDS tenha um desempenho de alto nível, por meio do reconhecimento de padrões de conexão de rede para conhecer os tipos de intrusão e retreinar seus métodos. Esse assunto pode ser dividido em categorias de aprendizado, conforme abordadas nesse capítulo.

O aprendizado supervisionado consiste em uma técnica em que os dados possuem rótulos bem definidos, nos quais o modelo se baseia e, a partir da investigação de diversos dados semelhantes, vai aumentando sua assertividade de previsão. Essa categoria de aprendizado também pode ser dividida em outras duas vertentes, como regressão, onde as saídas consistem em valores específicos de dados, e classificação, onde a saída consiste em uma classe de rótulos. Utilizando desses rótulos pré definidos, permite ao IDS uma precisão mais assertiva na detecção de ataques semelhantes, mas o torna limitado quanto a capacidade de detectar ataques que ainda não estão catalogados (CUNNINGHAM; CORD; DELANY, 2008).

Já o aprendizado não supervisionado não possui resultados pré definidos para que possa se basear, pelo contrário, ele busca encontrar esses rótulos. Sua função é encontrar a relação entre um conjunto de dados para, a partir disso, agrupar esses dados em diversas categorias, tudo isso a partir de observações que ele mesmo faz. Através dessa função, esse aprendizado permite que o IDS consiga investigar e detectar diversos tipos de ataques. No entanto, a falta desse parâmetro prévio faz com que o IDS esteja sujeito a um problema do falso positivo, ou seja, avalia como ataque uma atividade que talvez não seja maliciosa ou prejudicial ao sistema (CHOI et al., 2019).

O aprendizado por reforço consiste em um aprendizado baseado em recompensa, ou seja, a máquina executa uma determinada atividade e a medida que ele realiza corretamente ele é recompensado de forma positiva, e quando ele a executa incorretamente, não é

recompensado, ou pode receber uma recompensa negativa. Com base nessas recompensas, ele começa a aprender a melhor forma de se comportar em determinada situação. Esse aprendizado também possui uma grande eficácia, visto que o IDS pode ser recompensado ao executar algo corretamente, e uma dessas recompensas pode ser a auto calibração de seu conjunto de dados de treinamento para que ele aprenda ainda mais. Entretanto, essa abordagem também pode ter suas desvantagens, como por exemplo a possibilidade do IDS aprender errado, ou seja, entender que uma atividade ser maliciosa mesmo não sendo, o que pode acarretar em uma atualização errônea do seu conjunto de dados de treinamento e, com isso, uma perda em sua eficiência na detecção de intrusões (SILVA, 2023).

O aprendizado colaborativo, por sua vez, consiste no compartilhamento de informações entre múltiplos agentes. Essas informações podem incluir rótulos de dados, por exemplo, ou resultados de previsões, métricas obtidas, entre outros. Isso permite que os sistemas compartilhem conhecimento e aprendam mutuamente. A vantagem desse aprendizado dentro do contexto de IDS é aumentar a precisão e eficácia do sistema em tomar decisões na classificação de amostras. No entanto, uma desvantagem é o fato de que um dos agentes pode colaborar com informações imprecisas, comprometendo o aprendizado e desempenho dos agentes (AYALA; YANO, 1998).

No contexto deste trabalho, são utilizados dois tipos de aprendizado. Na fase de treinamento dos classificadores, é utilizado o aprendizado supervisionado, pois essa etapa envolve o processamento de um conjunto de dados rotulado. Isso permite que o modelo classifique futuras amostras com base nos padrões aprendidos.

Além disso, a troca de conselhos para as amostras que os classificadores de um determinado agente não conseguiram rotular se enquadra no aprendizado colaborativo, no qual a Rede de Conselhos permite essa troca de informações entre os agentes, para que eles recebam sugestões de rótulos e ampliem seus próprios conjuntos de dados com tal aprendizado.

2.2 Trabalhos Relacionados

Na literatura, o tema aprendizado contínuo para IDSs já foi abordado por diversos autores, cada um apresentando características e resultados específicos. O artigo *Survey of intrusion detection systems: techniques, datasets and challenges* (KHRAISAT et al., 2019) apresenta uma visão geral e abrangente de diversos trabalhos utilizados para essa finalidade. Além disso, aborda os principais tipos de sistemas de detecção de intrusão, suas técnicas, vantagens e desvantagens, o que auxilia bastante para se obter um contexto melhor sobre essa área.

A abordagem de adaptação do *Deep Reinforcement Learning* (DRL) (LOPEZ-MARTIN; CARRO, 2020), utilizando conjuntos de dados previamente definidos para

treinar alguns algoritmos de aprendizado de máquina apresentou bons resultados principalmente para o algoritmo *Deep Q-Network*. Nessa abordagem, é criado um ambiente adaptado composto por uma amostragem de intrusões de treinamento gravadas, o qual irá gerar recompensas a partir dos erros de detecção que forem encontrados durante o treinamento, fazendo com que os resultados de detecção de intrusão sejam mais precisos.

Os autores Martina e Foresti (2021) trouxeram a implementação de um novo IDS, chamado SF-SOINN (*Soft-Forgetting Self-Organizing Incremental Neural Network*), que fornece capacidade de aprendizado contínuo, executa classificações rápidas, é robusto a ruídos, e obtém uma boa performance comparado a outras abordagens já existentes, visto que consegue se auto-organizar adotando um processo de exclusão de nós.

O artigo *A Deep Reinforcement Learning Approach for Anomaly Network Intrusion Detection System* (HSU; MATSUOKA, 2020) propõe a implementação de um sistema de detecção de intrusão de rede baseado em anomalia (ANIDS) com aprendizado por reforço profundo, com dois modos de funcionamento que alternam entre si de acordo com a necessidade. O sistema utiliza o modo de detecção, porém quando seu desempenho na detecção de anomalias estiver abaixo do limite predefinido, ele alterna para o modo de aprendizado, buscando aprender novos padrões de tráfego de rede e, com isso, poder voltar para o modo de detecção, já com um desempenho melhor. O sistema processa variáveis oriundas do processamento de dados brutos de tráfego de rede e fornece uma ação. Sua recompensa é calculada com base no rótulo e na ação que ele propôs e com base nisso o agente atualiza seu modelo.

No trabalho (JAVADPOUR et al., 2023), é proposto o DMAIDPS, um sistema de detecção e prevenção de intrusão multiagente e distribuído, projetado para ambientes de Internet das Coisas (IoT) na nuvem. Esse sistema enfrenta desafios devido à centralização característica da nuvem, o que limita a comunicação direta e a colaboração entre os sistemas de detecção de intrusão em rede (NIDS). Essa restrição pode impactar a eficiência da resposta a ameaças em redes complexas e interconectadas, evidenciando a necessidade de abordagens que facilitem a cooperação entre agentes de segurança.

Em 2023, foi proposta uma abordagem com certa semelhança em relação à Rede de Conselhos. Trata-se de uma rede colaborativa de NIDS para integração de redes IoT (PEDROSO et al., 2023). Esse modelo é composto por domínios de IoT, chamados de "modelos de ilha", e cada um desses domínios possui um servidor de monitoramento de dados (DMS), um gateway e um NIDS. Cada dispositivo IoT coleta dados e os enviam ao DMS. Sempre que um NIDS detecta uma anomalia, envia mensagens para alertar os outros NIDS da rede, para que atualizem seu banco de dados de regra. Com essa colaboração entre os NIDS, tem-se uma uma visão completa dos ambientes monitorados e regras atualizadas para detectar novas anomalias.

3 Desenvolvimento

Neste capítulo, serão abordados os métodos utilizados para modelar, organizar e implementar o novo modelo de Rede de Conselhos. Além disso, serão apresentados a arquitetura utilizada para comunicação dos microsserviços, os algoritmos de cada etapa de funcionamento, métodos para aprendizado contínuo dos detectores, entre outros.

3.1 Metodologia

Como mencionado na Seção 1.2, este trabalho tem como objetivo a implementação da nova estrutura da Rede de Conselhos de forma distribuída, por meio de microsserviços. Esses microsserviços serão implementados na linguagem Java (na versão 19) com o *framework* Spring Boot. Portanto, se faz necessário a criação de um serviço independente para cada um dos IDSs que a compõem, para que eles possam atuar de forma autônoma de acordo com o papel que desempenham nela.

A diferença de estruturação entre cada IDSs se dá apenas pela função que cada um possui dentro da rede. Desse modo, para garantir uma melhor escalabilidade do sistema, além de facilitar a manutenção do código, as tarefas serão distribuídas em métodos específicos, organizados em classes bem definidas e objetivas, garantindo uma organização modular e funcional.

Cada IDS possui seus próprios conjuntos de dados para a construção de modelos de predição dos classificadores, bem como para sua avaliação. Como parte das melhorias deste trabalho, essas fases, tais como as demais etapas do fluxo, estão contidas na organização modular do projeto, permitindo a abstração de métricas específicas para cada uma delas, como dados para análise de performance e tempo de execução.

Portanto, a fim de permitir uma comunicação fluida e escalável entre esses microsserviços, será adotado o sistema de mensageria, do inglês, *Message Broker* do Apache Kafka. Com ele, a comunicação entre o IDS que está solicitando conselho e os IDSs conselheiros se dá através dos paradigmas de publicação e assinatura, também conhecidos pelo termo inglês *publish/subscribe*, ou apenas pub/sub. Os conselheiros consomem as mensagens de um tópico específico para comunicação de conselhos, que são geradas e publicadas pelo IDS que, no decorrer da sua análise do conjunto de dados, encontrou situações de conflito.

Por fim, o IDS passa por uma etapa de aprendizado contínuo baseado em lotes após os aconselhamentos da Rede de Conselhos, consumindo cada conselho recebido e realizando a auto-calibração das *features* do conjunto de dados de treinamento. Com isso,

a colaboração da rede permite que o IDS esteja em constante aprendizado, tendo maior assertividade na detecção de intrusões.

Conforme introduzido no capítulo 1, uma abordagem muito utilizada para detecção de intrusões é a de múltiplos classificadores, dado que não há nenhum classificador eficaz o suficiente para detectar todo tipo de padrão de ataque com perfeição. Por isso, neste trabalho, são utilizados IDS que possuem múltiplos classificadores, os quais são o J48, Random Tree, Random Forest, Naive Bayes e Rep Tree,

Durante as etapas principais do fluxo dessa nova estrutura de Rede de Conselhos, são utilizadas algumas métricas para análise do desempenho dos classificadores. Dentre elas estão a acurácia, precisão, revogação, e *F1-Score*, todas calculadas a partir das métricas Verdadeiro Positivo (VP), Verdadeiro Negativo (VN), Falso Positivo (FP) e Falso Negativo (FN).

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \tag{3.1}$$

A acurácia tem o papel de calcular a razão do número de amostras classificadas corretamente em relação ao número total de amostras do conjunto de dados (CHICCO; JURMAN, 2020).

$$Revogação = \frac{VP}{VP + FN} \tag{3.2}$$

Revogação, do inglês, *recall*, mede a proporção de amostras classificadas corretamente como positivas em relação ao total de amostras positivas (RAJU et al., 2020).

$$Precisão = \frac{VP}{VP + FP} \tag{3.3}$$

A precisão, também conhecida pelo termo inglês *precision*, indica a proporção de previsões positivas corretas em relação ao total de previsões que são, de fato, positivas (JIAO; DU, 2016).

$$F1\text{-}Score = 2 \times \frac{Precisão \times Revogação}{Precisão + Revogação} \tag{3.4}$$

F1-Score é uma métrica muito usada no cenário de aprendizado de máquina, com o objetivo de avaliar o desempenho dos classificadores em detectar e classificar classes de amostras. Essa métrica tem como objetivo equilibrar as métricas de revogação e precisão, e considera tanto os falsos positivos quanto os falsos negativos. (SIBLINI et al., 2020).

3.2 Arquitetura da Rede de Conselhos

Neste capítulo, encontra-se a arquitetura original da Rede de Conselhos (QUINCOZES et al., 2019), cujo funcionamento foi explanado na Seção 2.1.1. Cada IDS realiza as etapas de treinamento e avaliação em paralelo e de forma *off-line*. A etapa de classificação de amostras desconhecidas ocorre durante a fase de detecção, na qual os conflitos tendem a ocorrer.

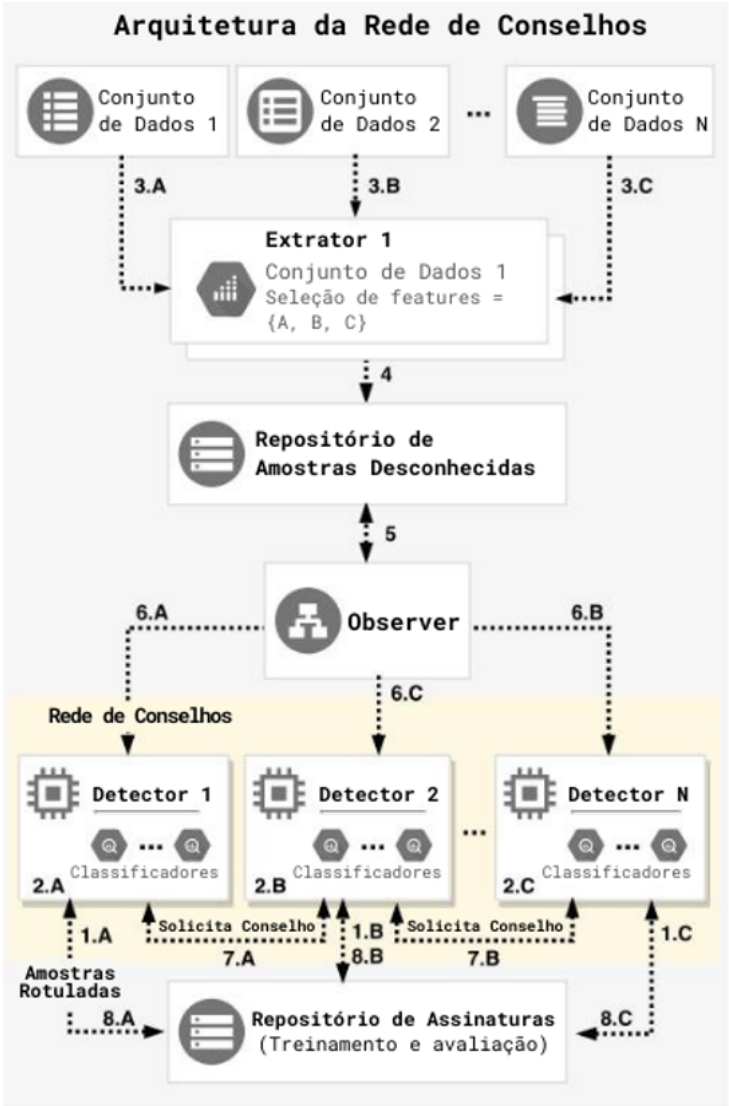

Figura 1 – Arquitetura da Rede de Conselhos (Fonte: Extraído de (QUINCOZES et al., 2019)).

Os detectores conselheiros realizam a assinatura do tópico de troca de conselhos, denominado *ADVICE_TOPIC* logo após concluir as tarefas de treinamento e avaliação. Dessa forma, recebem em tempo real as solicitações de conselhos realizadas pelo terceiro detector, que publica durante a etapa de detecção e, em seguida, também assina o tópico para iniciar a etapa de consumo e aprendizado com os conselhos.

3.3 Algoritmo de Treinamento e Avaliação

Como abordado no tópico anterior, o procedimento de construção do modelo de classificação ocorre em duas etapas, sendo elas o treinamento e avaliação dos classificadores. Ambas utilizam seus respectivos conjuntos de dados, com proporções relativas à quantidade de amostras e as classes existentes. A finalidade dessas etapas é fazer com que os classificadores adquiram uma base de conhecimento através do treinamento e, na avaliação, são mensuradas sua precisão e confiança para classificar futuras amostras desconhecidas. No Algoritmo 1 é possível observar a condução desse estágio.

Algoritmo 1: Etapas de Treinamento e Avaliacão de Classificadores

```
Entrada: k // Número de clusters
         1 datasetTreino // instâncias de treinamento
         2 datasetAvaliacao // instancias de avaliacão
         3 a // critério de selecão
Saída  : Classificadores selecionados para cada cluster

// Etapa de Treinamento
Para cada Classificador c em Classificadores faca
    c ← buildClassifier(datasetTreino);
fim

// Etapa de Avaliacao
clusters ← K-means(k, datasetAvaliacao);
Para cada cluster em clusters faca
    clusterInstances ← obtemInstancias(cluster);
    Para cada Classificador c em Classificadores faca
        Para cada ClusteredInstances i faca
            saida ← classify(i);
            atualizaAcuracia(c, saida);
            atualizaF1Score(c, saida);
        fim
    fim
    Para cada Classificador c em Classificadores faca
        // Seleciona se tem boa acurácia e F1-Score
        classificadoresSelecionados ← classifierSelection(c, a);
    fim
    cluster ← classificadoresSelecionados;
fim
```

Conforme abordado na Seção 2.1.1, a Rede de Conselhos utiliza de múltiplos classificadores. Neste trabalho, são utilizados os classificadores *J48*, *Random Tree*, *Random Forest*, *Naive Bayes* e *Rep Tree*, e a decisão final de cada classificação é tomada com base na combinação de suas saídas. Por esse motivo, existe a possibilidade de se ocorrer conflitos, visto que essas saídas podem divergir entre si.

Na etapa de treinamento, conforme descrito no Algoritmo 1, o modelo desses classificadores é construído através do conjunto de dados de treinamento. Em seguida, o conjunto de dados de avaliação é dividido, e os dados semelhantes são agrupados em *clusters*, cuja quantidade é definida previamente, através do algoritmo *K-means*. Em cada *clus-*

ter é feita a avaliação dos classificadores, determinando-se os classificadores com melhor desempenho para cada classe de ataques.

Ao final dessa etapa, é retornado, para cada *cluster*, a lista dos classificadores selecionados, que serão utilizados na etapa de detecção para classificar as amostras desconhecidas.

3.4 Algoritmo de Detecção

Na etapa de detecção, com os classificadores já treinados e avaliados, é iniciado o processo de classificação de amostras desconhecidas. Essa etapa já ocorre de forma *on-line*, ou seja, marca o início da comunicação da Rede de Conselhos.

Algoritmo 2: Etapa de Detecção

```
Entrada: centroids // centroides dos k clusters
         1 datasetDeteccao // instâncias desconhecidas para deteccao
Saída  : Classificacão da amostra desconhecida ou solicitacão de conselho

// Etapa de Deteccao
Para cada instancia em datasetDeteccao faca
    centroids ← K-means(instancia, datasetDeteccao);
    classificadoresSelecionados ← getSelectedClassifiers(centroids);
    Se classificadoresSelecionados == 0 entao
        RedeConselhos ← RequestAdvice(instancia);
        // Atividade da rede de conselhos
    fim
    Senao
        Para cada classificador em classificadoresSelecionados faca
            Class ← classify(instancia, classificador);
        fim
        Se Conflict() entao
            RedeConselhos ← RequestAdvice(instancia); // Atividade da rede de
              conselhos
        fim
    fim
fim
```

Conforme retratado no Algoritmo 2, o fluxo inicia-se extraindo um conjunto de dados contendo amostras desconhecidas. Para cada amostra, o algoritmo *K-means* é aplicado para identificar, entre os *clusters* existentes, o *cluster* que possui dados semelhantes à amostra. Após esse procedimento, a lista dos classificadores selecionados para esse *cluster* é retornada.

A partir da lista de classificadores retornada, são realizadas duas validações. Caso a lista esteja vazia, entende-se que não há classificadores capacitados o suficiente para realizar a classificação dessa amostra, tornando-se necessário recorrer à Rede de Conselhos

para solicitar conselho dos outros detectores. Se a lista possuir um ou mais classificadores selecionados, obtém-se o resultado da classificação de cada para a amostra. Se todos os classificadores retornarem o mesmo resultado, o IDS toma a decisão final. No entanto, se os resultados dos classificadores forem divergentes, tem-se um outro cenário de conflito, onde o IDS não consegue tomar a decisão final, sendo necessário, também, enviar a solicitação para a rede.

3.5 Envio de Requisição de Conselho

Para qualquer cenário que ocasione um conflito ao IDS, a tarefa principal da Rede de Conselhos é iniciada, onde o detector publica uma mensagem no tópico *ADVICE_TOPIC* do *broker* Kafka solicitando um conselho. Essa mensagem consiste em um *JSON* conforme representado na Figura 2.


```
{
  "id_conselheiro": "2",
  "id_sample": 176,
  "flag": "REQUEST_ADVICE",
  "sample": [2938.8389,-0.14082587,-0.0043433895,0.1178,-0.053885963,0.09623301,-0.042262208,
      0.1079050299792793,0.0458042692242843,0.0766139564220406,0.0680703784880698,0.0676218345718064,
      0.0679700659193829,-40.06914142757887,-38.99351293665677,-39.109851208369946,-39.48049923306098,
      -39.53601344896015,-39.484397868218366,2934.60631,2938.8389170470123,9,329,0,200,16,16,2,11000,
      1,186,1,2,0,2,1,1,1,25,186,1,0,0,0,0,0,0,0,0,4.232607047012152,0.00004985951227354235
  ],
  "timestamp": "2024-11-01T14:30:00.000Z"
}
```


Figura 2 – Mensagem de solicitação de conselho

A fim de facilitar a comunicação e troca dos conselhos na rede, o *JSON* contém informações objetivas para que os conselheiros consigam processar e enviar seus respectivos resultados. Entre essas informações estão um identificador do detector solicitante, o identificador da amostra na qual ocorreu o conflito, representado pelo seu índice no conjunto de dados de detecção, uma *flag* para indicar que a mensagem que está sendo publicada é do tipo *REQUEST_ADVICE*, a amostra completa e o seu *timestamp*, que informa o momento exato em que o conflito foi identificado.

Por meio da implementação da Rede de Conselhos de forma distribuída com microsserviços, bem como a comunicação entre eles realizada através da estrutura de mensageria com o Kafka, o IDS publica a requisição de conselho e continua a análise das amostras seguintes, sem a necessidade de interromper a tarefa de detecção para aguardar o processamento e respostas do conselheiros. Com isso, ele percorre todo o conjunto de dados de detecção, enviando requisições de conselho à medida que se depara com cenários conflitantes. Em paralelo a isso, os conselheiros iniciam o processo de processamento e geração de conselho.

3.6 Algoritmo Generate Advice

Os detectores que atuam na Rede de Conselhos com o papel de conselheiros, ao concluir as etapas de treinamento e avaliação dos classificadores, realizam a assinatura do tópico *ADVICE_TOPIC* e aguardam a publicação de solicitações de conselhos. Quando essa requisição é realizada, os IDSs conselheiros iniciam o procedimento de geração do conselho, representado pelo Algoritmo *Generate Advice*.

Algoritmo 3: GENERATE ADVICE

```
Entrada: novaInstancia // Instancia a partir da amostra recebida
         1 datasetAvaliacao // instancias de avaliacão
Saída  : Conselho e F1-Score dos classificadores

// Processamento da amostra
Para cada novaInstancia em datasetAvaliacao faca
    novaInstancia ← DenseInstance(amostra);
    cluster ← K-means(novaInstancia);
    classificadoresSelecionados ← getSelectedClassifiers(cluster);
    Para cada classificador em classificadoresSelecionados faca
        Class ← classify(novaInstancia, classificador);
        Se classificador.getEvaluationF1Score() > bestF1Score entao
            bestResult ← result;
            bestF1Score ← classificador.getEvaluationF1Score();
        fim
    fim
    RedeConselhos ← ResponseAdvice(bestResult); // Publica o melhor resultado
     obtido
fim
```

Inicialmente, uma nova instância é criada através da amostra extraída da mensagem de solicitação de conselho e o *K-means* identifica o *cluster* correspondente para ela. A partir dessa identificação, os classificadores selecionados dentro desse *cluster* são retornados e inicia-se o processo de classificação da instância.

Em seguida, é realizada uma comparação entre os resultados obtidos pelos classificadores, e o conselho a ser publicado é determinado pelo resultado do classificador que apresentar o maior *F1-Score*, ou seja, considerado o de maior confiabilidade.

3.7 Envio de Conselho

Após processar a amostra enviada na requisição de conselho e obter o melhor resultado entre os classificadores, os IDS conselheiros realizam a publicação de uma mensagem do tipo *RESPONSE_ADVICE*, também no tópico *ADVICE_TOPIC*. A composição dessa mensagem contém os mesmos parâmetros das mensagens do tipo *RE-*

QUEST_ADVICE, demonstrada na Seção 3.5, com o acréscimo dos parâmetros *F1-Score* e *result*, que são os dados que o IDS solicitante necessita para escolher o conselho na qual irá aprender, conforme será abordado na Seção 3.8.


```
{
    "id_conselheiro": "1",
    "id_sample": 176,
    "flag": "RESPONSE_ADVICE",
    "sample": [2938.8389,-0.14082587,-0.0043433895,0.1178,-0.053885963,0.09623301,-0.042262208,
        0.1079050299792793,0.0458042692242843,0.0766139564220406,0.0680703784880698,0.0676218345718064,
        0.0679700659193829,-40.06914142757887,-38.99351293665677,-39.109851208369946,-39.48049923306098,
        -39.53601344896015,-39.484397868218366,2934.60631,2938.8389170470123,9,329,0,200,16,16,2,11000,
        1,186,1,2,0,2,1,1,1,25,186,1,0,0,0,0,0,0,0,0,4.232607047012152,0.00004985951227354235
    ],
    "f1score": 96.90721893310547,
    "result": 0,
    "timestamp": "2024-11-01T14:31:00.000Z"
}
```


Figura 3 – Mensagem de envio de conselho

Conforme retratado pela Figura 3, o *JSON* enviado como resposta do conselho, contém o identificador do conselheiro e da amostra, uma *flag* que sinaliza o tipo da mensagem como sendo uma resposta de conselho, além do *F1-Score* dos classificadores empregados para classificar essa amostra e o seu resultado. Esse resultado, por sua vez, é representado por um número que representa o valor da classe de ataque. Esses parâmetros, juntamente com a amostra, identificada como *sample*, possibilitam ao IDS solicitante o processamento do conselho e acompanhamento do rendimento de cada conselheiro.

3.8 Aprendizado com os Conselhos

Enquanto realiza a etapa de detecção, abordada na Seção 3.4, o IDS envia as solicitações de conselho sempre que encontra um cenário de conflito, e continua a classificação das amostras subsequentes. Ao concluir essa tarefa, ele assina o tópico *ADVICE_TOPIC* para consumir os conselhos publicados pelos outros IDSs.

Durante essa etapa, o IDS solicitante armazena um *cache* para cada amostra recebida dos conselhos, contendo como chave o ID dessa amostra e, como valor, uma lista com as mensagens que possuam esse identificador. Quando essa lista atinge o tamanho correspondente ao número máximo de conselhos para uma determinada amostra, o critério de parada para o armazenamento de *cache* dessa amostra é acionado. Em seguida, realiza-se a comparação entre os *F1-Score* de cada conselho a fim de que o IDS solicitante escolha o maior e o utilize na decisão final da classificação da amostra. Após esse processo, a amostra recebe, como rótulo, o resultado enviado pelo conselho e é adicionada ao conjunto de dados de treinamento. Esse processo é exemplificado por meio do Algoritmo 4.

Algoritmo 4: CONSUMO E SELECAO DE CONSELHOS

```
Entrada: ADVICE_TOPIC // Topico de compartilhamento de conselhos
         1 datasetTreino // instancias de treinamento
         2 criterioParada ← maxConselhos // numero de conselheiros
Saída  : Amostra rotulada e adicionada ao conjunto de dados de treinamento

// Assina topico para consumir os conselhos
3  assinar(ADVICE_TOPIC);
4  Para cada mensagem em ADVICE_TOPIC faca
       // verifica se a amostra ja foi consumida
5      Se mensagem.getIdSample() nao processada entao
6          idAmostra ← mensagem.getIdSample();
7          armazenaCache(idAmostra, mensagem);
8          Se tamanhoCache(idAmostra) == criterioParada entao
9              resultAmostra ← getBestAdvice(idAmostra); // escolhe o melhor conselho
               // alimenta a base com a amostra rotulada
10             datasetTreino.add(amostra, resultAmostra);
11         fim
12     fim
13 fim
```

Após a conclusão das etapas de processamento e seleção dos conselhos que irá realizar o aprendizado, o IDS solicitante passa a ter um novo conjunto de dados de treinamento, contendo as amostras iniciais somadas às que foram obtidas através dessas etapas. A partir dessa nova base de conhecimento, torna-se necessário repetir as etapas de treinamento e avaliação, abordadas na Seção 3.3, com o objetivo de atualizar o modelo de classificação do IDS e aprimorar sua capacidade de detecção.

3.9 Implementação

Para avaliar a nova abordagem deste trabalho, foi proposto um novo modelo de Rede de Conselhos, estendido da implementação tradicional, utilizando microsserviços para organizar a estrutura de forma distribuída. Neste trabalho, foram implementados três microsserviços, denominados IDS 1, IDS 2 e IDS 3, nos quais os IDS 1 e IDS 3 atuam como conselheiros na rede, e, por isso, executam apenas as tarefas de construção do modelos e avaliação dos mesmos. O IDS 2, por sua vez, além de executar ambas as etapas, também realiza a etapa de detecção para classificar as amostras e acionar o fluxo da Rede de Conselhos conforme necessário. Para se realizar essa implementação, foi necessária a utilização das ferramentas Spring Boot, Kafka e Docker.

O Spring Boot é um *framework* da linguagem de programação Java que facilita o processo de desenvolvimento de microsserviços. Sua utilização permite simplificar o gerenciamento de dependências e métricas, além da promover a configuração de servidores e bibliotecas de forma otimizada (REDDY, 2017).

Para garantir a eficiência do fluxo de trabalho e execução dos microsserviços e do

Apache Kafka, cujo paradigmas foram abordados na Seção 3.1, foi empregada a utilização dos contêineres do Docker, uma vez que essa ferramenta possui a característica de conter apenas os dados que a aplicação realmente precisa, além de oferecer um ambiente mais compacto em relação às máquinas virtuais (ANDERSON, 2015).

Antes de iniciar a execução dos fluxos de quaisquer microsserviços (IDS 1, IDS 2 e IDS 3) é necessário garantir que o ambiente do Kafka esteja inicializado no Docker. Para isso, executa-se o comando *docker-compose up*. Dessa forma, é assegurado que os contêineres do *Zookeeper*, que gerencia o *cluster* do Kafka, o *broker* do próprio Kafka, e o *Kafdrop*, que permite a visualização das atividades dentro desse *broker*, estejam funcionais para a execução dos microsserviços.

3.10 Distribuição dos Conjuntos de Dados

Como premissa desta implementação, foram distribuídos entre os microsserviços os seus respectivos conjuntos de dados, cada um contendo uma composição de amostras para cada classe de ataque. Para a geração desses conjuntos de dados, foi utilizada a ferramenta *Efficacious Reproducer Engine for Network Operations* (ERENO) (QUINCOZES et al., 2023). Por meio da Tabela 1, é possível observar as classes de ataque que compõem o conjunto de dados de treinamento, avaliação e detecção de cada IDS, bem como a quantidade de amostras para cada uma dessas classes.

Classe	**IDS 1**			**IDS 2**			**IDS 3**		
	Treino	**Avaliação**	**Detecção**	**Treino**	**Avaliação**	**Detecção**	**Treino**	**Avaliação**	**Detecção**
normal	9999	5000	-	9999	9999	14996	9999	5000	-
random replay	4999	2500	-	-	4999	5000	4999	2500	-
inverse replay	4999	2500	-	-	4999	5000	4999	2500	-
masquerade fake fault	500	500	-	4999	4999	5000	4999	2500	-
masquerade fake normal	500	500	-	4999	4999	5000	4999	2500	-
injection	4999	2500	-	-	4999	5000	4999	2500	-
high StNum	-	0	-	-	4999	2500	4999	2500	-
poisoned high rate	500	500	-	-	4998	2500	4998	2500	-
Total	26496	14000	-	19997	44991	44996	44991	22500	-

Tabela 1 – Divisão das classes de ataque nos conjuntos de dados de treinamento

Essa distribuição heterogênea foi utilizada com o intuito de se estabelecer uma diferença significativa de conhecimento entre os microsserviços que compõem a Rede de Conselhos. Conforme abordado na Seção 3.9, o IDS 2 representa o microsserviço responsável pela execução da fase de detecção e envio das solicitações de conselho à rede. Portanto, seu conjunto de dados para essa etapa foi estabelecido contendo todas as classes de ataques descritas que compõem os conjuntos de dados de treinamento dos demais microsserviços. Além disso, seu conjunto de dados de treinamento foi organizado de forma que seus classificadores tivessem conhecimento de apenas duas classes de ataque. Em contrapartida,

os IDSs conselheiros tiveram seus classificadores treinados com uma variedade maior de classes de ataque, com o objetivo de contribuir com conselhos de forma mais eficaz.

Com a construção do modelo dos classificadores do IDS 2, seguida pela avaliação dos mesmos com um conjunto de dados de avaliação, e a execução da etapa de detecção de intrusões com a composição do conjunto de dados conforme explanado na Tabela 3.10, o detector encontrou uma quantidade significativa de conflitos. Esse cenário serve como premissa para as análises e resultados deste trabalho.

4 Resultados

Neste capítulo, são apresentados os experimentos conduzidos para avaliar o desempenho da Rede de Conselhos sob a nova abordagem, juntamente aos resultados obtidos em cada um deles.

4.1 Avaliação de Classificadores

Em primeiro lugar, foi realizado um experimento com o objetivo de avaliar o desempenho de cada classificador em relação a cada classe de ataque. Esse experimento não envolveu o fluxo da rede de conselhos e na Figura 4 é possível visualizar esses resultados.

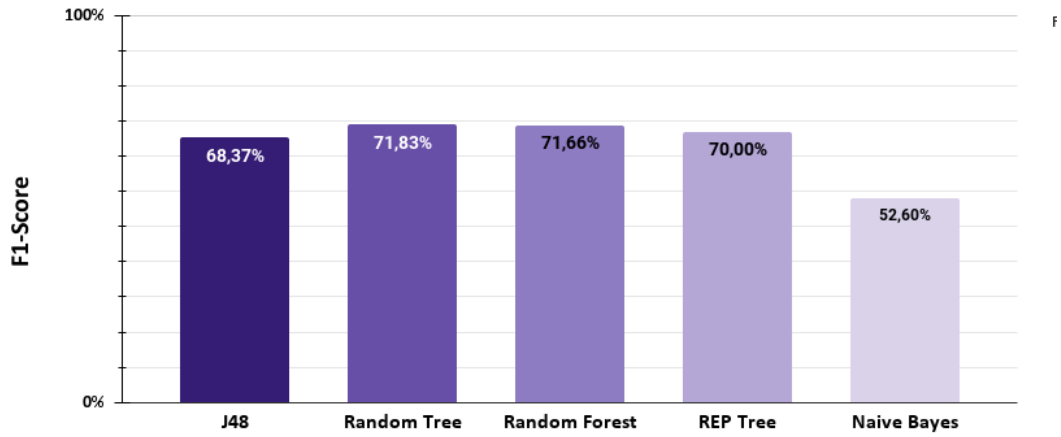

Figura 4 – *F1-Score* por classificador (sem Rede de Conselhos)

Ao analisar os dados contidos na Figura 4, percebe-se que a performance individual de cada classificador, mensurada pelo *F1-Score*, é relativamente baixa, com o seu maior valor sendo 71,83%. Com isso, deduz-se que, se a Rede de Conselhos fosse substituída por apenas um desses classificadores para classificação de todas as amostras, o desempenho do sistema como um todo seria comprometido, devido à baixa eficácia de cada modelo de lidar com a variedade de ataques presentes no conjunto de dados. Por outro lado, a Rede de Conselhos possibilita uma estratégia colaborativa, unindo o conhecimento de diferentes classificadores a fim de minimizar os conflitos e melhorar a precisão de detecção.

4.2 Aprendizado com Todos os Conselhos

Como prova de conceito da nova arquitetura para a Rede de Conselhos, foi conduzido um experimento na qual o IDS 2 envia uma solicitação de conselho para cada conflito

encontrado e, em resposta à isso, os IDS 1 e IDS 3 enviam seus respectivos conselhos. Isso significa que, para cada conflito enviado para à rede, têm-se dois conselhos correspondentes, fazendo com que o IDS 2 necessite escolher o conselho a ser utilizado para atualizar a sua base de conhecimento (Seção 3.8).

Com todos os conselhos processados e as amostras adicionadas ao conjunto de dados de treinamento, o IDS recomeça todo o fluxo de funcionamento para validar o aprendizado adquirido. Os classificadores têm seus modelos de classificação atualizados com o novo conjunto de dados e são avaliados em seguida. A partir disso, executa-se a fase de detecção, com os classificadores possuindo um conhecimento maior.

Por meio desse experimento, os dados obtidos comprovaram a contribuição da nova abordagem da Rede de Conselhos para o aprendizado do IDS, bem como para a eficiência e agilidade do fluxo de detecção. Em primeiro lugar, foi possível perceber uma redução significativa no número de conflitos encontrados na etapa de detecção após aprender com todos os conselhos. Na primeira execução dessa tarefa, que foi a premissa para a atuação da Rede de Conselhos, foram registrados um total de 29515 conflitos e, nessa segunda execução, foram encontrados 1313 conflitos, representando uma redução de aproximadamente 95%.

Além disso, os resultados gerais foram abstraídos a partir das métricas mensuradas para as amostras que os classificadores do IDS 2 conseguiram classificar sem conflitos, somadas às métricas dos conselhos obtidos. Esse somatório resulta no total de amostras pertencentes ao conjunto de dados de detecção utilizado por esse microsserviço.

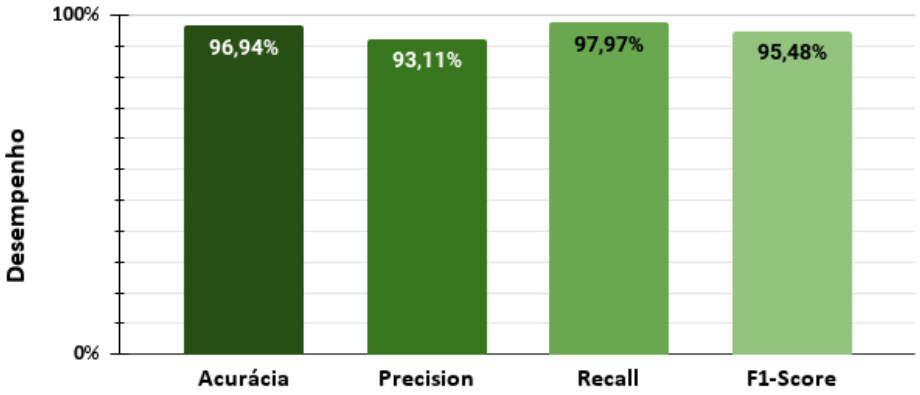

Figura 5 – Métricas dos resultados gerais obtidos

Além desses resultados, após realizar um reteste único da etapa de detecção com os modelos de classificação treinados e avaliados após o aprendizado com todos os conselhos, foram extraídos os dados conforme a Figura 6.

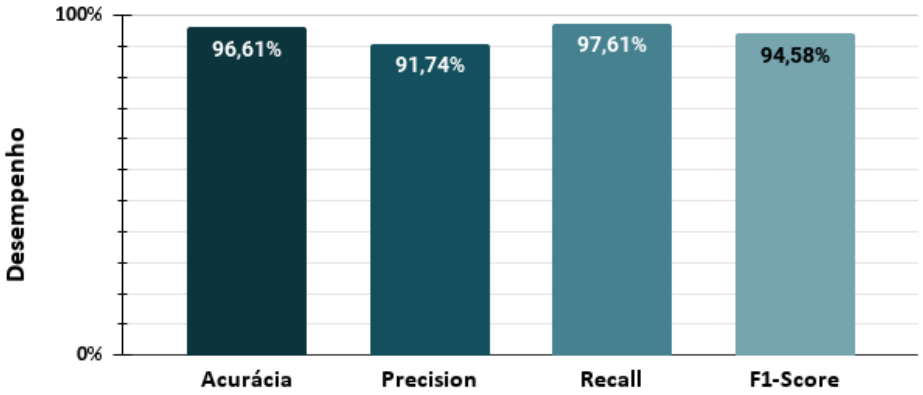

Figura 6 – Métricas dos resultados de reteste após aprendizado com conselhos

Adicionalmente, também foram extraídas as mesmas métricas, porém referentes aos conselhos enviados de forma geral, bem como aos que foram enviados somente pelo IDS 1 ou pelo IDS 3, a fim de avaliar seus respectivos desempenhos como conselheiros na Rede de Conselhos. Esses dados são representados por gráficos, conforme as Figuras 6, 8 e 9.

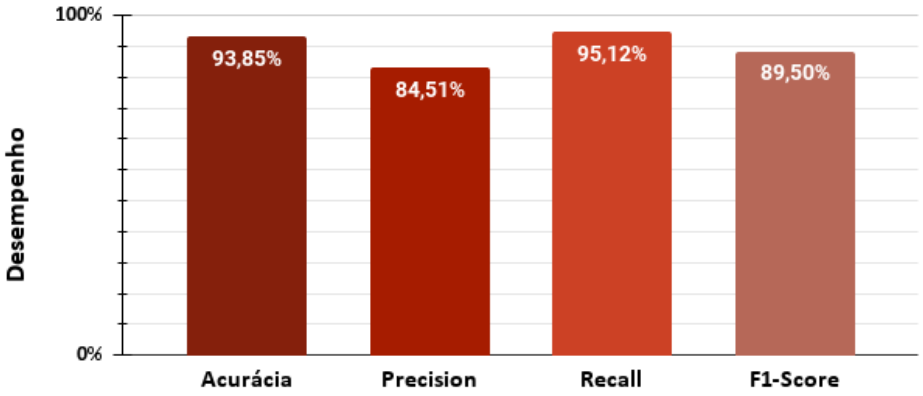

Figura 7 – Métricas de todos os conselhos obtidos

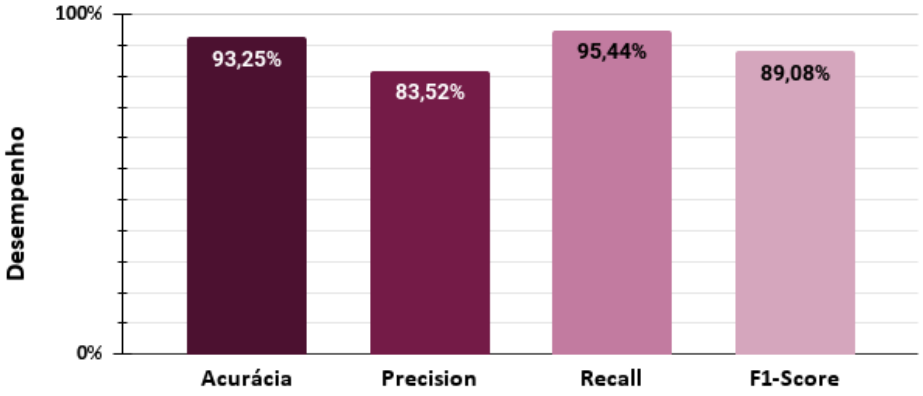

Figura 8 – Métricas dos conselhos enviados pelo IDS 1

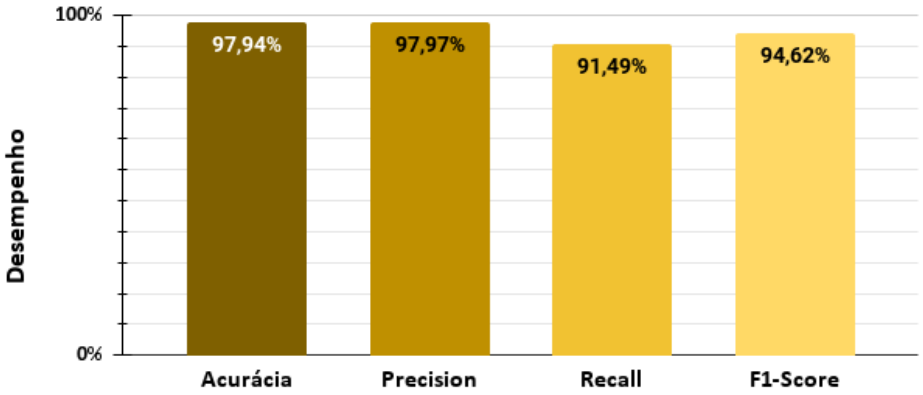

Figura 9 – Métricas dos conselhos enviados pelo IDS 3

Uma outra abordagem para avaliar os modelos de classificação é utilizando a matriz de confusão. Ela estrutura os resultados da classificação em uma tabela de dupla entrada, possibilitando a comparação entre os rótulos reais e as predições do modelo. Cada classe da matriz é usada para indicar as previsões corretas e errôneas para cada classe, permitindo a identificação de padrões de acerto e erro (POWERS, 2020).

A Figura 10 apresenta a matriz de confusão resultante deste experimento. Por meio dessa matriz, é possível examinar a maneira eficaz como o IDS 2, após a aprendizagem com todos os conselhos, retém o conhecimento obtido, gerando uma elevada taxa de acerto na segunda fase de detecção. Nota-se que, mesmo sem solicitar novos conselhos nessa execução, os classificadores mantiveram uma alta precisão na classificação das amostras. Este resultado demonstra que a Rede de Conselhos não só soluciona conflitos pontuais, mas também auxilia na autonomia do sistema a longo prazo, reduzindo a necessidade de novas intervenções.

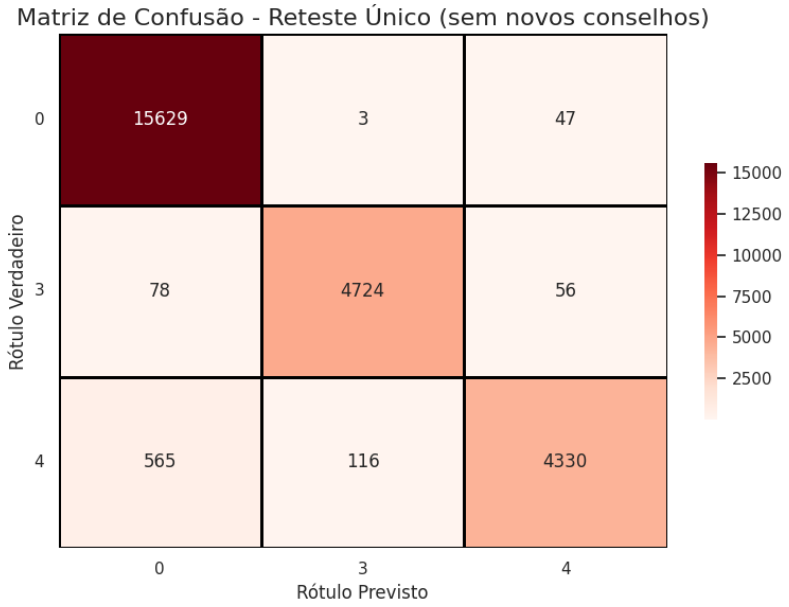

Figura 10 – Matriz de Confusão do Reteste Único

4.3 Aprendizado a Cada Conselho

Além do experimento de aprendizado com todos os conselhos, foi realizado um cenário no qual, para cada conselho recebido, o microsserviço executa novamente a etapa de treinamento, para atualizar o modelo dos classificadores, seguida pelas etapas de avaliação e detecção. Dessa forma, é possível avaliar a influência de cada conselho no aprendizado do detector, possibilitando identificar quais foram os conselhos positivos e negativos ao seu desempenho.

No entanto, foi observado que essa é uma operação com alto custo de processamento, e pode não ser a melhor abordagem. Essa limitação se deve ao fato de que, para o conjunto de dados de treinamento do IDS 2, composto por 19997 instâncias, conforme explanado da Tabela 3.10, o microsserviço leva, em média, 50 segundos para construir seu modelo de classificação, sendo aumentado à medida que novos conselhos são processados e, consequentemente, novas amostras são adicionadas à esse conjunto de dados.

Com base na distribuição dos conjuntos de dados, abordada na Seção 3.10, foram encontrados 29515 conflitos. Isso significa que, considerando uma média de 50 segundos para execução da etapa de treinamento, e tendo em vista que esse procedimento será realizado a cada conselho, o tempo total de execução seria de aproximadamente 17 dias.

Após as primeiras 24 horas de execução, o IDS 2 processou e aprendeu com um total de 1500 conselhos. Nesse ponto, foi constatado que o *F1-Score* dos classificadores para a etapa de detecção estava em 92.82%, e sua acurácia em 94.24%.

A Tabela 2 apresenta um comparativo entre os conselhos fornecidos e a precisão dos conselheiro IDS 1 e IDS 3 ao longo do experimento. Nota-se que o IDS 1 emitiu um total de 1080 conselhos até o momento, dos quais 1040 estavam corretos, o que indica um alto nível de acerto. Por outro lado, o IDS 3 emitiu 420 conselhos, com uma assertividade de 402. Esses dados possibilitam avaliar, individualmente, a influência dos conselheiros no aprendizado do sistema, e demonstra a eficácia de cada um no aprimoramento do modelo do IDS 2, podendo ser útil, em futuras implementações, para auxiliar na tomada de decisão sobre qual conselheiro é o mais confiável na rede.

Conselheiros	**Conselhos**	**Corretos**
IDS 1	1080	1040
IDS 3	420	402

Tabela 2 – Comparativo entre os conselhos e assertividade dos conselheiros

Este experimento demonstrou que, com essa estratégia, é possível identificar quais conselhos estão sendo favoráveis ou prejudiciais ao desempenho do IDS na tarefa de detecção de intrusões. Isso possibilita a realização de experimentos adicionais, bem como a remoção dos conselhos que impactam negativamente o sistema. Contudo, o alto tempo de execução e custo de processamento tornam essa estratégia inviável em termos práticos.

Uma alternativa é a implementação do paradigma de *multithreading*, que possibilita a execução simultânea de diversas tarefas em um único processo (TANENBAUM; BOS, 2015). Ao utilizar a abordagem de *multithreading*, o sistema seria capaz de processar cada conselho simultaneamente, atualizando seu modelo de classificação sem interromper o seu fluxo atual de detecção. Com isso, o tempo de duração do processo seria reduzido significativamente e o desempenho do sistema como um todo seria otimizado.

5 Conclusão

No contexto de segurança cibernética, a Rede de Conselhos desempenha um papel fundamental no aprimoramento de Sistemas de Detecção de Intrusões (IDSs), pois permite a colaboração de sistemas heterogêneos na classificação de amostras desconhecidas, de forma que o conhecimento é compartilhado e a eficácia na detecção de intrusões aumente. No entanto, a aplicação tradicional, implementada de forma monolítica, apresenta grandes limitações no que se refere ao seu desempenho e escalabilidade, visto que o fluxo da rede necessita ser dinâmico, com os sistemas operando em paralelo e se comunicando conforme necessário. Além disso, à medida que o conjunto de dados utilizados aumenta, o tempo de execução do projeto como um todo também aumenta, uma vez que as etapas de treinamento, avaliação e detecção de cada detector demandam maior duração, principalmente por serem executadas de forma sequencial para cada um dos detectores da rede.

Para enfrentar os desafios expostos acima, foi desenvolvida uma nova arquitetura para a Rede de Conselhos, orientada à microsserviços que representam cada IDS da rede. Cada microsserviço executa suas tarefas de forma autônoma, sem a necessidade de aguardar o fluxo dos outros IDSs. Uma vez desacoplados, foi implementada uma estrutura de mensageria com o Kafka que, além de permitir maior agilidade e eficácia na comunicação entre os microsserviços, permite que o detector publique as requisições de conselhos e avance na tarefa de detecção das amostras subsequentes, enquanto os conselheiros processam essas solicitações e retornam os respectivos conselhos.

Como prova de conceito, foram utilizados conjuntos de dados heterogêneos, divergentes em tamanho e tipos de classes, e distribuídos entre os IDSs. A partir disso, foram mensuradas as métricas de tempo para a execução das etapas realizadas por cada detector. Com isso, constatou-se que, na realização das etapas *off-line* dos IDSs, especificamente as tarefas de treinamento e avaliação dos classificadores, a nova abordagem obteve uma redução de 50% de duração em relação à forma monolítica, a qual o fluxo ocorria de forma sequencial. Adicionalmente, os resultados demonstraram que, no uso do Kafka para compartilhamento de mensagens, o intervalo de espera do microsserviço, desde a solicitação do conselho até o retorno dos dois conselheiros é inferior a 1 segundo. Portanto, considerando que o IDS publica as solicitações em tempo real enquanto realiza a etapa de detecção, todos os conselhos já estarão disponíveis para processamento e escolha assim que esse processo for finalizado, comprovando, mais uma vez, que a nova arquitetura da Rede de Conselhos é mais eficiente do que a abordagem tradicional.

Em trabalhos futuros, planeja-se implementar o paradigma de *multithreads* na

Rede de Conselhos, onde o fluxo ocorre em uma *thread* central. Nessa abordagem, para cada conselho obtido, uma nova *thread* é iniciada, permitindo que o detector execute as etapas de treinamento, avaliação e detecção de forma independente. Após processar cada conselho, as métricas são calculadas ainda na *thread* adicional e, se for constatado que o aprendizado agregou positivamente ao modelo de classificação, retorna-se a *thread* integrando os resultados obtidos à ela. Por fim, com o uso de *multithreads*, pretende-se implementar o aprendizado passivo entre IDSs. Essa abordagem se tornou possível pela estrutura desenvolvida neste trabalho, e consiste em uma metodologia na qual, a cada conselho aprendido, o IDS envia um *feedback* à rede, e outros IDSs possam aprender com a mesma amostra, mesmo não tendo encontrado conflitos prévios.

Referências

ANDERSON, C. Docker [software engineering]. **IEEE Software**, IEEE, v. 32, n. 3, p. 102–c3, 2015. Citado na página 25.

AYALA, G.; YANO, Y. A collaborative learning environment based on intelligent agents. **Expert Systems with Applications**, Elsevier, v. 14, n. 1-2, p. 129–137, 1998. Citado na página 14.

BRASIL, C. **Levantamento mostra que ataques cibernéticos no Brasil cresceram 94%**. 2022. Acesso em: 08 nov. 2024. Disponível em: <https://www.cnnbrasil.com.br/tecnologia/levantamento-mostra-que-ataques-ciberneticos-no-brasil-cresceram-94/>. Citado na página 9.

_____. **Brasil é o 4º país da América Latina com mais ameaças digitais no primeiro semestre de 2024, diz pesquisa**. 2024. Acesso em: 08 nov. 2024. Disponível em: <https://www.cnnbrasil.com.br/economia/macroeconomia/brasil-e-o-4o-pais-da-america-latina-com-mais-ameacas-digitais-no-primeiro-semestre-de-2024-diz-p>. Citado na página 9.

CHICCO, D.; JURMAN, G. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. **BMC genomics**, Springer, v. 21, p. 1–13, 2020. Citado na página 17.

CHOI, H.; KIM, M.; LEE, G.; KIM, W. Unsupervised learning approach for network intrusion detection system using autoencoders. **The Journal of Supercomputing**, Springer, v. 75, p. 5597–5621, 2019. Citado na página 13.

CUNNINGHAM, P.; CORD, M.; DELANY, S. J. Supervised learning. In: **Machine learning techniques for multimedia: case studies on organization and retrieval**. [S.l.]: Springer, 2008. p. 21–49. Citado na página 13.

DocuSign. **Segurança de Rede: Como Proteger a Informação Corporativa**. 2023. [Acessado em: 11 de novembro de 2024]. Disponível em: <https://www.docusign.com/pt-br/blog/seguranca-de-rede>. Citado na página 9.

DUA, M. et al. Machine learning approach to ids: A comprehensive review. In: IEEE. **2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)**. [S.l.], 2019. p. 117–121. Citado na página 9.

HSU, Y.-F.; MATSUOKA, M. A deep reinforcement learning approach for anomaly network intrusion detection system. In: IEEE. **2020 IEEE 9th International Conference on Cloud Networking (CloudNet)**. [S.l.], 2020. p. 1–6. Citado na página 15.

JAVADPOUR, A.; PINTO, P.; JA'FARI, F.; ZHANG, W. Dmaidps: A distributed multi-agent intrusion detection and prevention system for cloud iot environments. **Cluster Computing**, Springer, v. 26, n. 1, p. 367–384, 2023. Citado na página 15.

JIAO, Y.; DU, P. Performance measures in evaluating machine learning based bioinformatics predictors for classifications. **Quantitative Biology**, Springer, v. 4, p. 320–330, 2016. Citado na página 17.

KHRAISAT, A.; GONDAL, I.; VAMPLEW, P.; KAMRUZZAMAN, J. Survey of intrusion detection systems: techniques, datasets and challenges. **Cybersecurity**, Springer, v. 2, n. 1, p. 1–22, 2019. Citado na página 14.

LABS, S. C. **Mid-Year Cyber Threat Report**. [S.l.], 2024. Citado na página 9.

LOPEZ-MARTIN, M.; CARRO. Application of deep reinforcement learning to intrusion detection for supervised problems. **Expert Systems with Applications**, Elsevier, v. 141, p. 112963, 2020. Citado na página 14.

MARTINA, M. R.; FORESTI, G. L. A continuous learning approach for real-time network intrusion detection. **International Journal of Neural Systems**, World Scientific, v. 31, n. 12, p. 2150060, 2021. Citado na página 15.

OLIVEIRA, I. **Levantamento mostra que ataques cibernéticos no Brasil cresceram 94%**. 2022. "<https://www.cnnbrasil.com.br/tecnologia/levantamento-mostra-que-ataques-ciberneticos-no-brasil-cresceram-94/>". Atualizado 22/08/2022. Citado na página 9.

PEDROSO, C.; BATISTA, A.; BRISIO, S.; RODRIGUES, S.; SANTOS, A. **A Direct Collaborative Network Intrusion Detection System for IoT Networks Integration**. 2023. Disponível em: <https://www.researchgate.net/publication/381326230_A_Direct_Collaborative_Network_Intrusion_Detection_System_for_IoT_Networks_Integration>. Citado na página 15.

POWERS, D. M. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. **arXiv preprint arXiv:2010.16061**, 2020. Citado na página 30.

QUINCOZES, S. E.; ALBUQUERQUE, C.; PASSOS, D.; MOSSÉ, D. A survey on intrusion detection and prevention systems in digital substations. **Computer Networks**, Elsevier, v. 184, p. 107679, 2021. Citado na página 9.

_____. Ereno: A framework for generating realistic iec–61850 intrusion detection datasets for smart grids. **IEEE Transactions on Dependable and Secure Computing**, IEEE, 2023. Citado na página 25.

QUINCOZES, S. E.; RANIERY, C.; NUNES, R. C.; ALBUQUERQUE, C.; PASSOS, D.; MOSSÉ, D. Counselors network for intrusion detection. **International Journal of Network Management**, Wiley Online Library, v. 31, n. 3, p. e2111, 2021. Citado 2 vezes nas páginas 10 e 12.

QUINCOZES, S. E.; SANTOS, C. R. P. dos; NUNES, R. C.; ALBUQUERQUE, C. V. N. de; PASSOS, D. G.; MOSSE, D. A counselors-based intrusion detection architecture. In: **LANOMS**. [S.l.: s.n.], 2019. Citado na página 18.

RAJU, V. G.; LAKSHMI, K. P.; JAIN, V. M.; KALIDINDI, A.; PADMA, V. Study the influence of normalization/transformation process on the accuracy of supervised classification. In: IEEE. **2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)**. [S.l.], 2020. p. 729–735. Citado na página 17.

REDDY, K. S. P. **Beginning Spring Boot 2: Applications and microservices with the Spring framework**. [S.l.]: Apress, 2017. Citado na página 24.

SIBLINI, W.; FRÉRY, J.; HE-GUELTON, L.; OBLÉ, F.; WANG, Y.-Q. Master your metrics with calibration. In: SPRINGER. **International Symposium on Intelligent Data Analysis**. [S.l.], 2020. p. 457–469. Citado na página 17.

SILVA, M. C. S. da. Inteligência artificial e etapas do processo de aprendizagem. **Publicações**, 2023. Citado na página 14.

TANENBAUM, A. S.; BOS, H. **Modern operating systems**. [S.l.]: Pearson Education, Inc., 2015. Citado na página 32.