ROGER ROBSON DOS SANTOS

APRENDIZAGEM POR REFORÇO PARA DETECÇÃO DE INTRUSÃO AO LONGO DO TEMPO

Dissertação apresentada ao Programa de Pós Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

CURITIBA

2021

ROGER ROBSON DOS SANTOS

APRENDIZAGEM POR REFORÇO PARA DETECÇÃO DE INTRUSÃO AO LONGO DO TEMPO

Dissertação apresentada ao Programa de Pós Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de Concentração: *Ciência da Computação*

Orientador: Prof. Dr. Altair Olivo Santin
Coorientador: Prof. Dr. Eduardo Kugler Viegas

CURITIBA

2021

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

AC	*Actor-Critic*
Ada	*Adaboosting*
ABC	*Artificial Bee Colony*
AFS	*Artificial Fish Swarm*
CAN	*Controller Area Network*
CFS	*Correlation-based Feature Selection*
DT	*Decision Tree*
DRL	*Deep Reinforcement Learning*
DQN	*Deep Q-Network*
DDQN	*Double Deep Q-Network*
FP	Falso Positivo
FN	Falso Negativo
FCM	*Fuzzy C-Means Clustering*
GBT	*Gradient Boosting*
GBFS	*Gradient Boosting Feature Selection*
HNB	*Hidden Naïve Bayes*
IDS	*Intrusion Detection System*
NIDS	*Network-based Intrusion Detection System*
MAWI	*Packet traces from WIDE backbone*
PG	*Policy Gradient*
RF	*Random Forest*
RFA	*Recursive Feature Addition*
WFI	*Weighted Feature Importance*

Resumo

Diversos trabalhos sobre detecção de intrusão baseado em rede (NIDS - *Network-based Intrusion Detection System*) tem sido encontrado na literatura. Contudo, apesar dos resultados promissores reportados, como as altas taxas de acurácia, a grande maioria não avalia a longevidade do modelo e a precisão da classificação ao longo do tempo. Este trabalho propõe uma nova abordagem para detecção de intrusão baseada rede através de técnicas baseada em aprendizado por reforço que resiste a longos períodos sem atualizações de modelo, dividida em duas estratégias de desenvolvimento. Primeiramente, a proposta aplica modelos de aprendizado de máquina como uma tarefa de aprendizado de reforço buscando a longevidade do modelo e a precisão da classificação ao longo do tempo. Posteriormente, as atualizações do modelo são realizadas por meio de uma técnica de *transfer learning* aplicada em um mecanismo de janela deslizante que diminui significativamente a necessidade de recursos computacionais e intervenção humana. Experimentos realizados usando um conjunto de dados que abrange 8 TB de dados e quatro anos de tráfego de rede real, indicam que as abordagens atuais na literatura não são capazes de tratar as mudanças de comportamento ao longo do tempo do tráfego de rede. Adicionalmente, a técnica proposta sem atualizações do modelo atinge taxas de precisão semelhantes aos esquemas de detecção tradicionais com atualizações semestrais. No caso de realizar atualizações no modelo proposto, isso diminui os falsos positivos em até 8%, os falsos negativos em até 34% e a variação de precisão em 6% em comparação as outras abordagens. Além disso, a tarefa de atualização requer apenas 7 dias de dados de treinamento sendo quase 5 vezes menos custosa computacionalmente.

Palavras Chaves: Aprendizagem por Reforço, Aprendizagem de Máquina, Detecção de Intrusão.

Capítulo 1

Introdução

1.1 Contextualização

O número de ataques cibernéticos tem aumentado significativamente, atualmente representando quase um quinto de todo o tráfego de rede no mundo [1]. De acordo com um relatório da Cisco [2], mais de sete milhões de ataques às redes de computadores foram identificados em 2018 e esse número deve dobrar até o final de 2023. Como resultado, as empresas precisam ter acesso às soluções de segurança que possam detectar esse número crescente de ataques de rede ao longo do tempo.

Os sistemas de detecção de intrusão *Intrusion Detection System* (IDS) são amplamente implantados para monitorar e identificar ataques de rede. Para tanto, possuem o objetivo de classificar as atividades maliciosas e neutralizá-las em um determinado ambiente de rede. As soluções da literatura frequentemente fazem uso de duas abordagens principais para realizar essa tarefa de detecção de intrusão [3]. Em primeiro lugar, as abordagens baseadas no uso indevido utilizam assinaturas com padrões de ataques de rede conhecidos para identificá-los no tráfego de rede. No entanto, estes sistemas detectam apenas assinaturas previamente conhecidas [4], deixando os sistemas desprotegidos contra ataques de *zero day*, por exemplo. Por outro lado, as abordagens baseadas em comportamento analisam o comportamento de todos os eventos de rede em um determinado ambiente para sinalizar fluxos de rede inadequados com base em suas semelhanças, por exemplo, aplicando um modelo de aprendizagem de máquina. Soluções que adotam essa abordagem geralmente são capazes de detectar novos ataques, mas somente se eles se comportarem da mesma forma que os ataques previamente conhecidos

durante a construção do modelo comportamental [5], [6].

O número cada vez maior de novos ataques identificados tem motivado a proposta de vários trabalhos (por exemplo, [3], [7], [8]) com o objetivo de criar formas de detecção de intrusão baseadas em comportamentos que empregam aprendizagem de máquina através de técnicas de reconhecimento de padrões. Essas soluções devem construir seus modelos de aprendizagem de máquina com base em conjuntos de dados massivos que, por sua vez, contêm milhões ou até bilhões de eventos de rede, que representam de forma confiável o comportamento diverso esperado nos respectivos ambientes de produção [5]. Embora os modelos de aprendizagem de máquina sejam capazes de identificar eventos associados à ataques desconhecidos através do treinamento, o uso desses conjuntos de dados implica em fases de treinamento de alto custo computacional assim como vários desafios relacionados à rotulação de tais eventos [9].

1.2 Motivação

O comportamento em ambientes de rede pode variar consideravelmente ao longo do tempo, seja devido à exploração de novos ataques, ou, devido ao surgimento de novos serviços [5], [6]. Diversos comportamentos foram evidenciados em ambientes de produção de rede que tornam modelos de aprendizagem de máquina rapidamente desatualizados, exigindo que os administradores de sistemas realizem atualizações frequentes e custosas [5]. A justificativa para as atualizações do sistema, são que os modelos de aprendizagem de máquina desatualizados não conseguem manter as taxas de precisão na identificação de ataques de rede obtidas durante a fase de teste, tornando-se não confiáveis e colocando em risco os sistemas durante seu uso em ambientes de produção [6]. Neste caso, os alertas de IDS aumentam e se tornam em sua maioria falsos positivos, impulsionando os administradores a desconsiderá-los enquanto um novo modelo de aprendizagem de máquina atualizado não estiver disponível. Surpreendentemente, na literatura, os autores frequentemente não se preocupam com a longevidade dos modelos desatualizados e ignoram o custo relacionado as atualizações periódicas do modelo. Em muitos casos, negligenciam os desafios relacionados a tarefa de retreinamento do modelo [10].

As atualizações dos modelos utilizados para reconhecimento de padrões exigem a coleta de eventos atualizados, auxílio de um especialista para rotulagem dos eventos e a execução do processo de retreinamento do modelo com um alto custo computacional atrelado [3]. Sendo

assim. tal procedimento pode demandar semanas ou até meses de assistência humana especializada, muitas vezes não disponível em algumas organizações ou, ainda, extremamente custosas [11]. Portanto, manter modelos de aprendizagem de máquina confiáveis e com maior longevidade por longos períodos de tempos é uma necessidade para a devida implantação de IDS em ambientes de produção, considerando que as atualizações de modelo não são fáceis ou baratas.

Os modelos obtidos através das abordagens tradicionais de reconhecimento de padrões não são projetados para durar muito tempo em relação a suas taxas de classificação [6]. Porém, de modo geral, os autores de trabalhos anteriores buscam altas taxas de acurácia na classificação. Entretanto, não se preocupam com as graves diminuições de desempenho na detecção intrusão com o passar do tempo [5]. Atualmente existem poucas técnicas de aprendizagem de máquina para detecção de intrusão que avaliam a longevidade de seus modelos utilizados [12]. Logo, torna-se necessário que estes modelos possuam um tempo de longevidade maior, para que possam continuar a ser utilizados mesmo enquanto estão sendo geradas as coletas de eventos de rede para uma eventual atualização do modelo.

Os modelos de aprendizagem de máquina desatualizados devem ser atualizados o mais rapidamente possível. No entanto, mesmo a identificação de modelos desatualizados é um desafio [10], uma vez que o administrador deve avaliar manualmente se a precisão do modelo atual ainda atende à precisão medida na fase de teste. Em geral, as abordagens propostas dependem de configurações supervisionadas que assumem que o rótulo adequado do evento estará sempre disponível [13]. Porém, o rótulo adequado do evento é desconhecido em produção, ao contrário do que acontece na fase de teste. As abordagens atuais negligenciam o impacto da duração do modelo, tipicamente assumindo que as atualizações periódicas podem ser facilmente realizadas sem considerar os custos ou, ainda, quando o modelo de aprendizagem de máquina atual se tornará desatualizado [5].

As atualizações do modelo devem exigir satisfatoriamente custo computacionais mais baixos e a menor quantidade de dados rotulados (por exemplo, devido aos altos custos de rotulagem de eventos e o custo computacional relacionado) [5]. As soluções de reconhecimento de padrões tradicionais descartam seus modelos de detecção de intrusão em uso e criam modelos novos com base nos dados de treinamento recém-obtidos. Portanto, cada atualização exige uma quantidade significativa de dados para extrair o novo comportamento de aprendizagem de máquina de maneira adequada, além de demandar recursos computacionais

significativos. Neste caso, os dados de treinamento devem ser tão grandes quanto os originais (apesar do conhecimento prévio disponível do modelo desatualizado descartado) [3]. Consequentemente, os esquemas de detecção de intrusão baseados em aprendizagem de máquina permanecem como um tópico de pesquisa, sendo raramente usados em ambientes de produção, devido à frequente mudança de comportamento do tráfego de rede e aos desafios que ele enfrenta para tratar tais aspectos.

Na literatura podem ser encontradas técnicas capazes de atualizar os modelos periodicamente. O *Streaming Learning* é uma técnica que permite atualizar o modelo periodicamente, fazendo com que o algoritmo incremente o comportamento de rede dos novos dados informados. Contudo, a grande maioria dos trabalhos assumem que o cenário de detecção de intrusão é supervisionado [13] e o rótulo do evento sempre se encontrará disponível. Porém, como ressaltado anteriormente, no cenário de detecção de intrusão de produção é inviável um especialista rotular todos os dados que trafegam diariamente na rede, tornando este um problema para atualização periódica do modelo.

Em diversos trabalhos os autores utilizam conjuntos de dados muito antigos ou até mesmo simulados que, por sua vez, não são capazes de refletir o comportamento diverso que ocorre ao longo do tempo na rede [5]. Este fato impossibilita a criação de sistemas capazes de identificar ataques de rede com eficácia em tráfegos de rede real, uma vez que as bases frequentemente utilizadas para este fim não apresentam a variabilidade do tráfego ao longo do tempo.

1.3 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um novo modelo de detecção de intrusão baseada em rede através de aprendizagem por reforço, capaz de manter a longevidade do modelo, aumentar a precisão das classificações, diminuir a variabilidade das classificações e tornar o processo de atualização menos custoso computacionalmente. Para tanto, o modelo busca diminuir a variabilidade das acurácias obtidas ao longo do tempo, através de uma nova métrica de recompensa durante seu treinamento, fazendo uso da distância para a classe correta, diferentemente das abordagens da literatura que buscam apenas a acurácia. Além disso, a técnica facilita a atualização dos modelos ao longo do tempo, fazendo uso da técnica de *transfer learning*, logo, demanda um volume menor de eventos de redes rotulados, fazendo com que

diminua significativamente o custo humano da rotulagem destes dados, assim como o custo computacional relacionado as atualizações dos modelos.

1.4 Objetivos Específicos

O objetivo geral se desdobra nos seguintes objetivos específicos:

I. Avaliar as técnicas da literatura, em relação à longevidade do modelo, precisão, variabilidade das classificações, assim como o custo relacionado as atualizações dos modelos;
II. Desenvolver um método baseado em aprendizagem por reforço para detecção de intrusão baseado em rede;
III. Desenvolver um método baseado em aprendizagem por reforço para atualização do modelo através de técnicas de *transfer learning*;
IV. Avaliação e melhoria dos resultados.

1.5 Contribuições

Este trabalho apresenta como principais contribuições:

- Um novo modelo de aprendizagem por reforço para detecção de intrusão, que melhora significativamente a taxa de classificação, melhora a longevidade do modelo obtido e reduz a variação da acurácia de classificação ao longo do tempo;
- Uma avaliação de precisão, variação e longevidade da classificação, das técnicas de aprendizagem de máquina amplamente utilizadas para detecção de intrusão, pois experimentos indicam que os métodos atuais na literatura não lidam com mudanças no tráfego de rede ao longo do tempo (ou seja, sua precisão diminui com o passar do tempo), enquanto também exigem atualizações de modelo frequentes e custosas computacionalmente, o que os tornam inviáveis para implantações em ambientes de produção;

1.6 Contribuições científicas

A tabela a seguir apresenta as publicações submetidas ou publicadas durante o mestrado, relacionadas ao tema do documento:

Tabela 1. Publicações realizadas e/ou submetidas como autor principal.

Nome do Trabalho	**Autores**	**Lugar da Publicação**	**Qualis**
A Long-lasting Reinforcement Learning Intrusion Detection Model	Roger R. dos Santos, Eduardo K. Viegas, Altair O. Santin e Vinicius Cogo	AINA 2020	A2
Um Sistema de Detecção de Intrusão Baseado em Aprendizagem por Reforço	Roger R. dos Santos, Eduardo K. Viegas, Altair O. Santin e Jackson Mallmann	SBSeg 2020	A4
Reinforcement Learning for Intrusion Detection: More Model Longness and Fewer Updates	Roger R. dos Santos, Eduardo K. Viegas, Altair O. Santin e Vinicius Cogo	TNSM 2021 (submetido)	Q1
Improving Intrusion Detection Confidence Through a Moving Target Defense Strategy	Roger R. dos Santos, Eduardo K. Viegas e Altair O. Santin	GLOBECOM 2021 (submetido)	A1
A Reminiscent Intrusion Detection Model Based on Deep Autoencoders and Transfer Learning	Roger R. dos Santos, Eduardo K. Viegas e Altair O. Santin	GLOBECOM 2021 (submetido)	A1

Adicionalmente, durante o mestrado foram desenvolvidos outros trabalhos como coautor, apresentado na tabela a seguir:

Tabela 2. Publicações realizadas durante o mestrado como coautor.

Nome do Trabalho	**Autores**	**Lugar da Publicação**	**Qualis**
Sistema de Detecção de Intrusão Confiável Baseado em Aprendizagem por Fluxo	Eduardo K. Viegas, Altair O. Santin, Roger R. dos Santos e Vilmar Abreu	SBSeg 2020	A4
A Host-based Intrusion Detection Model Based on OS Diversity for SCADA	Bruno B. Bulle, Altair O. Santin, Eduardo K. Viegas e Roger R. dos Santos	IECON 2020	A2

A Multi-View Intrusion Detection Model for Reliable and Autonomous Model Updates	Rivaldo L. Tomio, Eduardo K. Viegas, Altair O. Santin e Roger R. dos Santos	ICC 2021	A1
A Machine Learning Model for Detection of Docker-based APP Overbooking on Kubernetes	Felipe Ramos, Eduardo K. Viegas, Altair O. Santin, Pedro Horchulhack, Roger R. dos Santos e Allan Espindola	ICC 2021	A1
Mitigando os Efeitos de Gan em Classificação de Imagens	Jackson Mallmann, Altair O. Santin, Alceu Britto e Roger R. dos Santos	SBSeg 2019	A4
PPCensor: Architecture for Real-time Pornography Detection in Video Streaming	Jackson Mallmann, Eduardo K. Viegas, Altair O. Santin, Roger R. dos Santos e Jhonatan Geremias	Future Generation Computer Systems	A2

1.7 Estrutura do Documento

O Capítulo 2 apresenta a fundamentação teórica na qual a linha deste trabalho se baseia. O capítulo 3 descreve os trabalhos relacionados ao tema. Já o capítulo 4 traz um estudo do estado da arte e o capítulo 5 traz a proposta aplicada no trabalho. O capítulo 6 traz a avaliação dos resultados. Por fim, no capítulo 7 será apresentado a conclusão do trabalho.

Capítulo 2

Fundamentação Teórica

Um sistema de detecção de intrusão baseado em rede *Network-based Intrusion Detection System* (NIDS) é um sistema de defesa, que possibilita evitar quebras de políticas de segurança referentes à ataques de rede. O NIDS verifica se o tráfego de rede possui padrões maliciosos que quebram as políticas de segurança da organização e gera alarmes que relatam o problema. Neste capítulo serão abordados sistemas de detecção de intrusão baseados em rede, técnicas de aprendizagem de máquina para detecção de intrusão baseada em rede e a técnica de aprendizagem por reforço.

2.1 Aprendizagem de Máquina para Detecção de Intrusão

Um sistema de intrusão baseado em rede, normalmente, objetiva identificar em tempo real quebras de política de segurança referentes à ataques de rede. Esse processo ocorre com o sistema verificando o tráfego dos pacotes de rede que são gerados durante uma conexão de rede, tentando encontrar quebras de segurança.

Os administradores de sistemas geralmente recorrem aos sistemas de detecção de intrusão baseados em rede que empregam técnicas baseadas em assinatura ou em comportamento [5]. A técnicas com assinatura são limitadas a reconhecer somente atividades maliciosas, previamente conhecidas durante a fase de treinamento. Já as abordagens baseadas em comportamento geralmente podem identificar novos ataques desconhecidos que possuam

o mesmo comportamento dos ataques de rede conhecidos durante a fase de treinamento [5] [6].

Nos últimos anos, vários sistemas de detecção de intrusão baseados em rede altamente precisos foram propostos na literatura [22]-[24], [27]-[33]. Em geral, as abordagens propostas são compostas por quatro módulos sequenciais: módulos de aquisição de dados, extração de características, classificação e alerta. Primeiro, o módulo de aquisição de dados reúne dados de rede de um ambiente monitorado, por exemplo, pacotes de rede. Enquanto o módulo extração de características extrai características comportamentais dos pacotes de rede e cria um vetor para ser utilizado por algoritmos de aprendizagem de máquina. Em NIDS, as características comportamentais são frequentemente representadas por meio de um fluxo de rede, que compõe o comportamento da rede de um host/serviço em uma determinada janela de tempo. Por exemplo, a proporção de pacotes/*bytes* trocados em um intervalo de 15 segundos de tráfego de rede. O terceiro módulo de classificação, executa os algoritmos de classificação com base no vetor de características extraídas do fluxo de rede e o classifica como normal ou ataque, por exemplo. Finalmente, se um evento malicioso for encontrado, o módulo de alerta o relata ao administrador do sistema ou toma decisões pré-programadas.

Na classificação em NIDS, geralmente são utilizadas técnicas de reconhecimento de padrões. No reconhecimento de padrões, um modelo de aprendizagem de máquina é construído por meio de um processo denominado de treinamento. A fase de treinamento produz um modelo por meio de um conjunto de dados de treinamento, que contém atividades normais e maliciosas em um determinado período. Como resultado, o modelo de aprendizagem de máquina detecta eventos de acordo com o comportamento extraído do conjunto de dados de treinamento. Portanto, quando o comportamento do ambiente muda, seja pela descoberta de novos ataques ou por novos serviços, um novo modelo de aprendizagem de máquina deve ser construído. No entanto, o processo de atualização do modelo de aprendizagem de máquina é muito custoso, pois geralmente exige assistência humana de um especialista para rotular os eventos como ataque ou normal.

A criação de conjuntos de dados (para treinamento de modelos de aprendizagem de máquina para detecção de intrusão) normalmente são feitas através da extração de características de um fluxo de dados de rede real, que possua métricas do comportamento de um ambiente de rede. Isto se mostra necessário devido ao tráfego de rede sofrer constantes mudanças à medida que novos conjuntos de ataques ou serviços surgem diariamente. Um

conjunto de dados muito antigo ou até mesmo simulado, pode impossibilitar a criação de um modelo de aprendizagem de máquina capaz de identificar ataques de rede em um ambiente real.

Tabela 3. Conjunto de características extraídas no nível de rede para cada agrupamento de características.

Grupo de Características	**Atributos (característica) Coletados**
Origem do endereço de IP Origem do endereço de IP e destino do endereço de IP Origem e destino dos serviços e portas	Tamanho médio dos Pacotes Quantidade de Pacotes (PSH Flag) Quantidade de Pacotes (SYN e FIN Flags) Quantidade de Pacotes (FIN Flag) Quantidade de Pacotes (SYN Flag) Quantidade de Pacotes (ACK Flag) Quantidade de Pacotes (RST Flag) Quantidade de Pacotes (ICMP Redirecionamento Flag) Quantidade de Pacotes (ICMP Tempo Excedido Flag) Quantidade de Pacotes (ICMP Incessível Flag) Quantidade de Pacotes (ICMP Outros tipos de Flag)

A Tabela 3 mostra um exemplo das características dos eventos de rede utilizado para detecção de intrusão [9]. Neste exemplo, um conjunto de 11 características são extraídas em uma janela de tempo de 15 segundos que contém eventos de quatro grupos: hosts, host vs. host, serviço vs. serviço e ambos os grupos anteriores. Dessa forma, é gerado um vetor com as características extraídas dos 4 grupos, conforme a Tabela 3, que possui duas características adicionais, o número de pacotes e o número de bytes, resultando em 46 características para cada fluxo de rede. O conjunto de características extraídas compõe o vetor que será utilizado como entrada pelo módulo de classificação, que irá classificar como normal ou ataque, por exemplo. Diversas abordagens podem ser utilizadas para tarefa de classificação, na qual a maioria utiliza aprendizagem de máquina com técnica de reconhecimento de padrões [3]. Por fim, se um determinando evento de rede for classificado como ataque, o módulo de alerta o reporta ao administrador do sistema.

Técnicas de aprendizado de máquina têm sido aplicadas com sucesso em diversas abordagens, incluindo a detecção de fraude [14], diagnóstico médico [15] e reconhecimento óptico de caracteres [16]. Contudo, apesar dos resultados promissores, a detecção de intrusão

utilizando aprendizagem de máquina é raramente implantada em ambientes de produção [5]. O comportamento dos ambientes de rede pode variar consideravelmente ao longo do tempo, devido à exploração de novos ataques e o surgimento de novos serviços diariamente [5], [6] e [11].

Em resumo, construir modelos de detecção de intrusão com aprendizagem de máquina de maneira confiável é uma tarefa significativamente desafiadora. Os conjuntos de dados utilizados no treinamento devem abranger comportamentos de eventos esperados de um ambiente de produção, o que nem sempre é viável devido à natureza altamente variável dos ambientes de rede. O modelo de classificação baseado em aprendizagem de máquina deve generalizar adequadamente o comportamento vivenciado na fase de treinamento [3].

Na literatura, os autores muitas vezes negligenciam ou omitem o desempenho de suas soluções em ambientes de produção, destacando apenas as altas taxas de acurácia obtidas na fase de teste. Além disso, mesmo que um modelo de aprendizagem de máquina seja construído com generalização, ele se tornará perderá ineficaz com o passar do tempo [5]. A consequência é que o modelo de aprendizagem de máquina deve ser generalizável e atualizado assim que começar a diminuir sua taxa de acurácia esperada. Apesar da grande maioria das abordagens não serem utilizadas em um ambiente de produção, a maioria ainda negligência a natureza do tráfego de rede altamente variável. Ao mesmo tempo, eles também assumem que atualizações periódicas do modelo podem ser facilmente realizadas, enquanto os desafios envolvidos em tal processo permanecem negligenciados [5], [6].

2.2 Aprendizagem por Reforço

Diferente das técnicas tradicionais de aprendizagem de máquina, a aprendizagem por reforço é uma abordagem que visa encontrar uma ótima política (por exemplo, atingir 90% de ações corretas no ambiente de treinamento) de aprendizagem durante a fase da construção do modelo [17]. As técnicas baseadas em aprendizagem por reforço são realizadas por meio de uma entidade chamada de “agente”. Essa entidade executa ações em um determinado ambiente simulado ou real, por meio de mecanismos de observação e ação. Para cada ação o agente recebe uma recompensa, podendo ser positiva ou negativa, dependendo da sua ação. O objetivo dele é maximizar as recompensas ao longo do tempo, através de tentativa e erro. Sempre que uma recompensa é positiva o agente tenta realizar ações parecidas para

maximizar cada vez mais os resultados das suas ações. Porém, quando a ação é negativa, ele passa a evitar este tipo ação. É importante ressaltar que existem casos em que o agente não conseguirá obter nenhuma recompensa positiva, por exemplo, se ele estiver executando ações em um labirinto, neste caso todas suas ações serão evitadas até que ele encontre uma saída.

Para determinar suas ações, normalmente um agente utiliza uma rede neural conforme apresentado na Figura 1. Contudo, as políticas da aprendizagem por reforço podem ser determinadas por qualquer algoritmo de aprendizagem de máquina, além disso, não é necessário que seja um algoritmo determinístico. As políticas podem ser criadas de diversas maneiras, um exemplo é criar 100 políticas e testá-las em um ambiente de treinamento. Ao final dos testes, 90 políticas foram piores e serão descartas, restando apenas 10 que foram positivas. Estas serão utilizadas para criar 90 novas políticas com características similares as 10 positivas. Cada nova política possui as características das 10 positivas, entretanto, elas possuem variações aleatórias que serão utilizadas para que o agente aprenda comportamentos diferentes dos conhecidos atualmente, dentro do ambiente treinamento [18].

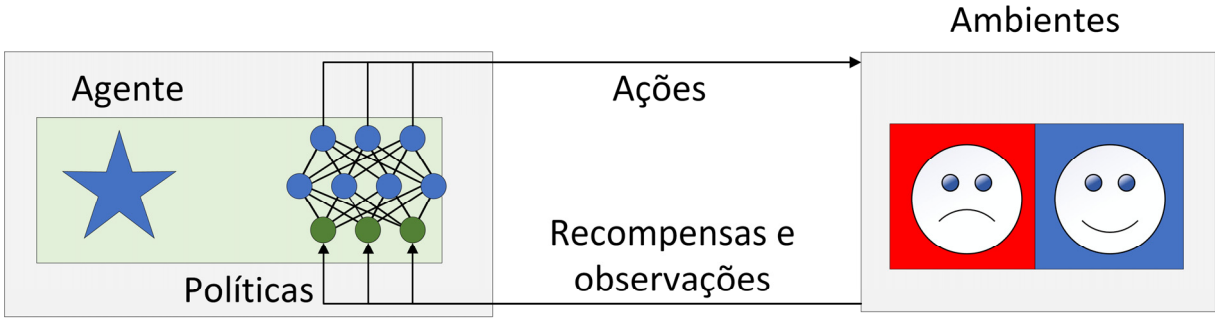


Figura 1. Aprendizagem por reforço, utilizando uma política de rede neural, adaptado de [18].

Diversas abordagens de aprendizagem por reforço utilizam as políticas por gradientes. Esta é capaz de avaliar a relação dos parâmetros utilizados em cada recompensa. Para tal técnica, é necessário utilizar a API OpenAI Gym [49], que será responsável por retornar 4 valores do ambiente de treinamento, detalhados a seguir [17]:

- **Observação**: retorna as observações de cada ação do agente dentro do ambiente, com o objetivo de criar iterações aleatórias com características das ações positivas, para utilizar em ações futuras do agente;
- **Recompensa**: retorna o valor da recompensa dada ao agente após executar uma ação no ambiente;

- **Finalizado**: retorna ao sistema um valor, informando que foram finalizadas todas as iterações do ambiente;
- **Informações**: retorna informações úteis sobre as ações ou do treinamento do agente no ambiente.

Para aplicar as políticas de rede neural, normalmente é utilizado a biblioteca TensorFlow [50]. Assim, o algoritmo receberá as observações geradas durante o treinamento através da API OpenAI Gym [49]. Em seguida, o algoritmo irá estimar a probabilidade para as próximas ações com características aleatórias baseadas nestas probabilidades. Dessa forma, é possível trazer um exemplo: Se houver apenas duas ações possíveis para o agente, a rede neural irá gerar apenas uma saída, que será a probabilidade das próximas ações. Logo, se a ação 1 for "Andar para Frente", a ação 0 for "Andar para Trás" e a rede neural retornou à probabilidade de 0,60. Neste caso, será escolhido a ação "Andar para Frente" com 60% de probabilidade, enquanto a ação "Andar para Trás" tem apenas 40% de probabilidade. A Figura 2, mostra um exemplo de como são geradas as probabilidades da rede neural [18].

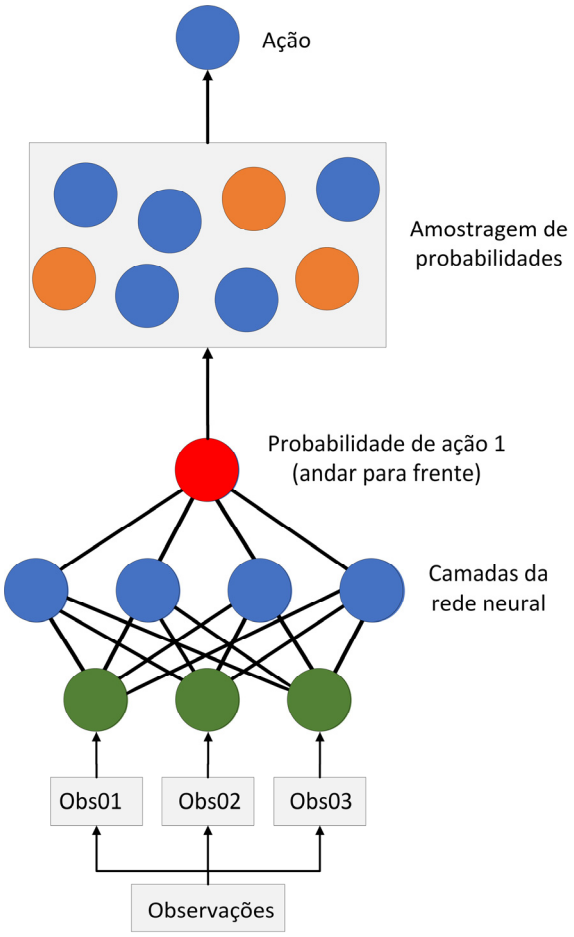


Figura 2. Políticas da rede neural, adaptado de [18].

Na Figura 2, foi gerado uma probabilidade de ação para o agente “Andar para Frente”, contudo, as políticas da rede neural permitem o equilíbrio entre as ações do agente, podendo gerar a probabilidade de o algoritmo também “Andar para Trás”. Assim, é possível que o agente encontre um equilíbrio certo entre maximizar o aprendizado e melhorar as suas ações.

Como exemplo, pode-se pensar em uma analogia onde você vai comprar um sorvete, cada vez é possível escolher aleatoriamente um sabor. A qualquer momento pode acontecer de um sabor agradar muito mais do que os sabores anteriores, fazendo com que o novo sabor seja escolhido. Em contrapartida, é possível encontrar sabores que não agradam, que por sua vez, não serão escolhidos. É importante ressaltar que, cada vez que as observações são retornadas, elas ficam armazenadas e são avaliadas com novas observações, geradas nas próximas iterações do agente. Assim, o agente estará sempre aprendendo o melhor e o pior caminho durante o treinamento [18].

A aprendizagem de máquina tradicional depende de um par de entrada e saída, onde um modelo de aprendizagem de máquina recebe de um determinado vetor de característica de eventos de rede como entrada e produz seu valor de classe estimada. As abordagens baseadas em aprendizagem por reforço são significativamente diferentes das tradicionais baseadas em aprendizagem de máquina [17]. As técnicas baseadas em aprendizagem por reforço não recebem o valor da classe de evento [3]. Em vez disso, depois que uma ação é executada, o agente recebe apenas a recompensa por sua ação e o estado do ambiente. O agente aprende o limite da decisão ideal em longo prazo, visto que otimiza suas recompensas ao longo do tempo.

Pode-se considerar em um ambiente de ataques de rede que o agente executa 100 ações, por exemplo. Porém, não sabe qual ação foi positiva ou negativa. A única coisa que ele sabe é se o ambiente foi invadido ou não por um atacante. Sendo assim, é utilizado uma estratégia de avaliação, com as somas das recompensas maximizadas por um valor de desconto, também chamado de *feedback* da ação [18]. O exemplo da Figura 3 mostra como o agente tenta acertar 3 vezes seguidas e recebe uma recompensa de +20, após a primeira ação, +10 após a segunda ação e -40 na terceira ação. Neste caso, é aplicado o valor de desconto de d = 0,95. Assim o retorno das 3 primeiras ações é representado na Equação 1. É importante verificar um valor correto de desconto para o ambiente de treinamento, assim, caso o valor de desconto seja próximo a 0, não haverá muito impacto nas recompensas

futuras. Por outro lado, se o valor de desconto for próximo a 1, as recompensas futuras terão impacto muito grande no futuro. Normalmente é utilizado o valor de desconto de 0,95 [18].

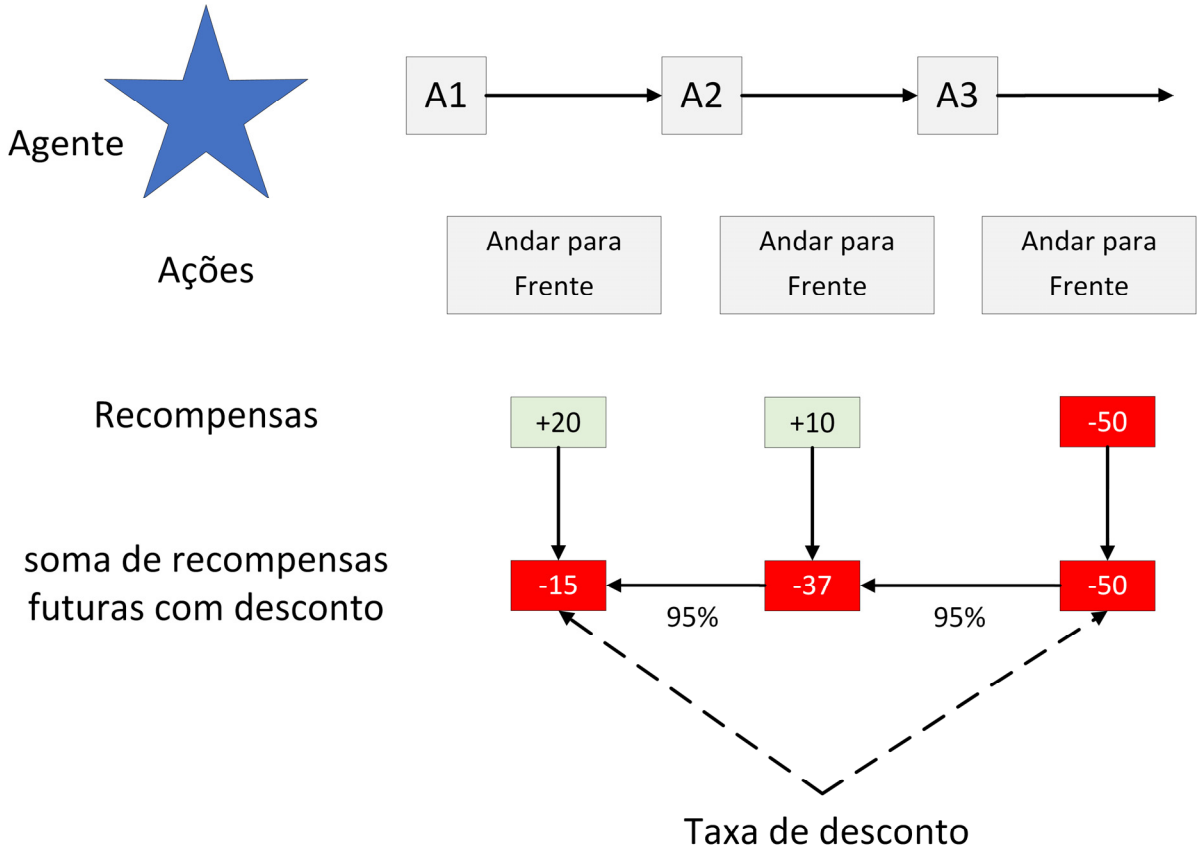


Figura 3. Calculando o retorno de uma ação: a soma das recompensas futuras com desconto, adaptado de [18]

$$20 + d\ x\ 10 + d^2 x\ (-40) = -15$$

Equação 1: Soma das recompensas futuras com desconto, da Figura 3, adaptada de [18].

O processo das políticas de gradiente gera probabilidades com as políticas da rede neural, produzindo ações em um ambiente de treinamento durante várias etapas. Em cada etapa calculam os gradientes das ações escolhidas e armazenam os valores. Após a execução de diversas iterações, é feito o cálculo de desconto de cada uma das ações, conforme mostrado anteriormente na Equação 1. Caso a ação seja positiva, significa que foi boa e devem ser aplicados os gradientes anteriores, para tornar a ação mais provável a ser escolhida no futuro. Contudo, se ação for negativa, significa que ela foi ruim e deverá ser aplicado os gradientes

opostos, para tornar uma ação menos provável no futuro. Em seguida, é feito o cálculo da média dos vetores dos gradientes para gerar novas ações futuras [18].

As políticas utilizam o conhecimento adquirido para gerar novas ações, esse processo ocorre por meio do processo de decisão de *Markov*. Neste caso, o algoritmo evolui aleatoriamente de um estado para outro. A cada etapa, um agente pode escolher uma das várias ações possíveis, as probabilidades de transição dependem da ação escolhida. Além disso, algumas transições de estado retornam alguma recompensa (positiva ou negativa). O objetivo do agente é encontrar uma política que maximize as recompensas ao longo do tempo.

Com o processo de decisão de *Markov* em execução, é utilizado o algoritmo de *Q-Learning*, que observa o agente de treinamento enquanto ele executa ações no ambiente e melhora gradativamente suas recompensas. Dessa forma, o algoritmo consegue identificar qual a política ideal para ser utilizada pelo agente no ambiente em cada iteração. Ao final, o algoritmo aplica as políticas ideais ao sistema e melhora gradativamente as ações do agente no ambiente de treinamento [18].

2.3 Conjunto de dados *MAWIFlow* [9]

A confiança dos mecanismos de detecção de intrusão baseados em rede é fortemente relacionada ao uso de um conjunto de dados de treinamento de tráfego de rede realista de ambientes de produção. No entanto, as abordagens atuais na literatura frequentemente constroem suas técnicas em cima de conjuntos de dados desatualizados, com várias falhas conhecidas [19]. Essas abordagens raramente são utilizadas em ambientes de produção, apesar dos resultados promissores relatados [5], [6]. Um conjunto de dados de treinamento [11] deve conter tráfego de rede real, que represente o comportamento dos ambientes de produção e protocolos de comunicação de rede validos. O tráfego de rede deve ser altamente diversificado em termos de protocolos usados e comportamento de rede. Os eventos de rede devem ser previamente rotulados como pertencentes a uma classe (por exemplo, normal ou ataque), para permitir a avaliação adequada do modelo e construção do classificador de aprendizagem de máquina. Finalmente, o conjunto de dados deve estar disponível publicamente para permitir o *benchmarking* adequado da proposta de detecção de intrusão.

Normalmente, as propostas na literatura coletam o tráfego de rede de um ambiente de teste controlado ou monitoram o comportamento da produção para suportar os requisitos

mencionados [9]. O primeiro permite o compartilhamento adequado de dados e rotulagem de eventos, devido à sua natureza controlada. No entanto, eles carecem de um conteúdo de rede altamente diversificado do ambiente de produção e geram tráfego de rede irreal. Contudo, os alguns casos produzem dados de rede reais e válidos que são altamente diversos, mas dificultam a rotulagem adequada de eventos e compartilhamento dos dados (por exemplo, devido às questões de privacidade associadas ao tráfego de rede real). Independentemente da abordagem adotada para a construção dos conjuntos de dados, os trabalhos da literatura não consideraram adequadamente a diversidade do comportamento do tráfego de rede. Isso significa que mesmo os melhores conjuntos de dados falharão, considerando que o tráfego de rede permanece inalterado para sempre.

O trabalho tem como proposta utilizar o conjunto de dados *MAWIFlow* [9]. Este conjunto compreende o tráfego de rede rotulado, válido e real, coletado de ambientes de produção por longos períodos. Mais especificamente, os fluxos de rede do conjunto de dados são extraídos dos traços de pacotes de rede MAWI [20], coletado diariamente por um intervalo de 15 minutos a partir de um link de trânsito entre o Japão e os EUA. Durante o período de gravação, o conjunto de dados foi constituído de um link de tráfego de rede de 1 Gbps. Além disso, as cargas úteis dos pacotes de rede são removidas e os campos de cabeçalho de pacote de rede confidenciais são anônimos. A rotulagem dos registros foi realizada utilizando uma técnica da MAWILab [21], que rotula os eventos anômalos diários (fluxos de rede) do MAWI [20], através de uma combinação de vários detectores de anomalias não supervisionados. Para este trabalho, foi considerado todo o tráfego de rede disponível para um intervalo de quatro anos, variando de 2016 a 2019. As anomalias de rede são classificadas de acordo com seus tipos de ataque rotulados pelo MAWILab [21]. Nesse caso, as anomalias da rede podem ser de vários tipos: varredura de portas, varredura de rede, negação de serviço, negação de serviço distribuída, entre outros ataques em nível de rede.

Devido à sua grande escala, mais de 8 TB de dados compostos por mais de 7 bilhões de fluxos de rede, o conjunto de dados *MAWIFlow* [9] foi construído usando o algoritmo de extração de características *BigFlow* [9], que é implementado em uma estrutura de processamento de *Big Data*. O algoritmo de extração de características extrai, para cada fluxo de rede 46 características, em uma janela de intervalo de 15 segundos, conforme listado na Tabela 3. Cada fluxo de rede é previamente rotulado como normal ou ataque, conforme

estabelecido pelos algoritmos não supervisionados do *MAWILab* [21]. A Tabela 4 mostra as estatísticas *MAWIFlow* [9] ao longo do período avaliado de quatro anos.

Tabela 4. Estatísticas do conjunto de dados *MAWIFlow* [9].

Propriedades	**Valores**
Média de pacotes de redes diários	120 Milhões
Média de fluxos de rede diários	23 Milhões
Média da taxa de internet diária	720 Mbps
Média dos fluxos anômalos diários	2.1 Milhões
Tamanho média do conjunto de dados diários	23.2 GB
Total de pacotes de rede	32.36 Bilhões
Total de fluxos de rede	7.23 Bilhões
Tamanho total do conjunto de dados	8.3 TB

2.4 Discussão

As técnicas de aprendizagem de máquina para detecção de intrusão já vêm sendo utilizadas em diversos trabalhos, a grande maioria dos trabalhos se preocupam em atingir uma alta taxa acurácia e não se preocupam com a variabilidade das mudanças de comportamento que ocorre diariamente. Isto é um problema que deve ser avaliado, pois o tráfego de rede diariamente sofre mudanças, fazendo com que um modelo de aprendizagem de máquina se torne ineficaz rapidamente e comece a gerar uma alta taxa de falsos positivos. Este problema faz com que as organizações deixem de utilizar estes sistemas em ambientes de produção, devido à falta de confiança nas classificações do sistema, o que permite não identificar ataques de rede com eficácia.

Diversos trabalhos encontrados com conjuntos de dados desatualizados ou muito antigos, não refletem o comportamento diverso de um tráfego de rede de ambientes de produção. É necessário utilizar um conjunto de dados que permita avaliar a mudança no comportamento do tráfego de rede de maneira eficaz, para que estes sistemas possam ser utilizados em ambientes de produção. Além disso, um conjunto de dados muito antigo possui comportamentos de ataques que já não existem atualmente, o que dificulta ainda mais a detecção de ataques de rede nestes sistemas.

As técnicas de aprendizagem por reforço para detecção de intrusão ainda se encontram na sua infância [7], com diversos autores utilizando abordagens para alcançar a maior precisão na classificação em um único conjunto de dados, ao invés de analisar a técnica ao longo do tempo. Entretanto, diversos trabalhos já utilizam técnicas de aprendizagem por reforço para atualização dos modelos. Contudo, a grande maioria utiliza a técnica para ambientes supervisionados, onde o rótulo sempre estará disponível, porém, este é outro problema que acontece em cenários de detecção de intrusão, devido a enorme massa de eventos de rede gerada diariamente, se torna muito custoso ter os dados rótulos a todo momento por um especialista.

Capítulo 3

Trabalhos Relacionados

A aprendizagem de máquina alcançou uma forte relação com o domínio de detecção de intrusão, uma vez que ambas as comunidades se beneficiam da otimização de algoritmos de aprendizagem de máquina em muitas dimensões (por exemplo, maximizando a precisão e minimizando alarmes falsos), enquanto abordam problemas do mundo real.

A grande maioria dos trabalhos busca maior precisão na detecção de ataques, como por exemplo, Koc et al. [22] utilizam *Hidden Naïve Bayes* (HNB) para identificar problemas detecção de intrusão que sofrem com a dimensionalidade de características altamente correlacionadas em fluxos de dados de rede em quatro grupos de ataques, comparando suas técnicas com diversas abordagens da literatura e atingindo altas taxas de acurácia. Além deste, o trabalho de Longari et al. [23] emprega uma abordagem com a técnica de LSTM e *Autoencoders,* para detecção de intrusão de redes *Controller Area Network* (CAN), apresentando resultados promissores. Contudo, os autores retratam que a técnica é relativamente lenta se comparado com outras. Otoum et al. [24] apresentam uma técnica de aprendizagem por reforço com o algoritmo de *Q-Learning* para detecção de intrusão em conjuntos de dados de redes de sensores sem fio, atingindo taxas de acurácia altíssimas. Todavia, nenhum destes trabalhos busca o objetivo da detecção de intrusão ao longo do tempo para avaliar a variabilidade dos tráfegos de rede.

Outros trabalhos da literatura buscam acelerar as atualizações do modelo realizando seleção de características, por exemplo, Akashdeep et al. [25] utilizam técnicas capazes de correlacionar e fazer o ganho de informação em cima das características do conjunto de dados, em seguida, os autores aplicaram o algoritmo de *Artificial Neural Network* (ANN) para a

abordagem proposta. Outro trabalho encontrado foi o de Hamed et al. [26] onde os autores aplicaram técnicas de *Recursive Feature Addition* (RFA) e a técnica de Brigama para escolher as características com maior relevância para identificação de ataques de rede, além disso, os autores fazem uma avaliação da precisão, da taxa de detecção e da taxa de falsos alarmes, para fazer a comparação entre diferentes sistemas da literatura. A técnica apresenta resultados promissores, entretanto, o trabalho não analisa as mudanças do comportamento da rede ao longo do tempo.

Yang et al. [27] constroem uma abordagem utilizando técnicas de *Deep Neural Network* para detectar comportamentos anormais no conjunto de dados NSL-KDD. Além disso, eles avaliam a abordagem comparando sua técnica com outras da literatura. Xiao et al. [28] destacam a falta de extração completa das características do comportamento de rede em diversos trabalhos da literatura. Este problema aumenta a alta taxa de falsos alarmes na detecção de ataques de rede. Desta forma, os autores propõem uma extração efetiva, automática das características de conjuntos de dados por meio de CNN, para identificar detecções de intrusão de maneira eficaz.

Farnaaz e Jabbar [29] aplicaram o classificador de *Random Forest* para classificação não linear, na tentativa melhorar a acurácia obtida no conjunto de dados NSL-KDD para detecção de intrusão de quatro grupos de ataques, DOS, PROBE, U2R e R2L. Da mesma forma, Sukumar et al. [30] lidam com as regras do classificador de *Random Forest* para realizar a detecção de intrusão, no conjunto de dados KDDcup99 versão original. Os autores conseguiram mostrar no trabalho uma melhora na precisão do sistema, sem levar em consideração a detecção de novos comportamentos de rede contidos no conjunto de dados. Enquanto Van et al. [31] utilizam técnicas de aprendizagem profunda (*deep learning*), através de *autoencoder* e RBM empilhados, gerando resultados em 4 grupos de ataques no conjunto de dados KDDcup99. Embora os autores tenham conseguido melhorar a detecção de intrusão em seus sistemas, os conjuntos de dados são muito antigos e apresentam diversas falhas (por exemplo, a falta de variabilidade do tráfego de rede de um ambiente de produção), interferindo na avaliação da confiabilidade do sistema ao longo do tempo. Um conjunto de dados com todo esse tempo de vida não reflete no comportamento da variabilidade da rede atualmente.

Gupta et al. [32] propõem uma abordagem de regressão aliada a uma abordagem de agrupamento com *K-Means* utilizando os dois conjuntos de dados KDDcpu99. Os autores mostraram uma comparação na detecção de intrusão entre esses dois conjuntos de dados obtendo uma acurácia média nos resultados e mostrando diferenças entre a classificação em

dois conjuntos de dados. Também foi encontrado uma abordagem baseada em algoritmos de agrupamento, proposto por Elisa et al. [33]. Sua abordagem realiza a detecção de intrusão em anomalias de rede não supervisionadas, ou seja, sem o conhecimento prévio do rótulo dos eventos. No entanto, o conjunto de dados utilizado no trabalho não é realista, enquanto a precisão obtida é significativamente menor do que outras propostas encontradas na literatura.

Seo e Song [34] apresentam uma abordagem de IDS baseado em *Generative Adversarial Network* (GAN) em dados de detecção de intrusão em redes de automóveis, os autores codificam os IDs dos barramentos dos veículos em único vetor, contendo quatro grupos de ataques: DoS, FUZZY, RPM e GEAR. A abordagem atinge taxas de acurácia altíssimas. Enquanto Hajisalem e Babaie [35] apresentam um método híbrido baseado nos algoritmos de *Artificial Bee Colony* (ABC) e *Artificial Fish Swarm* (AFS), além destas, as técnicas de *Fuzzy C-Means Clustering* (FCM) e *Correlation-based Feature Selection* (CFS) são aplicadas para remoção de características irrelevantes dos conjuntos de dados UNSW-NB15 e o NSL-KDD utilizados no artigo. Os resultados parecem promissores, contudo, o trabalho carece de realizar uma avaliação da variabilidade do tráfego de rede.

Farahnakian e Heikkonen [36] apresentam uma abordagem utilizando técnicas de *Deep Auto-Encoder* com quatro auto codificadores para extrair as características de conjunto de dados e fazer a detecção de intrusão. Choi et al. [37] utilizam uma abordagem proteção de *Controller Area Network* (CAN) em veículos, propondo um novo sistema de detecção de intrusão automotiva, o trabalho aponta resultados promissores. Também foram encontrados trabalhos em Internet das Coisas (IoT), onde Jan et al. [38] abordam a detecção de intrusão com o algoritmo de SVM, para detectar invasores que tentam injetar dados em uma rede IoT. A abordagem parece promissora, podendo identificar ataques através de duas ou três características da rede. Outro trabalho encontrado foi o de Alazzam et al. [39] que utilizou seleção de características para otimizar o processo de classificação com binarização dos dados e foi capaz de melhorar a eficácia do classificador. Contudo, apesar dos trabalhos utilizarem técnicas que melhoram a detecção por meio de seleção de características e aprimoramento dos dados, nenhum trabalho busca avaliar suas técnicas ao longo do tempo ou, ainda, identificar problemas de mudança de comportamento que existem em ambientes de produção.

Utilizando o conjunto de dados CICIDS2017 Viegas et al. [40] propuseram uma avaliação utilizando *Random Forest Regressor* para identificar a variabilidade do comportamento de um conjunto de dados real, capaz de representar a mudança de

comportamento em ambiente real, mostrando resultados promissores. Entretanto, o trabalho não busca avaliar o comportamento do ambiente ao longo do tempo.

Até mesmo modelos desatualizados podem ser reutilizados para criação de novos modelos. Caminero et al. [41] utiliza a técnica de aprendizagem por reforço para incorporar o aprendizado de um modelo de aprendizagem de máquina, na tentativa de criar o ambiente de aprendizagem por reforço. Outra técnica encontrada foi no trabalho de Upadhyay et al. [42], onde os autores utilizaram uma abordagem com *Gradient Boosting Feature Selection* (GBFS) antes de aplicar o algoritmo na classificação, fazendo a seleção das melhores características para aumentar a precisão e a velocidade da execução do algoritmo. O GBFS utiliza a técnica de *Weighted Feature Importance* (WFI) para reduzir a complexibilidade dos classificadores, em seguida, o trabalho empregou diversas técnicas de aprendizagem máquina baseadas em *Decision Tree*, que mostraram uma diminuição na taxa de Falso Positivo. Apesar do grande volume de trabalhos para detecção de intrusão, na grande maioria dificilmente é encontrado algum trabalho que avalie os modelos ao longo do tempo.

A mudança de comportamento foi explorada por Zhang et al. [43] onde os autores utilizam técnicas de *transfer learning* em uma rede neural, capaz de atualizar o modelo ao longo do tempo. Contudo, o trabalho não aborda a descoberta de novos serviços. Liang e Ma [44] também mencionaram recentemente o problema da capacidade de detecção de intrusão que decai gradualmente com o surgimento de ataques desconhecidos. No entanto, eles abordam apenas os incentivos para o retreinamento do modelo, atribuindo mais tokens baseados em *blockchain* para transações que classificam pacotes suspeitos desconhecidos. Seu *framework* ainda precisa baixar o banco de dados mais recente e treinar todo o modelo, sem considerar o conhecimento anterior. Além disso, eles não compararam a frequência com que essas atualizações são necessárias para manter a precisão da detecção de intrusão, durante a utilização em ambientes de produção.

Lopez-Martin et al. [7] traz a técnica de aprendizagem por reforço profunda *Deep Reinforcement Learning* (DRL), nos conjuntos de dados NSL-KDD e AWID com quatro modelos de DRL, sendo eles: *Deep Q-Network* (DQN), *Double Deep Q-Network* (DDQN), *Policy Gradient* (PG) e *Actor-Critic* (AC), para atingir o objetivo na identificação de ataques de rede. Outro trabalho nesta linha é de Sethi et al. [45], que apresentam um NIDS adaptável com vários agentes de aprendizagem por reforço para identificar ataques de redes nos conjuntos de dados NSL-KDD, UNSW-NB15 e AWID. Todavia, os trabalhos encontrados não avaliam a mudança de comportamento e a variabilidade da classificação ao longo do tempo.

Mazini et al. [46] aplica os algoritmos *Artificial Bee Colony* (ABC) e o *AdaBoost*, para tentar obter a melhor taxa de acurácia na classificação utilizando os conjuntos de dados NSL-KDD e ISCXIDS2012 e comparando os resultados com diferentes algoritmos da literatura. O conjunto de dados ISCXIDS2012, que compõe o tráfego real do período de 7 dias do ano de 2012, foi utilizado para avaliar o comportamento ao longo do tempo. Enquanto, Van et al. [47] utiliza técnicas de *Deep Learning* com o objetivo de lidar com a mudança de tráfego de rede em NIDS. Sua abordagem é avaliada com técnicas de RBM e *Autoencoder* ao longo de 7 dias, utilizando o mesmo conjunto de dados no trabalho apresentado anteriormente.

Tabela 5. Comparação dos trabalhos relacionados.

Trabalhos	**Tratam Mudança de Comportamento?**	**Tratam Atualização do Modelo?**	**Tratam variabilidade da acurácia ao longo do tempo?**	**Utilizam Conjunto de Dados de ambientes Produção?**
Koc et al. [22]	Não, a abordagem utiliza seleção de características para aumentar a precisão da detecção.	Não	Não	Não
Longari et al. [23]	Não, a abordagem busca atingir a maior acurácia comparado a outros algoritmos da literatura.	Não	Sim, avaliam a complexibilidade ao longo do tempo.	Sim, utilizando um conjunto de produção.
Otoum et al. [24]	Não, a abordagem busca atingir a maior acurácia comparado a outros algoritmos da literatura.	Sim, a atualização ocorre através de aprendizagem por reforço.	Não	Não
Akashdeep et al. [25]	Sim, tratam a mudança através de técnicas de ganho de informação e correlação	Não	Não	Não
Hamedet al. [26]	Não, a abordagem busca atingir a maior acurácia comparado a outros algoritmos da literatura.	Não	Sim, avaliam a complexibilidade ao longo do tempo.	Não

Trabalhos	Tratam Mudança de Comportamento?	Tratam Atualização do Modelo?	Tratam variabilidade da acurácia ao longo do tempo?	Utilizam Conjunto de Dados de ambientes Produção?
Caminero et al. [41]	Não, a abordagem busca atingir a maior acurácia comparado a outros algoritmos da literatura.	Sim, a atualização ocorre através de aprendizagem por reforço.	Não	Não
Zhang et al. [43]	Não, a abordagem busca atingir a maior acurácia comparado a outros algoritmos da literatura.	Sim, utilizando classificador baseado em CNN.	Não	Não
Lopez-Martin et al. [7]	Não, a abordagem busca atingir a maior acurácia comparado a outros algoritmos da literatura.	Sim, a atualização ocorre através de aprendizagem por reforço.	Não	Sim, conjunto de dados de dispositivos WIFI
Sethi et al. [45]	Não, a abordagem busca atingir a maior acurácia comparado a outros algoritmos da literatura.	Sim, a atualização ocorre através de aprendizagem por reforço.	Não	Sim, conjunto de dados de dispositivos WIFI
Mazini et al. [46]	Sim, tentam melhorar o comportamento avaliando uma linha do tempo.	Não	Sim, avaliam a complexibilidade ao longo do tempo.	Não
Van et al. [47]	Sim, tentam melhorar o comportamento avaliando uma linha do tempo.	Não	Sim, avaliam a complexibilidade ao longo do tempo.	Não

Embora o objetivo primário imediato de muitas propostas seja fornecer a maior precisão possível na detecção de ataques de rede (por exemplo, [22] - [24]), outros objetivos surgiram, uma vez que esses ataques (e o tráfego de rede) evoluem com o tempo. Alguns exemplos de objetivos adicionais endereçados por trabalhos relacionados incluem acelerar as atualizações do modelo, utilizando seleção de características (por exemplo, [25], [26]). Contudo, os trabalhos ainda não se preocupam em lidar com a mudança de comportamento do tráfego de rede ao longo do tempo. Logo, este é um problema que ocorre diariamente com o surgimento de novos grupos de ataques e serviços disponíveis.

Diversos trabalhos têm como objetivo de identificar ataques de redes em conjuntos de dados muito antigos e simulados, que não conseguem representar a diversidade do comportamento do tráfego de rede real de um sistema de produção que ocorre diariamente (por exemplo [27] – [33]). Um conjunto de dados deve ser empiricamente avaliado para utilização em abordagens que podem ser utilizadas em um ambiente de produção.

Outros trabalhos da literatura se preocupam em encontrar a melhor seleção de características em um conjunto de dados para melhorar a precisão e a velocidade do algoritmo, não se preocupando com a diversidade do tráfego de rede. Diversos trabalhos utilizam conjuntos de dados realistas e simulados ao mesmo tempo para testar a técnica da proposta (por exemplo [7], [23], [26], [41], [43], [46], [47]). Todavia, os trabalhos buscam uma alta taxa de acurácia, velocidade nos processos de treinamento e melhora nos resultados, porém, não avaliam de maneira eficaz as mudanças do comportamento do tráfego de rede, tornando ineficazes em ambientes de produção, devido à falta de avaliação.

Embora muitos desses trabalhos alcancem altas taxas de precisão para detecção de intrusão na fase de teste, a grande maioria não avalia a métrica ao longo do tempo, fazendo com que essa abordagem não possa ser utilizada em um ambiente de grande variabilidade no tráfego de rede. Alguns deles (por exemplo, [41], [42]) mencionam que a adoção de um conjunto de classificadores como trabalho futuro pode ser uma solução capaz de aumentar as chances de detecção de ataques de rede com maior precisão. No entanto, se todos os seus modelos adotados diminuírem a precisão da detecção imediatamente após a fase de teste, este classificador de conjunto pode não ser muito útil. Visto que o impacto nas mudanças no tráfego de rede nem mesmo foram avaliadas.

Existem trabalhos que utilizam técnicas com diferentes redes neurais, capazes de atingir acurácias altíssimas (por exemplo [34], [36]) em conjuntos de dados realistas que representam a variabilidade de um ambiente de produção. Contudo, os trabalhos avaliam somente aquele

conjunto de dados como um todo, e não avaliam a variabilidade dos dados ao longo do tempo. Ficando uma dúvida de quanto é a longevidade destes modelos criados e de quanto em quanto tempo um administrador de sistema deverá atualizar os modelos.

Outros trabalhos utilizam métricas para atualização do modelo ao longo do tempo, com técnicas de *transfer learning* em aprendizagem por reforço, (por exemplo [7], [24], [41], [45]). O que torna mais fácil e rápido a atualização do modelo, entretanto, os trabalhos não avaliam a precisão da detecção dos modelos atualizados ao longo do tempo. As avaliações da variabilidade da acurácia ao longo do tempo, é algo imprescindível para verificar se o modelo é confiável durante utilização em um ambiente de produção, (por exemplo [23], [26], [46], [47]). Os autores avaliam a variabilidade da acurácia ao longo do tempo, todavia, em todos os trabalhos, o período avaliação é muito curto (de apenas uma semana), o que não permite avaliar de maneira eficaz a variabilidade das classificações ao longo do tempo.

A grande maioria dos trabalhos encontrados na literatura se preocupam com a detecção de intrusão em um único conjunto de dados, utilizando técnicas para atingir a maior taxa de precisão na identificação de ataques, além disso, diversos trabalhos trazem propostas interessantes, criando técnicas capazes de encontrar o conjunto de características mais relevante durante a classificação, aumentando a precisão na detecção e a velocidade da classificação do algoritmo. Outras abordagens se preocupam em atualizar os modelos periodicamente, fazendo com que um classificador passe a identificar novamente ataques de rede ao longo do tempo. As abordagens listadas anteriormente, dificilmente se preocupam com a variabilidade da mudança do comportamento de rede, problema este que ocorre diariamente em ambientes de produção. É necessário que as técnicas desenvolvidas para detecção de intrusão sejam avaliadas quanto à variabilidade da mudança de comportamento ao longo do tempo, além de serem avaliadas por conjuntos de dados realistas de sistemas de produção que permitem avaliar esta métrica. Por fim, é necessário a avaliação do custo de atualização, que não costuma ser avaliado na grande maioria dos trabalhos encontrados na literatura.

A proposta deste trabalho busca identificar a tendência da queda da precisão ao longo do tempo que ocorre na grande maioria das abordagens de aprendizagem existentes para NIDS [9]. Esse trabalho avaliou quatro abordagens tradicionais de aprendizagem de máquina. Ele diminuiu consideravelmente sua precisão (até 10%) logo no primeiro mês. Assim, formou uma tendência contínua onde a técnica acaba perdendo mais de 30% de sua precisão no ano avaliado. No entanto, a variabilidade da precisão, longevidade do modelo e o impacto da precisão em períodos mais extensos (por exemplo, quatro anos) ainda são desconhecidos na literatura.

Finalmente, as oportunidades abertas motivam a avaliação de métodos tradicionais com períodos mais extensos, além disso, o estudo de sua variação de precisão e longevidade do modelo para tirar conclusões mais gerais sobre essa tendência de queda na taxa de precisão para tarefas de classificação de tráfego de rede. Esses problemas/oportunidades em aberto contribuem para as metas previamente abordadas de fornecer soluções NIDS mais confiáveis e fáceis de manter para operadores de segurança.

Capítulo 4

Avaliação do Estado da Arte

Esta seção investiga os principais aspectos que tornam a detecção de intrusão baseada em rede uma tarefa desafiadora para técnicas baseadas em aprendizagem de máquina. Para tal tarefa, foi utilizado um conjunto de dados real de ambiente de produção para detecção de intrusão e avaliação do desempenho das abordagens de detecção de intrusão baseadas em aprendizagem de máquina amplamente utilizadas.

4.1 A confiança da aprendizagem de máquina para detecção de intrusão

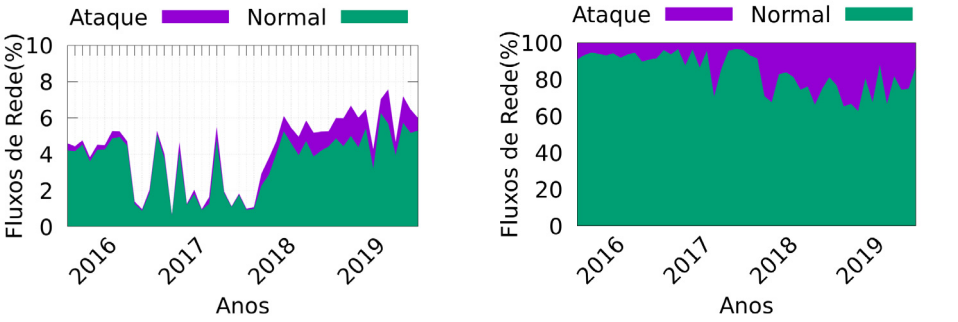
Esta avaliação visa responder as seguintes questões de pesquisa:

- (RQ1) *Qual é o comportamento das abordagens baseadas em aprendizagem de máquina, amplamente utilizadas, em termos de precisão ao longo do tempo quando nenhuma atualização do modelo é realizada*?
- (RQ2) *Qual é o impacto das atualizações periódicas do modelo na precisão das abordagens baseadas em aprendizagem de máquina, amplamente utilizadas*?

Para avaliar as abordagens baseadas em aprendizagem de máquina amplamente utilizadas foram selecionados quatro classificadores frequentemente aplicados para detecção de intrusão, o *Decision Tree* (DT), *Random Forest* (RF), *Adaboosting* (Ada) e *Gradient Boosting* (GBT). Cada classificador, com o objetivo de garantir determinismo, utilizou a mesma semente (42) durante o processo de treinamento. O classificador DT foi implementado por meio do

algoritmo C4.5, com fator de confiança de 0,25 e gini como a métrica de qualidade da divisão do nó. Os classificadores de *ensemble* (RF, Ada e GBT) foram implementados com 100 árvores de decisão, onde cada um deles também usa gini como a métrica de qualidade de divisão de nó. O classificador GBT depende de um valor de taxa de aprendizagem de 0,1, com desvio para a função de *loss*. O classificador Ada usa o SAMME como algoritmo de impulso e 1,0 como taxa de aprendizado.

A Figura 4, representa o número de fluxos de rede distribuídos ao longo do tempo no conjunto de dados *MAWIFlow* [9]. Devido à natureza desequilibrada do *MAWIFlow* [9], uma subamostra aleatória sem reposição é usada em cada conjunto de treinamento, responsável por balancear a quantidade de ocorrência entre as classes. Os classificadores foram implementados por meio da API *Scikit-Learn* v0.24. Os classificadores são avaliados quanto às suas taxas de falso positivo (FP) e falso negativo (FN), onde o FP denota a proporção de eventos normais classificados erroneamente como um ataque.



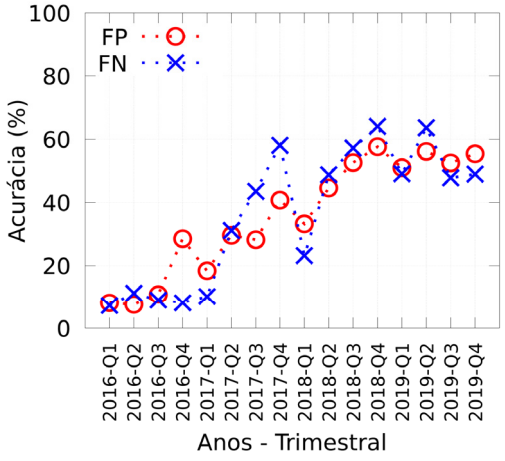
A) Números de Fluxos de Rede ao longo do tempo

B) Distribuição dos fluxos de rede (%) ao longo do tempo

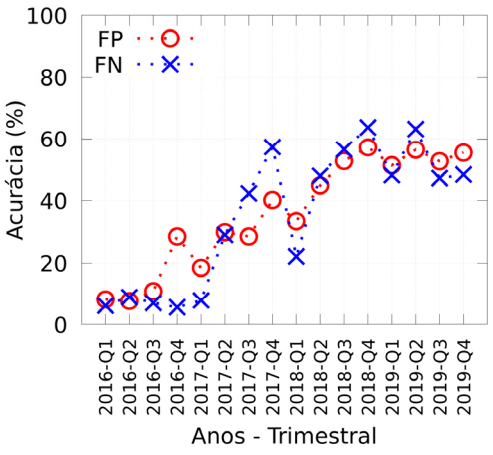
Figura 4. Distribuição do fluxo de rede *MAWIFlow* [9] ao longo dos quatro anos.

Para investigar melhor a variabilidade da precisão nas classificações ao longo do tempo, quando nenhuma atualização do modelo é realizada, foi comparado o *boxplot* das precisões obtidas no treinamento (janeiro de 2016) com aquelas obtidas no período restante. A Figura 6 mostra a distribuição da precisão (*boxplot*) de todos os classificadores de aprendizagem de máquina avaliados sem atualizações de modelo ao longo do tempo. Independentemente do classificador, os intervalos de erro aumentam significativamente com o passar do tempo em comparação com aqueles medidos durante o período de treinamento. Em média, o intervalo

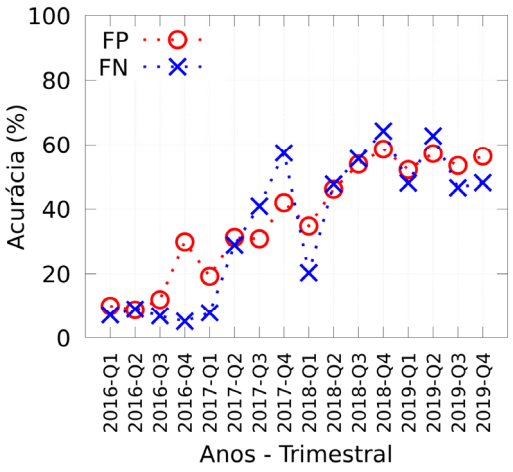
interquartil de erro de RF aumenta em 3,4% e 4,0% para as taxas de FP e FN a cada seis meses de longevidade adicional do modelo, aumentando significativamente sua variação de precisão ao longo do tempo. As técnicas avaliadas permanecem incapazes de fornecer confiança na detecção de intrusão meses após a fase de treinamento do classificador. No entanto, a distribuição da precisão aumenta com o passar do tempo, conforme observado por um aumento no número de *outliers* (Figura 6, 2016 e 2017) e os intervalos interquartis ao longo do tempo (Figura 5, 2018 e 2019). Deve-se observar que uma alta variação da taxa de erro ao longo do tempo para o mecanismo de detecção de intrusão afeta significativamente a percepção do administrador sobre a confiança do sistema. Essa falta de confiança ocorre pois, mesmo que o mecanismo de detecção de intrusão possa atingir altas taxas de precisão com modelos desatualizados, o operador irá desconsiderar os alertas assim que uma alta taxa de erro ocorrer. Consequentemente, para uma implantação de produção confiável, os mecanismos de detecção de intrusão devem atingir altas taxas de precisão e apresentar baixa variabilidade de precisão ao longo do tempo.



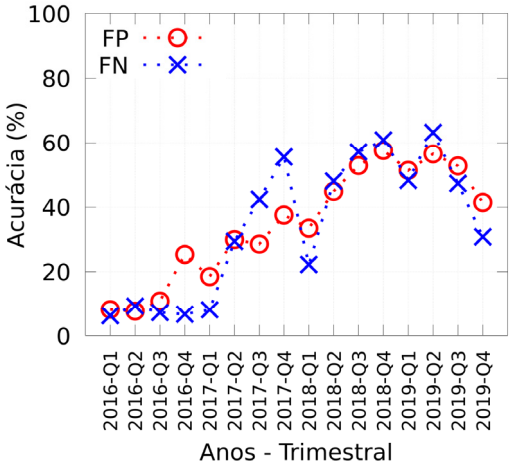
A) *Decision Tree*



B) *Random Forest*



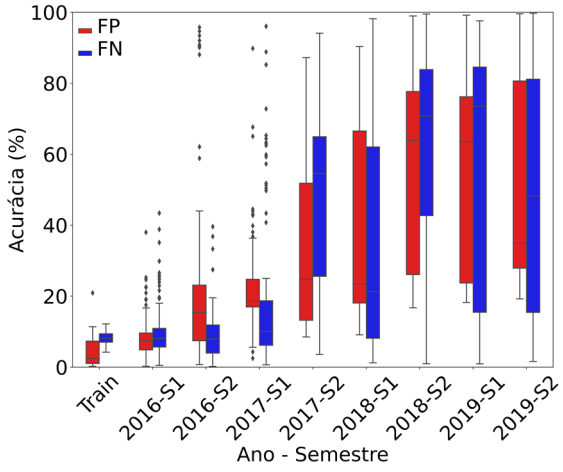
C) *Adaboosting*



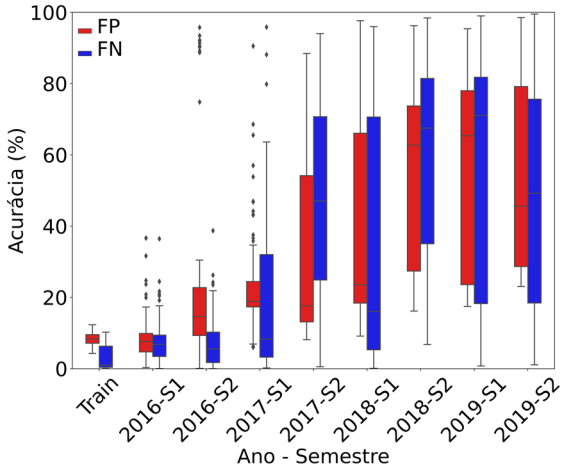
D) *Gradient Boosting*

Figura 5. Desempenho de acurácia ao longo do tempo, em uma base trimestral de vários algoritmos de aprendizagem de máquina, em todo o conjunto de dados MAWIFlow [9]. Os classificadores são treinados com dados de janeiro de 2016 e não são atualizados ao longo do tempo.

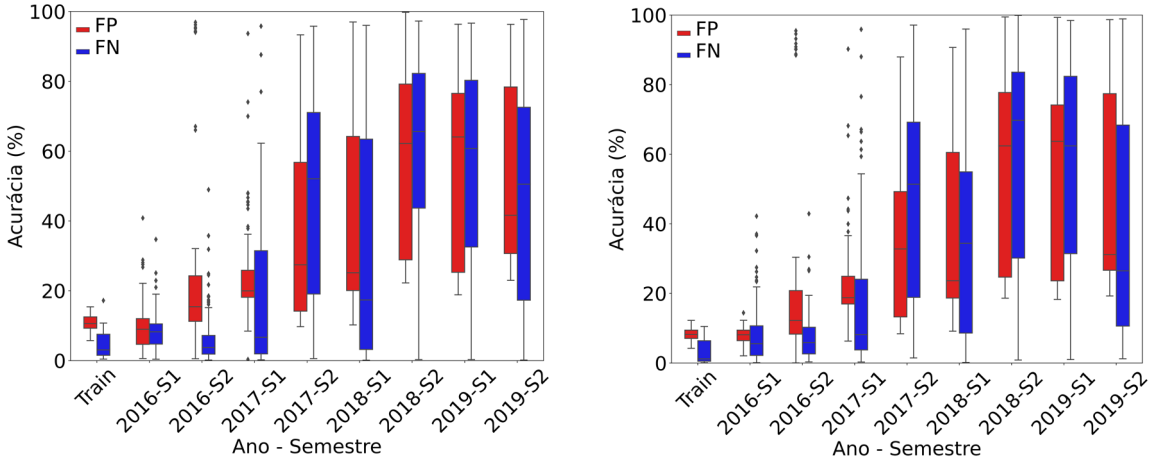
Os classificadores de aprendizagem de máquina avaliados não são capazes de lidar com o comportamento do tráfego de rede ao longo do tempo se os modelos não forem atualizados. Contudo, as técnicas tornam-se pouco confiáveis nos meses imediatos após o período de treinamento, conforme observado por taxas de erro mais altas (Figura 5) e maior variabilidade nas taxas de erro medidas (Figura 6). Portanto, as abordagens de detecção de intrusão de aprendizagem de máquina frequentemente usadas em ambientes de produção devem ser atualizadas regularmente para manter a confiança da classificação.



A) *Decision Tree*



B) *Random Forest*



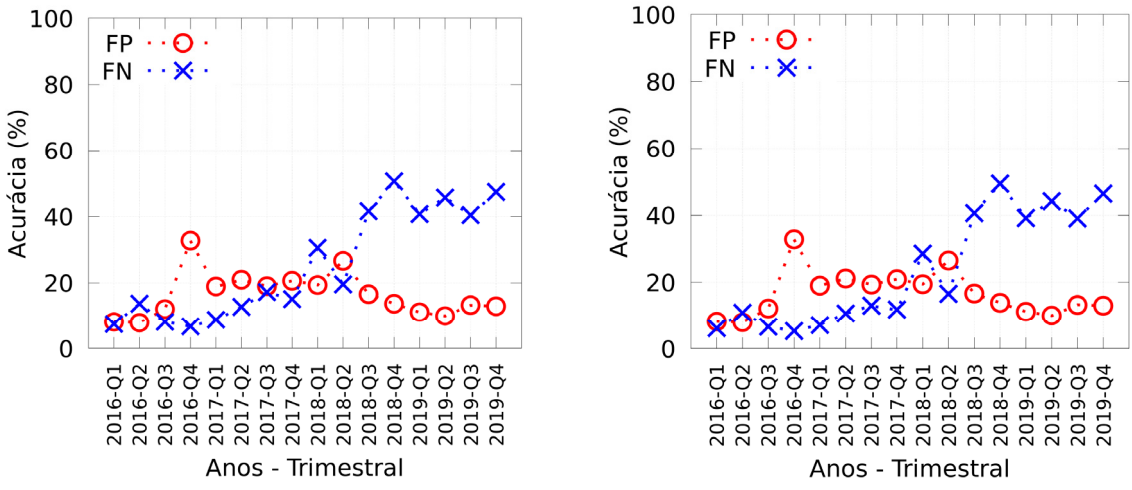
C) *Adaboosting* D) *Gradient Boosting*

Figura 6. Distribuição semestral das acurácias diárias para vários algoritmos aprendizagem de máquina em todo o conjunto de dados *MAWIFlow* [9]. Os classificadores são treinados com dados de janeiro de 2016 e não são atualizados ao longo do tempo.

Para responder ao RQ2, foi investigado o impacto da acurácia que as atualizações periódicas do modelo causam nos modelos de aprendizagem de máquina sem atualizar. Primeiro, foi considerado um modelo de longevidade de 6 meses e 1 mês como um período de treinamento, o que significa que houve a atualização dos modelos de aprendizagem de máquina para cada semestre utilizando os dados *MAWIFlow* [9], que ocorreram um mês antes. Os modelos de aprendizagem de máquina são atualizados no final de cada janeiro e julho em qualquer ano - usando os dados que ocorreram nos últimos 30 dias naquele determinado período. É importante mencionar que as atualizações frequentes do modelo devem ser estabelecidas de acordo com as necessidades do administrador.

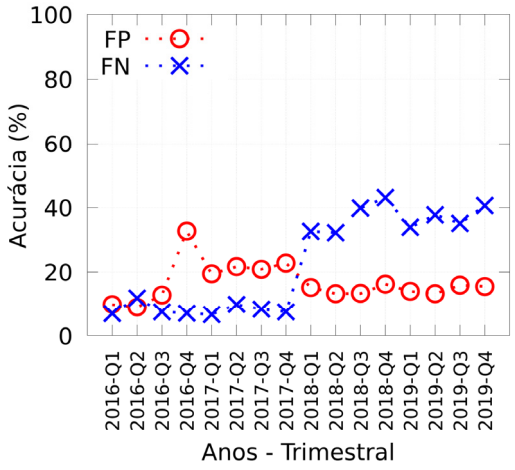
A Figura 7 mostra o desempenho da acurácia em todos os dados *MAWIFlow* [9] quando as atualizações do modelo são realizadas a cada 6 meses. Percebe-se que, de maneira geral, o desempenho de precisão de todos os classificadores avaliados melhora significativamente quando comparado com o resultado sem atualizações. Por exemplo, o classificador de RF atualizado periodicamente melhorou sua taxa média de FP de 35,1% para 16,5%, enquanto também melhorou sua taxa média de FN de 36,0% para 23,4% em comparação com os resultados sem atualizações do modelo. No entanto, a longevidade do modelo varia ao longo do tempo, conforme observado por taxas mais altas de FP e FN em 2018 e 2019. Nesse caso, os modelos de aprendizagem de máquina avaliados deveriam ter sido atualizados com mais frequência para aumentar a taxa de precisão na classificação.

Devido às dificuldades relacionadas à identificação dos modelos desatualizados, deverá aumentar a periodicidade de sua atualização, independentemente se o modelo atual ainda é confiável ou não, como ocorreu de 2016 a 2017. Também foi realizada a investigação da variabilidade da acurácia quando o modelo obtém taxa de erro média medida como taxas de FP e FN em todas as atualizações de dados do *MAWIFlow* [9] realizadas. A Figura 8 mostra a distribuição de precisão (*boxplot*) dos classificadores de aprendizagem de máquina avaliados quando as atualizações do modelo são realizadas a cada semestre. As atualizações periódicas do modelo também diminuem a variação da precisão significativamente ao longo do tempo, conforme observado por um intervalo interquartil inferior em 2016 e 2017. Em 2018 e 2019 (o período que exige um modelo com atualizações mais frequente), o intervalo interquartil aumenta por uma variabilidade maior nas classificações sem atualizações. Consequentemente, os modelos de aprendizagem de máquina desatualizados, possuem uma longevidade de 6 meses e podem diminuir significativamente e variar sua precisão, afetando a confiança da detecção de intrusão.

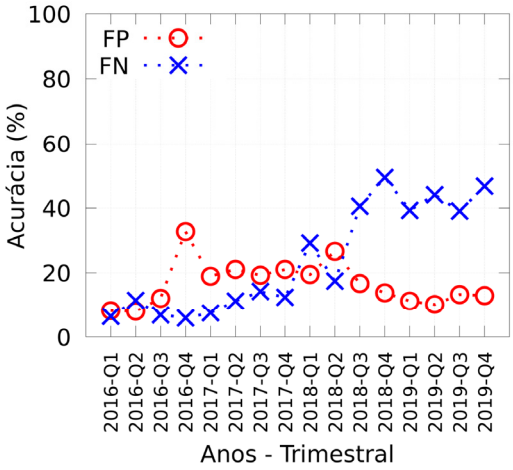


A) *Decision Tree*

B) *Random Forest*

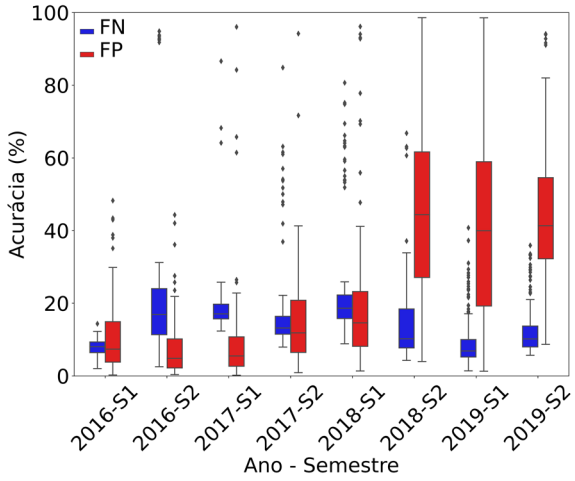


C) *Adaboosting*

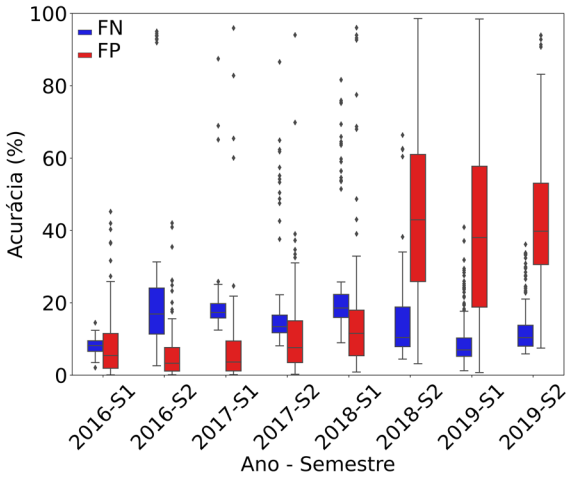


D) *Gradient Boosting*

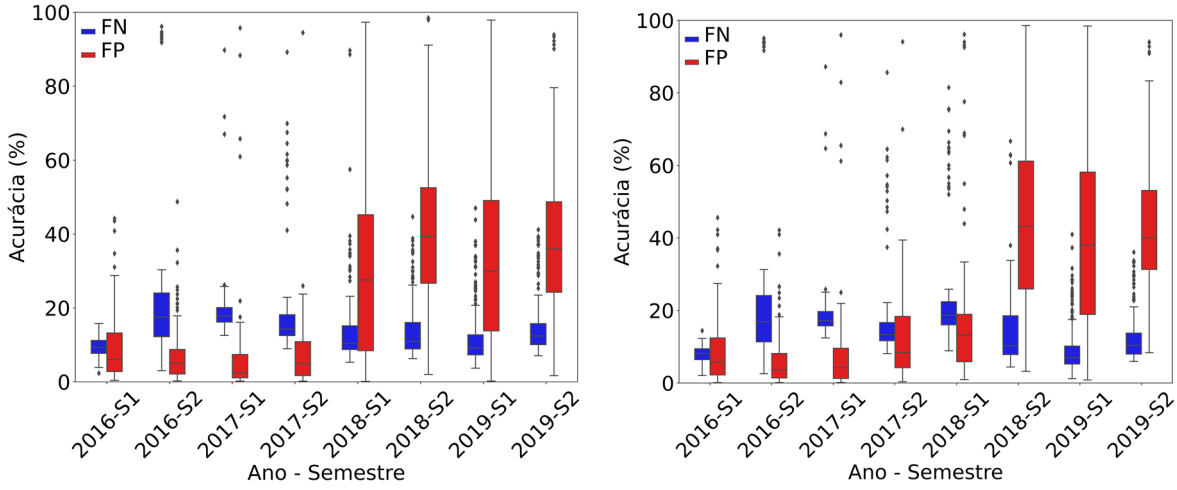
Figura 7. Desempenho de acurácia ao longo do tempo em uma base trimestral de vários algoritmos de aprendizagem de máquina em todo o conjunto de dados *MAWIFlow* [9]. Os classificadores são atualizados a cada intervalo de 6 meses, com 1 mês de dados de treinamento.



A) *Decision Tree*



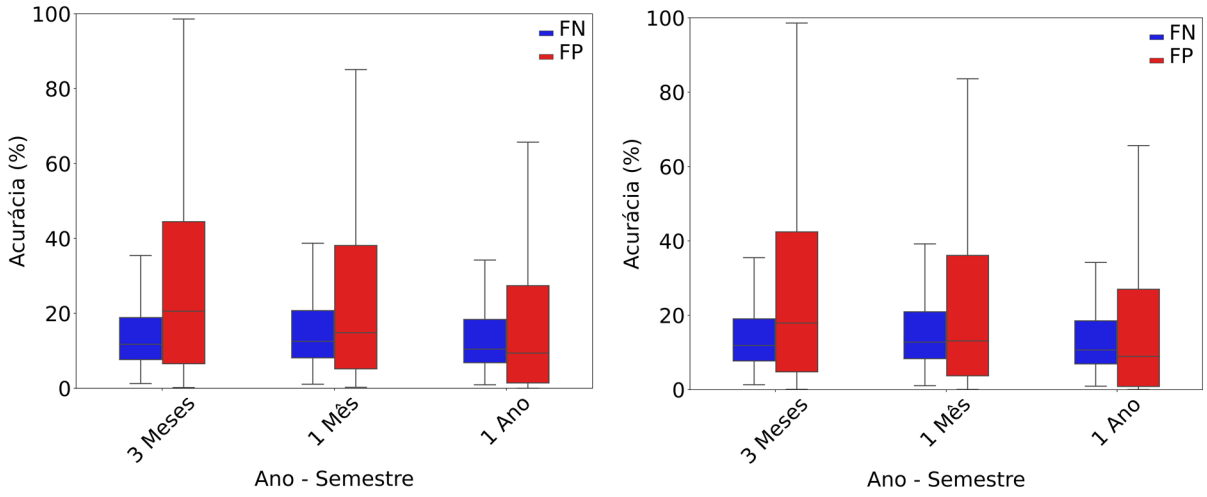
B) *Random Forest*



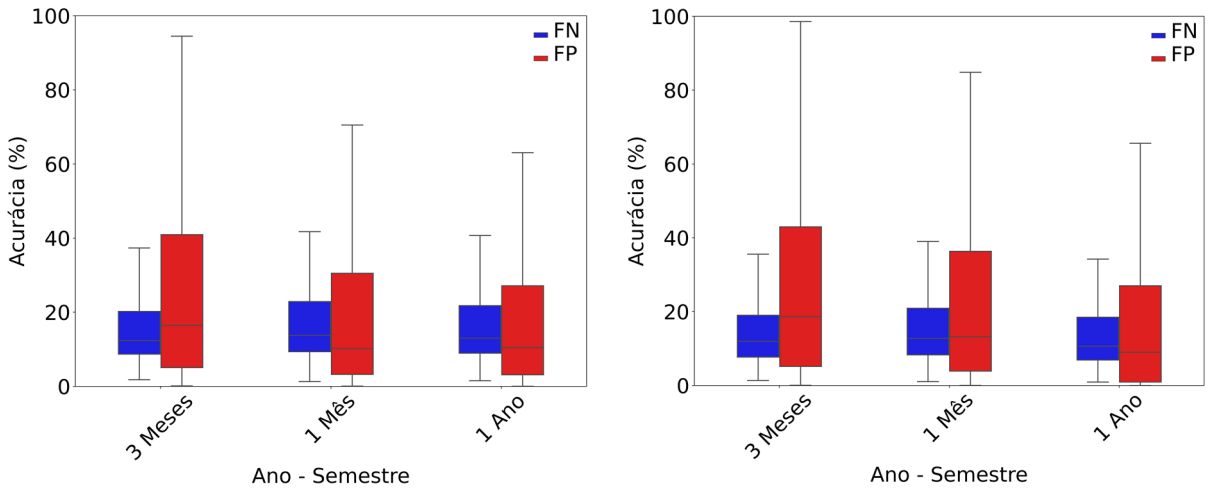
C) *Adaboosting* D) *Gradient Boosting*

Figura 8. Distribuição semestral das acurácias diárias ao longo do tempo de vários algoritmos de aprendizagem de máquina em todo o conjunto de dados *MAWIFlow* [9]. Os classificadores são atualizados a cada intervalo de 6 meses, com um valor de 1 mês de dados de treinamento.

Por fim, é avaliado como as atualizações de aprendizagem de máquina mais frequentes afetam a acurácia do classificador do modelo e sua variação ao longo do tempo. Para este fim, foi calculado a taxa média de erro, medida através das taxas FP e FN, enquanto é variado a frequência de atualização do modelo no *MAWIFlow* [9]. A Figura 9 mostra a longevidade do modelo e a comparação da taxa de erro média para os classificadores avaliados. Visivelmente, uma longevidade de modelo mais baixa não melhora significativamente a taxa de erro média, porém, diminui a variação de precisão, conforme observado pelos intervalos interquartis mais baixos. Portanto, as abordagens tradicionais de aprendizagem de máquina exigem atualizações de modelo mais frequentes para fornecer uma classificação confiável.



A) *Decision Tree* B) *Random Forest*



C) *Adaboosting* D) *Gradient Boosting*

Figura 9. Longevidade do modelo e compensação de taxa de erro média no conjunto de dados *MAWIFlow*. A longevidade do modelo estabelece a frequência de atualização do modelo, enquanto a taxa média de erro é medida como a média das taxas FP e FN em todos os dados *MAWIFlow* [9].

4.2 Discussão

O comportamento de rede ao longo do tempo é desconhecido, as mudanças na variabilidade do tráfego de rede são frequentemente negligenciadas na literatura. Em geral, os autores contam com tráfego de rede constante, que fornece conjuntos de dados que não representam a natureza dinâmica nos ambientes de produção. Como consequência, os modelos obtidos são imprecisos e, em muitos casos, irreais. Ao contrário da literatura, o conjunto de dados *MAWIFlow* [9] é composto por mais de 8 TB de dados que abrangem quatro anos de tráfego de rede. Este conjunto de dados permitiu avaliar as abordagens atuais na literatura quanto à sua confiança ao longo do tempo. Surpreendentemente, as técnicas amplamente utilizadas são incapazes de lidar com o comportamento ao longo do tempo do tráfego de rede. Sua precisão de detecção diminui significativamente com o tempo, enquanto sua variação aumenta com o passar do tempo. A execução de atualizações periódicas do modelo, apesar dos desafios que podem representar, pode afetar a precisão negativamente e diminuir sua variação. As abordagens da literatura atual devem ser usadas com um modelo de longevidade pequeno (por exemplo, alguns dias ou semanas), para fornecer confiança ao longo do tempo, devido à

expectativa de pequena longevidade do modelo, as técnicas propostas anteriormente raramente são utilizadas em ambientes de produção. O principal motivo é que os modelos de aprendizagem de máquina com uma longevidade tão pequena, não podem ser atualizados com a frequência necessária.

Capítulo 5

Proposta

Nesta seção, é apresentado um novo modelo de aprendizado por reforço para detecção de intrusão confiável, mesmo em ambientes com variabilidade no tráfego de rede. O objetivo principal é aumentar a longevidade do modelo para manter a confiança do sistema por um período maior, sem a necessidade de atualizações frequentes do modelo. Além disso, a proposta pretende diminuir o custo humano e computacional necessários para atualizar os modelos periodicamente. A Figura 10 mostra uma visão geral do modelo proposto, que está organizado em dois estágios principais, a Classificação e o Treinamento – Construção do Agente Confiável.

A classificação dos eventos de rede é realizada por meio da técnica de aprendizagem por reforço. Seu objetivo é classificar os eventos de rede ao longo do tempo como normal ou ataque. Os eventos de rede são descritos como um estado do ambiente, enquanto o agente de aprendizagem por reforço emite uma ação que representa a classe do evento de acordo com a política de aprendizagem por reforço.

O desenvolvimento do agente é realizado em duas etapas. Primeiro, quando o sistema é implantado pela primeira vez criando um modelo novo ou para a atualização de um modelo já existente. Na fase de implantação, o treinamento é executado em uma abordagem de aprendizagem por reforço com um agente que é construído do zero. Enquanto, a atualização do modelo pode diminuir significativamente os custos computacionais usando o modelo de produção atual através da técnica de *transfer learning* para atualizar o modelo de aprendizagem por reforço. Para a tarefa de atualização, apenas uma subamostra do conjunto de eventos de rede é utilizada, pois a técnica aproveita o conhecimento prévio do modelo atual. Portanto, ele aplica um mecanismo de janela deslizante nos eventos de rede do ambiente de produção. Por

exemplo, atualizando o agente com eventos da última semana e gerando o novo modelo de aprendizagem por reforço. Na segunda etapa, a abordagem cria políticas para o agente, através de um novo algoritmo de aprendizagem por reforço que visa estender a longevidade do modelo. A proposta pode aumentar a longevidade, por meio da exatidão do modelo, ao invés da sua acurácia.

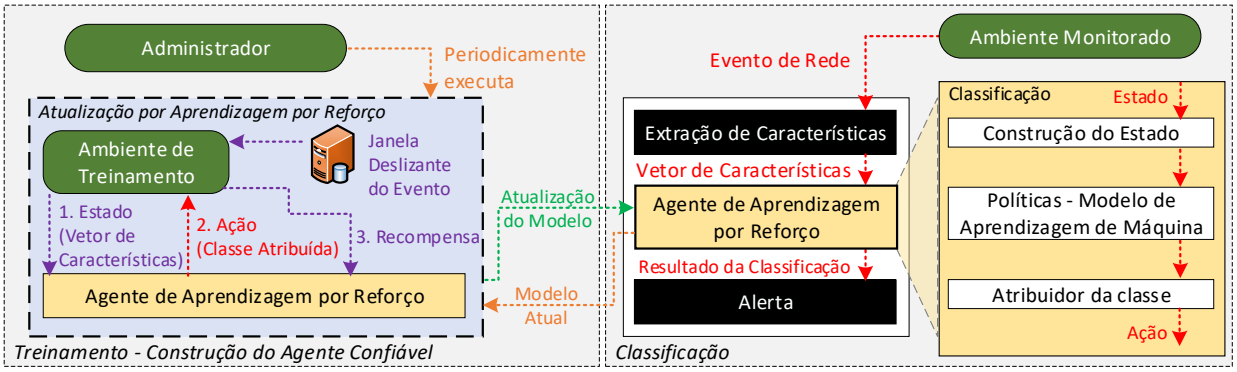


Figura 10. Visão geral da proposta com o modelo de detecção de intrusão de aprendizagem por reforço para classificação e atualização do modelo.

5.1 Treinamento - Criação do Agente Confiável

As técnicas de aprendizagem por reforço têm produzido resultados promissores em diversos campos onde as recompensas dos agentes podem ser coletadas de forma autônoma, como uma consequência direta de uma determinada ação. No entanto, em implantações de produção, os sistemas de detecção de intrusão podem não obter facilmente o rótulo do evento correto e as recompensas podem não ser avaliadas. Em outras palavras, uma ação pode ser realizada, mas as recompensas podem não ser atribuídas apropriadamente. Durante a construção do modelo, os sistemas de detecção de intrusão devem ter acesso prévio aos rótulos dos eventos, permitindo obter as recompensas adequadas. O objetivo principal deste modelo é alavancar a técnica de aprendizagem por reforço para construir uma política em que o agente pode ser utilizado para criar modelos com longevidade, mesmo em ambientes de produção. A abordagem da construção de agente confiável lida com a detecção de intrusão como uma tarefa de aprendizagem por reforço, fazendo uso das seguintes relações:

- **Ambiente Simulado**: Conjunto de dados de treinamento de detecção de intrusão real, contendo um conjunto de eventos de rede com mudanças de comportamento de rede ao longo do tempo que ocorrem em ambientes de produção.
- **Estado**: Evento passado ao ambiente de treinamento, que deve ser analisado pelo agente de detecção de intrusão. Pode ser um evento normal ou de ataque de rede.
- **Ação**: O resultado da classificação do agente de um determinado estado do ambiente como normal ou ataque.
- **Recompensa**: A recompensa que um agente recebe por uma determinada ação gerada de acordo com o estado atual do ambiente.

O presente trabalho traz a detecção de intrusão como uma tarefa de aprendizagem por reforço, demonstrada na Figura 10 (Treinamento - Construção do agente confiável). Pode-se notar que um agente recebe um estado do ambiente de treinamento (ou seja, um evento de rede) como entrada. Este atua utilizando uma classe atribuída ao evento e recebe uma respectiva recompensa, baseada na distância da classe correta do evento. O procedimento de treinamento pode ser executado usando um novo agente (por exemplo, a cada semestre) contendo o conhecimento dos modelos treinados anteriormente. Deve-se observar que a periodicidade da atualização do modelo deve ser definida de acordo com as necessidades do administrador. Para tal, a abordagem conta com uma janela deslizante de eventos para facilitar a atualização do modelo. Assim, menos eventos podem ser utilizados nas atualizações do modelo se o conhecimento prévio do comportamento do ambiente estiver disponível. Portanto, aproveita-se o modelo desatualizado para representar o histórico do comportamento do ambiente, enquanto o atualiza por meio da técnica de *transfer learning*. Como resultado, as atualizações do modelo podem ser realizadas como uma simples adaptação da política do agente, ao invés de construí-la novamente do zero. O processo de atualização do modelo pode ser realizado com menos eventos, exigindo menos intervenção humana para o procedimento de rotulagem dos eventos de rede e custos computacionais no processo de treinamento.

Além da facilidade de atualização do modelo, o desafio da longevidade dele também deve ser abordado. Portanto, a presente proposta depende do uso de métricas de exatidão para a classificação das recompensas, na fase de treinamento para estender a longevidade do modelo de aprendizagem por reforço. O cálculo da recompensa do agente é mostrado na Equação 2, onde a confiança institucional representa a confiança da classificação do modelo de

aprendizagem de máquina em uma determinada instância institucional, FP denota falsos positivos e FN denota falsos negativos.

$$recompensa = \begin{cases} 1 - confiança^{\,Institucional}, & se\ FP\ ou\ FN \\ confiança^{\,Institucional}, & se\ não \end{cases}$$

Equação 2: Cálculo da Confiança

Classificações confiáveis em eventos classificados incorretamente recebem menos recompensas, enquanto as classificadas corretamente recebem sua confiança de classificação como uma recompensa. Assim, as recompensas são calculadas como uma medida de aproximação da confiança do classificador com a classe do evento correto. Os valores de confiança do classificador são independentes do algoritmo utilizado para classificação. Por exemplo, o classificador *Random Forest* produz seus valores de confiança com a proporção de suas árvores de decisão que, por sua vez, classificaram uma determinada instância como a classe atribuída ao evento.

A proposta visa aumentar a longevidade do modelo e, ao mesmo tempo, diminuir a variação da precisão por meio da exatidão na classificação, que é medida através dos valores de confiança do classificador, ao invés de apenas buscar uma precisão maior. Como resultado, os modelos de aprendizagem de máquina serão construídos para melhorar a exatidão da classificação em todos os eventos de entrada, ao invés de apenas aumentar a precisão em um subconjunto de dados (resultando em menos longevidade causada por *overfitting* no modelo de aprendizagem por reforço).

Para implementar o modelo de aprendizagem por reforço para detecção de intrusão e as medidas de avaliação para recompensa proposta, foi desenvolvido uma versão do conhecido algoritmo *Q-Learning* de aprendizagem por reforço [48], capaz de analisar o desempenho do agente ao longo do treinamento e retornar um *feedback* para melhorar as ações. A implementação, é executada em cada treinamento do modelo ou da atualização (Figura 10, Treinamento - Construção de Agente Confiável),

A Figura 11 mostra o pseudocódigo do algoritmo de *Q-learning* proposto, que é semelhante ao algoritmo tradicional de *Q-Learning*. Inicialmente, o algoritmo recebe um conjunto de dados de treinamento, estes dados serão utilizados para as atuações do agente no ambiente simulado (S). Em seguida, é inicializado um agente aleatoriamente, na primeira implantação do modelo (ou seja, criação de um modelo novo), ou com um agente desatualizado

(ou seja, atualização do modelo), em um processo de atualização de modelo que segue a técnica de *transfer learning*. O algoritmo é executado até que tenha convergido, atingindo, um nível de acurácia esperado definido pelo especialista. Em resumo, o algoritmo seleciona aleatoriamente em cada iteração um estado (no caso, uma instância) do conjunto de dados de treinamento (ou seja, um ambiente simulado), produz uma ação relacionada (ou seja, confiança com base no treino) e recebe uma recompensa através do cálculo da Equação 2. Ou seja, ele aproxima os valores de confiança do modelo de classificação aos rótulos dos eventos corretos para maximizar suas recompensas ao longo do tempo. Dessa forma, melhora a capacidade de generalização sobre todos os dados de treinamento (ambiente simulado) e aumenta a longevidade do modelo.

Require:
States $S = \{instance_1, \ldots, instance_n\}$ ▷ Conjunto de dados
Actions $A = \{normal, attack\}$ ▷ aprendizagem de máquina (classes)
Função da recompensa $R : S \times A \rightarrow \mathbb{R}$ ▷ Função para calcular a recompensa do agente, conforme Equação 1
Função de transição $R : S \times A \rightarrow \mathbb{S}$
taxa de aprendizagem $\alpha \in [0, 1]$, tipicamente $\alpha = 0.1$
Fator de desconto $\gamma \in [0, 1]$
procedure QLEARNING(S, A, R, T, α, γ)
  Initialize $Q : S \times A \rightarrow \mathbb{R}$ agente novo ou de um modelo desatualizado
  while Enquanto Q não convergir **do**
    inicia um estado aleatório $s \in S$
    while s **do**
      Calcula o valor de π de acordo com o Q e a estratégia de exploração (e.g. $\pi(x) \leftarrow \arg\max_a Q(x, a)$)
      $a \leftarrow \pi(s)$ ▷ Estabelece o estado de confiança da classificação
      $r \leftarrow RewardFunction(s, a)$ ▷ Recebe a recompensa por meio da Equação 1
      $s' \leftarrow GetInstance(instance_i + 1)$ ▷ Recebe um novo estado
      $Q(s', a) \leftarrow (1 - \alpha) \cdot Q(s, a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s', a'))$
      $s \leftarrow s'$
    end while
  end while
  return Q
end procedure

Figura 11. Pseudocódigo – (Algoritmo de *Q-learning*) proposto para detecção de intrusão, responsável por criar um agente de aprendizagem por reforço de maneira confiável.

5.2 Classificação

O funcionamento da proposta para a classificação confiável de eventos em implantações de sistemas de produção é mostrado na Figura 10 (Classificação). O algoritmo recebe um evento de tráfego de rede como entrada do ambiente simulado monitorado (por exemplo, um pacote de rede). Os dados coletados são representados como um vetor, extraído através de um módulo

de extração de características (por exemplo, extração de várias características baseadas em fluxo de rede, conforme mostrado na Tabela 4).

O vetor de características é fornecido como entrada para o agente de aprendizagem por reforço confiável implantado em um ambiente simulado. O processo de classificação faz a construção do estado (ou seja, recebe um evento de rede). O estado é enviado para o algoritmo de aprendizagem de máquina contendo as características de um evento de rede. Este estado recebe as políticas do sistema (por exemplo, uma rede neural gerando probabilidades que serão utilizadas como políticas do sistema), aplica a política do agente com seu modelo de aprendizagem de máquina. A política produz a ação (valores de confiança) e o resultado da classificação pode ser estabelecido (conforme a Figura 10, Atribuidor da classe). Por fim, um alerta pode ser gerado se o evento for classificado como um ataque.

5.3 Discussão

O comportamento ao longo do tempo do tráfego de rede é um desafio para a detecção de intrusão baseada em aprendizagem de máquina. A proposta visa abordar três aspectos principais relacionados às mudanças no tráfego da rede: as atualizações do modelo, sua longevidade e a variação da precisão.

Além disso, a proposta aproveita o conhecimento prévio de modelos desatualizados sobre o ambiente, para facilitar a atualização do modelo utilizando *transfer learning*. A hipótese é que agentes desatualizados podem ser usados em tarefas de atualização de modelo para diminuir significativamente os custos computacionais, a quantidade de dados de treinamento necessários, fazendo o modelo convergir mais rapidamente e, também, os custos relacionados à rotulagem de eventos que normalmente são alcançados por meio de auxílio de um especialista ou técnicas de aprendizagem não supervisionada.

A exatidão do modelo proposto visa aumentar sua longevidade ao longo do tempo. Portanto, a proposta busca a aproximação da precisão do modelo, para que todos os eventos utilizados no treinamento possam aumentar sua longevidade. Sendo assim, a proposta não favorece apenas um subconjunto de eventos, como é comumente feito na literatura por abordagens baseadas em detecção de intrusão, para atingir acurácias altíssimas, logo a proposta busca atingir o objetivo de criar modelos com maior longevidade, precisão na classificação e menor variação dos resultados. O resultado notável da presente proposta é a variação da

precisão diminuída pela longevidade do modelo treinado, com o objetivo de adquirir maior precisão do modelo em todos os dados de treinamento.

A criação do modelo por meio de ações do agente no ambiente simulado, possibilita que um agente aprenda através de tentativa e erro. Estas ações ampliam o conhecimento do agente e aumentam a efetiva precisão do algoritmo. Além disso, a abordagem por aprendizagem por reforço, utiliza políticas que se baseiam nas atividades anteriores do agente, todavia, geram ações aleatórias, permitindo que um agente descubra novos caminhos e realize novas tentativas em um ambiente simulado de detecção de intrusão, em busca de resultados com maior exatidão e menor variabilidade nas detecções de intrusão de rede.

O algoritmo de *Q-learning* empregado na proposta permite que o aprendizado do agente evolua ao longo do tempo, pois em cada iteração, o algoritmo observa quais foram as melhores e piores ações realizadas pelo agente. Assim, as ações são armazenadas e o algoritmo calcula novas ações derivadas das melhores ações armazenadas que podem agregar no aprendizado do agente.

Capítulo 6

Avaliação

A avaliação do modelo proposto visa responder às seguintes questões de pesquisa:

- (RQ3) *Como o modelo proposto se comporta sem as atualizações ao longo do tempo*?
- (RQ4) *Qual é o impacto das atualizações do modelo na proposta*?
- (RQ5) *O modelo proposto pode fornecer maior confiança e menor variabilidade do que as técnicas tradicionais*?

As seções a seguir descrevem como foi construído o presente modelo e seu desempenho no conjunto de dados *MAWIFlow* [9].

6.1 Construção de modelo

A proposta de aprendizagem por reforço para detecção de intrusão (Figura 8 - algoritmo de *Q-learning*) foi implementada utilizando a API OpenAI Gym [49]. A cada treinamento ou atualização do modelo, o algoritmo cria um ambiente simulado de teste que reproduz o procedimento do treinamento proposto (ver Seção 5.1). A proposta se baseia em um *MultiLayer Perceptron* (MLP) como uma política de aprendizagem por reforço (ou seja, modelo de aprendizagem de máquina).

O algoritmo escolhido permite que o procedimento para realizar a atualização do modelo desatualizado proposto (Seção 5.1) seja executado utilizando as probabilidades dos neurônios do MLP mais antigos, por meio de um procedimento de *transfer learning*. Para tal procedimento, foram utilizados os dados de janeiro de 2016 (30 dias) para o primeiro

treinamento do modelo, que possibilita a avaliação subsequente das atualizações da proposta, que contará com o agente desatualizado.

A atualização do modelo depende de apenas 7 dias de dados (Figura 10, Janela deslizante dos eventos de rede). O MLP junto do algoritmo proposto é executado em cada atualização do modelo com uma taxa de aprendizado de 0,3, uma taxa de retro propagação de 0,2 e utiliza a API TensorFlow [50] para criação do algoritmo. O algoritmo executa 5.000 épocas para construir o modelo inicial, onde cada época executa 100 voltas, que enviam ao ambiente simulado 10.000 instâncias de rede em cada volta.

A cada período, o algoritmo calcula os gradientes da política do algoritmo proposto de acordo com as recompensas obtidas durante o treinamento (ver Equação 2), a partir da classificação de 10 mil instâncias de treinamento. Como um critério de convergência durante a atualização do modelo, o algoritmo executa 5000 épocas ou atinge 90% de precisão. Esses parâmetros foram identificados empiricamente, sua variação resulta em resultados de classificação semelhantes.

Para criação dos modelos, execução dos testes e avaliação dos resultados, foi utilizado o conjunto de dados *MAWIFlow* [9], publicamente disponível e de tráfego de rede real de ambiente de produção. Contudo, diferente do trabalho *MAWIFlow* [9], que contempla apenas o tráfego de 1 ano de dados, este trabalho trouxe para avaliação o mesmo processo de extração de características e rotulação da base (ver Seção 2.3), porém com o período de quatro anos (2016-2019) de tráfego de rede.

6.2 Longevidade do modelo de classificação

O primeiro experimento visa responder RQ3 e avaliar o desempenho do modelo proposto ao longo do tempo quando nenhuma atualização é realizada. Semelhante ao que foi feito anteriormente, o algoritmo proposto é aplicado para criar uma política para o agente utilizando os dados de janeiro de 2016 dos dados do *MAWIFlow* [9], como ambiente de treinamento (Figura 10). O modelo gerado pelo agente é utilizado em todo o conjunto de dados do *MAWIFlow* [9] sem atualizações de modelo. A Figura 12 mostra o desempenho do erro médio considerando as taxas de FP e FN do modelo proposto em uma base de quatro anos, com os gráficos apresentando as médias de três meses.

O modelo proposto manteve sua longevidade e confiança por períodos mais extensos, mantendo suas taxas de FP mais próximas das medidas na fase de treinamento do classificador ao longo dos quatro anos. Por exemplo, a abordagem proposta apresentou uma taxa média de erro de 18,9% e 8,8% para FP e FN considerando o primeiro ano de implantação (2016).

Assim a proposta mostra que a cada mês após o período de treinamento, em média, há um aumento de 4,2% e 0,3% nas taxas de FP e FN, respectivamente, considerando um modelo de longevidade de 1 ano. Além disso, o modelo proposto apresentou taxas médias de FP e FN ao longo dos quatro anos de 16,4% e 23,5%. O modelo atingiu taxas de acurácia semelhantes às obtidas por técnicas tradicionais, como o classificador RF com longevidade de 6 meses (Figura 7), que apresentou 16,5% e 23,4% de taxas de FP e FN. Portanto, o modelo forneceu uma alta taxa de precisão de detecção de intrusão, mesmo quando nenhuma atualização ocorre, com taxas de precisão semelhantes obtidas pelas técnicas da literatura com um modelo de longevidade de 6 meses.

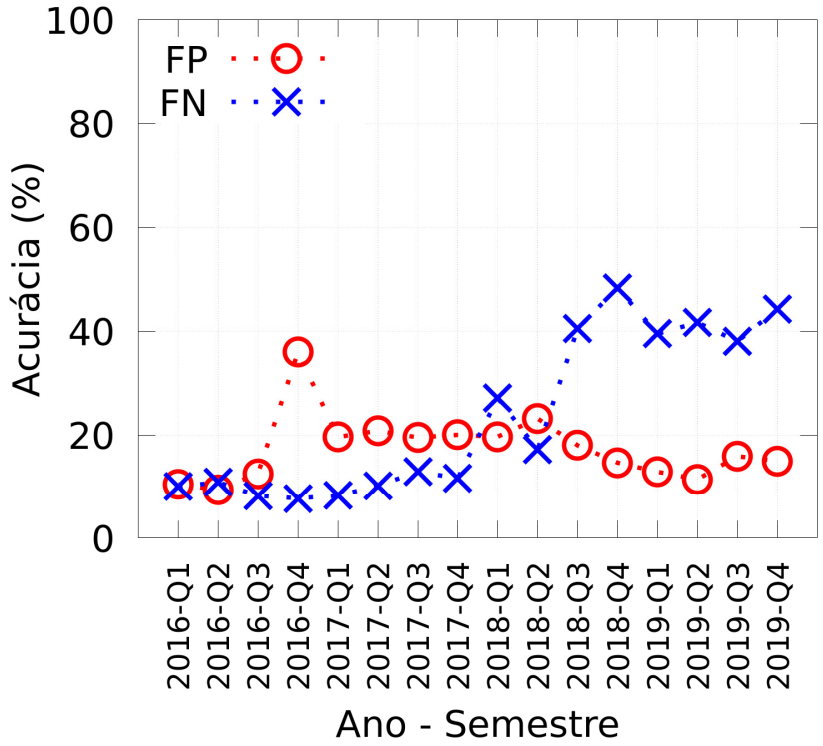


Figura 12. Desempenho de acurácia ao longo do tempo em uma base trimestral da proposta em todo o conjunto de dados MAWIFlow [9]. A proposta é treinada com dados de janeiro de 2016 e não é atualizada ao longo do tempo.

Também foi avaliada a variação do modelo proposto ao longo do tempo sem atualizações do modelo, conforme mostrado na Figura 13. A variação da taxa de precisão é significativamente menor do que a das abordagens tradicionais de aprendizagem de máquina. O método proposto apresentou intervalos interquartílicos médios em 2016-S1 de apenas 3,4% e 1,9% de FP e FN, respectivamente. Em contraste, as abordagens tradicionais (por exemplo, RF) apresentaram um intervalo interquartil médio em 2016-S1 de 3,1% e 6,1% das taxas de FP e FN, respectivamente. Além disso, se for considerado todo o *MAWIFlow* [9] com quatro anos de tráfego, a proposta apresenta um intervalo interquartil médio de 13,5% e 13,3% das taxas de FP e FN, enquanto o classificador RF apresenta 29,3% e 36,6% de intervalo do interquartil (Figura 6), respectivamente em ambos os casos. Consequentemente, a abordagem proposta aumenta a longevidade dos modelos de detecção de intrusão, melhorando a precisão dos sistemas e reduzindo a variação da precisão do modelo ao longo do tempo.

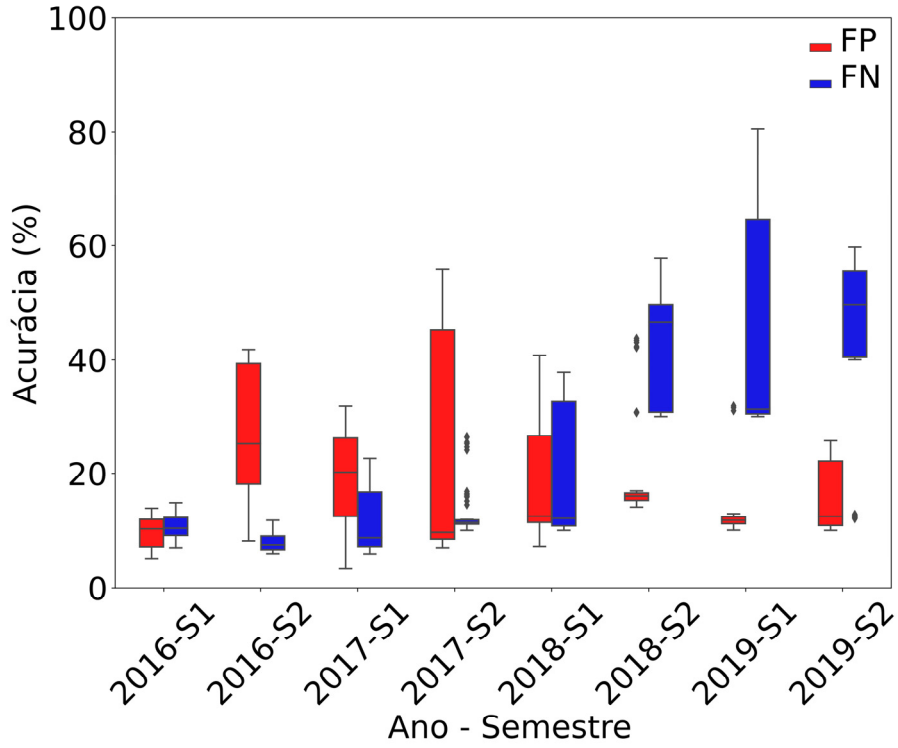


Figura 13. Distribuição semestral das acurácias da proposta em todo o conjunto de dados *MAWIFlow* [9]. A proposta é treinada com dados de janeiro de 2016 e não é atualizada ao longo do tempo.

Para responder à pergunta RQ4, foi realizado uma atualização periódica do modelo em na abordagem proposta. Nesse caso, de forma semelhante aos experimentos realizados na Seção 4, foi executado o procedimento de atualização proposto a cada semestre, levando em consideração apenas os eventos que ocorreram nos últimos sete dias (Figura 10, Janela Deslizante do Evento). Como o modelo aproveita o conhecimento prévio sobre os dados, por meio do modelo desatualizado, com a técnica de *transfer learning* (Seção 5.1), primeiro foi avaliado como os modelos desatualizados podem facilitar as atualizações do modelo. A Figura 14 mostra a convergência da proposta de acordo com a época de atualização do modelo, considerando se o modelo desatualizado é utilizado ou não no segundo semestre de 2016.

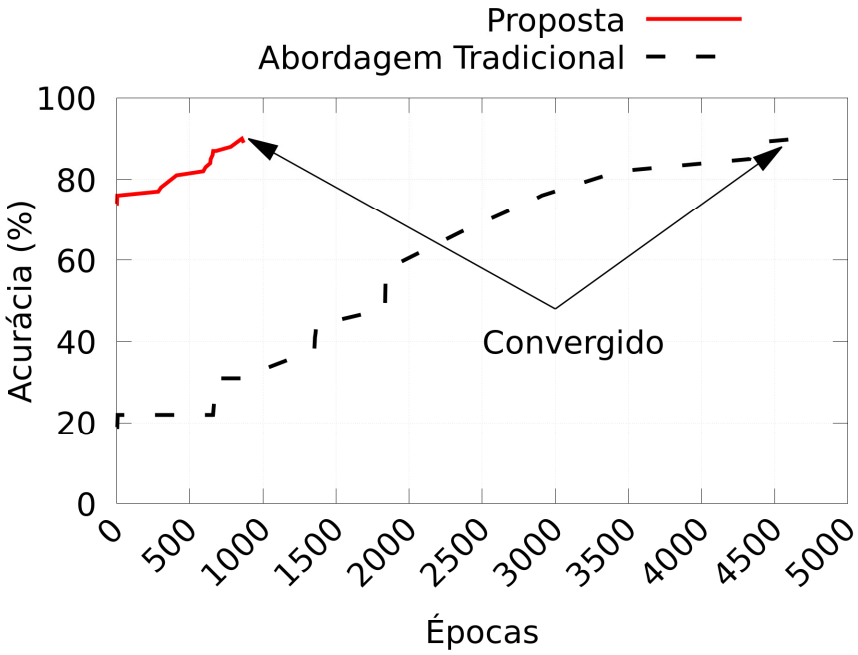


Figura 14. Convergência do treinamento da proposta, levando em consideração um agente desatualizado utilizado no processo de atualização do modelo e sua comparação com o retreinamento do zero. A precisão foi medida como a proporção do total de eventos corretamente classificados. Resultados semelhantes foram encontrados em todos os procedimentos de atualização do modelo, durante as atualizações do modelo no segundo semestre de 2016.

A abordagem proposta que aproveita o modelo desatualizado converge com significativamente menos épocas, atingindo 90% de taxa de acerto com apenas 970 épocas, mesmo com dados de apenas 1 semana. Em contraste, 4610 épocas devem ser executadas se o modelo desatualizado não for usado. Foram encontrados resultados semelhantes do período em que ocorre a atualização do modelo. A abordagem proposta que aproveita o modelo desatualizado pode diminuir significativamente as necessidades computacionais no

procedimento de treinamento, devido à diminuição do número de épocas. Também diminui o número de eventos que devem ser rotulados devido a utilização de menos procedimentos durante o treinamento. Em média, considerando todas as atualizações com longevidade do modelo de 6 meses, ao contrário de um mês usado por outras técnicas avaliadas, a abordagem proposta aproveitando o modelo desatualizado foi capaz de convergir com apenas 21,1% das épocas, contando com apenas uma semana de dados de treinamento.

A Figura 15 mostra a precisão da abordagem proposta ao longo do tempo com uma periodicidade de atualização do modelo de 6 meses. Nesse caso, as taxas de erro são significativamente mais baixas e mais estáveis com o passar do tempo. A periodicidade nas atualizações do modelo permitiu que a proposta alcançasse altas taxas de precisão ao longo dos quatro anos de dados do *MAWIFlow* [9]. A técnica proposta apresentou uma média de 14,5% e 10,0% das taxas de FP e FN ao longo do tempo, respectivamente.

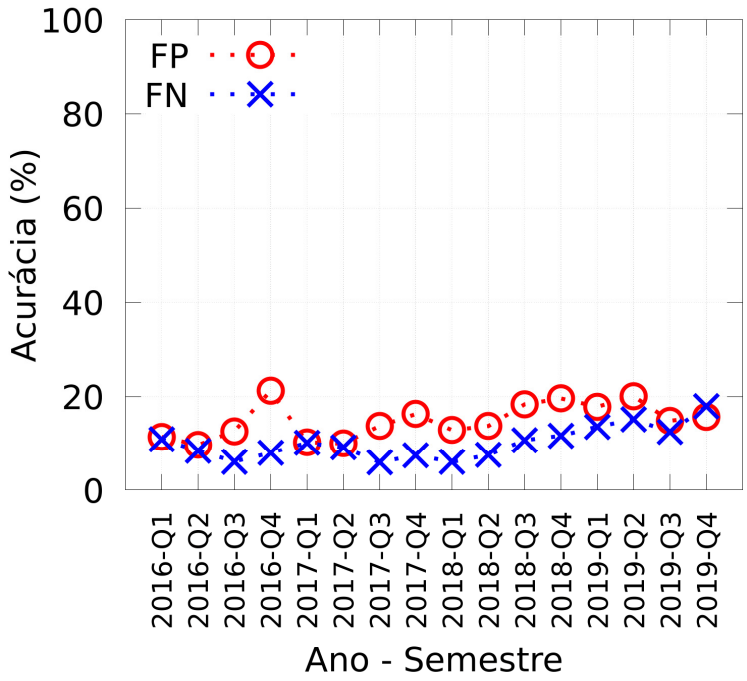


Figura 15. Desempenho de acurácia ao longo do tempo em uma base trimestral da proposta em todo o conjunto de dados MAWIFlow [9]. A proposta é treinada com dados de janeiro de 2016 e atualizada semestralmente com dados de 1 semana.

A Figura 16 mostra a variação da precisão ao longo do tempo da abordagem proposta com atualizações periódicas do modelo. A variação da precisão diminui significativamente, apresentando um intervalo interquartil médio de apenas 4,4% e 8,5% das taxas de FP e FN, enquanto as técnicas tradicionais de aprendizagem de máquina (por exemplo, *Random Forest*)

(Figura 8) apresentam uma média de 16,9% e 5,3% (ou seja, 3,84 × mais FP para apenas uma redução de 0,37 × em FN), respectivamente em todos os casos. Portanto, o modelo proposto aumenta significativamente a confiança da detecção de intrusão ao longo do tempo, estendendo a longevidade do modelo e diminuindo a variação da precisão ao longo do tempo.

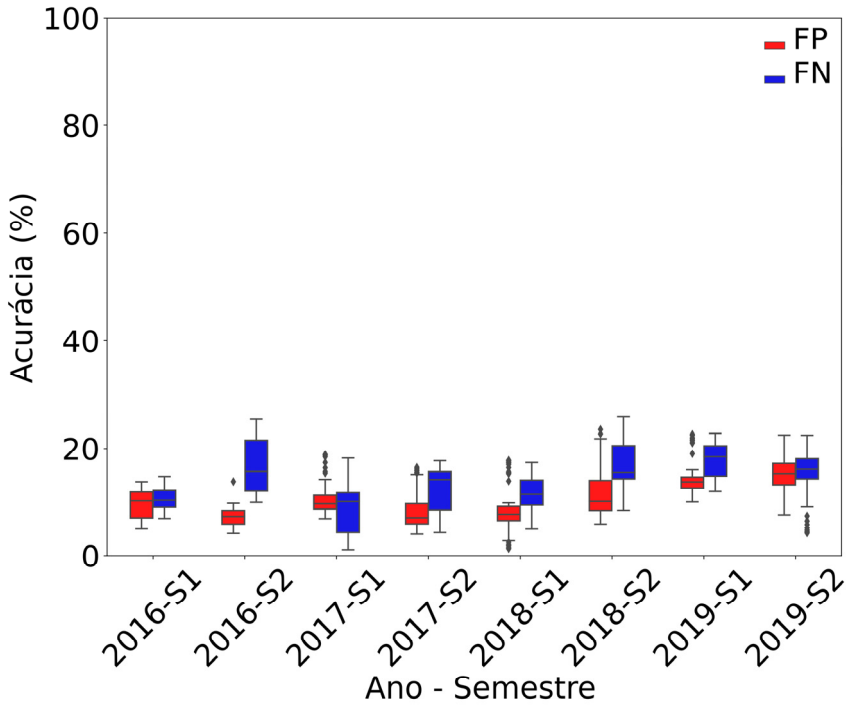


Figura 16. Distribuição semestral das acurácias diárias da proposta em todo o conjunto de dados MAWIFlow [9]. A proposta é treinada com dados de janeiro de 2016 e atualizada semestralmente com dados de 1 semana.

Também foi avaliado se a periodicidade das atualizações do modelo afeta a precisão da proposta. A Figura 17 mostra a relação entre a periodicidade de atualização do modelo e as taxas de precisão da proposta do trabalho. A proposta atinge taxas de precisão significativamente mais altas, mesmo quando a longevidade do modelo é considerada. Mais especificamente, mesmo com uma longevidade do modelo de 2 anos, atinge melhor precisão de classificação do que as técnicas tradicionais que consideram uma longevidade do modelo de 1 mês (Figura 9). Em resumo, a proposta aumenta a longevidade do modelo, a variação da classificação e diminui a periodicidade da atualização. No entanto, ela fornece taxas de precisão mais altas do que as técnicas tradicionais, mesmo quando nenhuma atualização de modelo ocorre.

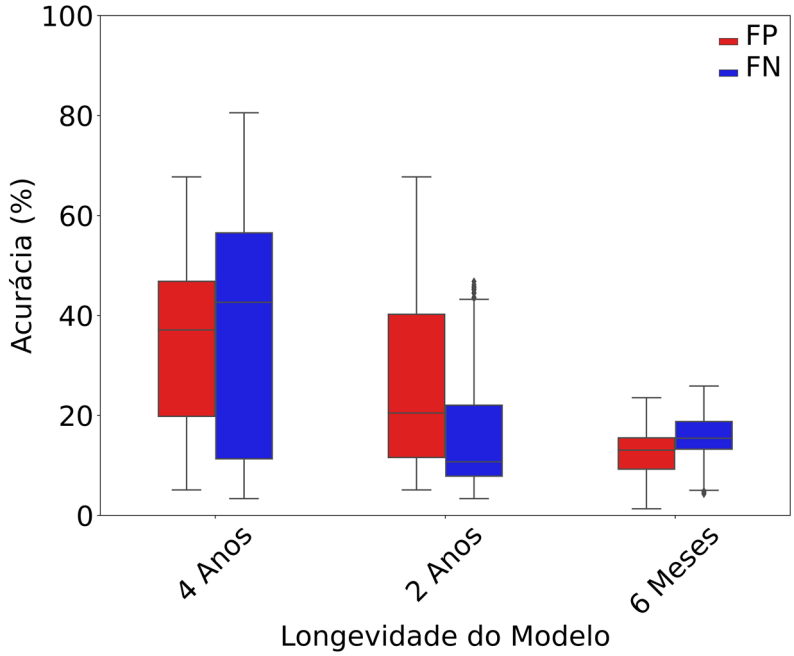
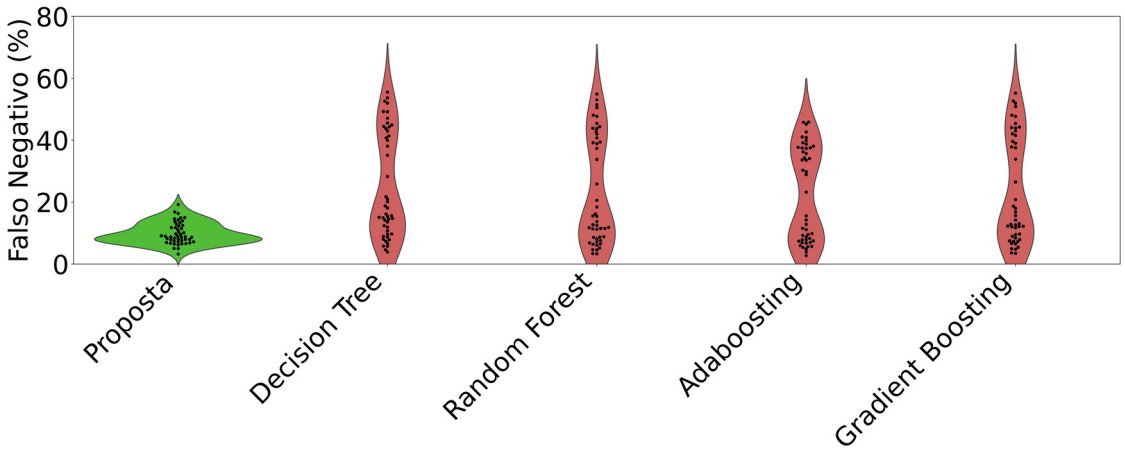


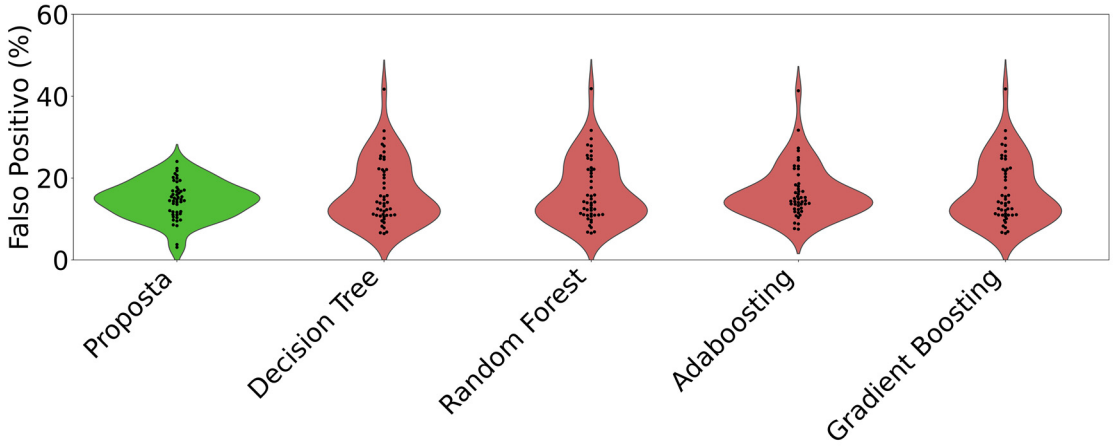
Figura 17. Distribuição da abordagem proposta ao longo do tempo e taxa de erro média todo o conjunto de dados *MAWIFlow* [9]. A longevidade do modelo é medida com a frequência da atualização, enquanto a taxa média do erro é medida com as taxas de FP e FN em todo conjunto de dados do *MAWIFlow* [9].



A) Falso Negativo

Por fim, para responder à questão RQ5, foi comparado as acurácias obtidas pelo modelo proposto e aquelas das técnicas tradicionais de aprendizagem de máquina em todo conjunto de dados *MAWIFlow* [9]. A Figura 18 mostra a distribuição mensal das taxas de precisão de cada técnica avaliada com um modelo de longevidade de 6 meses. O percentil do modelo proposto foi alcançado em 20,1% e 14,6% das taxas de FP e FN, respectivamente. Consequentemente, as técnicas tradicionais alcançaram o percentil do FP em 27,8%, 27,8%,

24,9% e 28,1% para o DT, RF, Ada e GBT, enquanto a percentil do FN foi alcançada em 49,2%, 48,1%, 41,1% e 48,1%, respectivamente. Assim, o modelo proposto fornece maior precisão de classificação na maioria das vezes (ao longo dos meses do conjunto de dados), melhorando a taxa média de FP em até 8,0% e a taxa média de FN em até 34,6% quando comparada às técnicas tradicionais.



B) Falso Negativo

C)

Figura 18. Distribuição da precisão diária da abordagem proposta em todo o conjunto de dados *MAWIFlow* [9]. A proposta é treinada com dados de janeiro de 2016 e atualizada semestralmente com dados de 1 semana.

6.3 Discussão

A técnica proposta foi avaliada utilizando quatro anos do conjunto de dados do conjunto MAWIFlow [9]. Para avaliação, o modelo criado utilizou o mês de janeiro de 2016 (30 dias) para sua criação.

A primeira avaliação mostrou que, mesmo sem atualizações, a proposta é capaz de manter a longevidade e a confiança da classificação do modelo por períodos mais extensos. E, quando comparado com as abordagens tradicionais, se mostra significativamente melhor. Além disto a abordagem mostra que as taxas de acurácia foram semelhantes as taxas das abordagens tradicionais. Sendo assim, o modelo foi capaz de fornecer uma alta precisão de intrusão, mesmo

sem nenhuma atualização. Ainda, quanto a variação do modelo, foi apresentado que é significativamente menor que os algoritmos tradicionais da literatura, reduzindo a variação da precisão do modelo ao longo do tempo.

A aplicação de atualizações periódicas na proposta, demonstra que mesmo utilizando apenas um conjunto com 7 dias de dados a cada 6 meses, foi capaz de atualizar o modelo com eficácia, convergindo muito mais rápido durante a criação. A avaliação do modelo com atualização, mostra que a precisão melhorou significativamente, com taxas mais baixas e estáveis de FP e FN. Além disso a variação do diminuiu significativamente comparado com as técnicas tradicionais, ou seja, o modelo estendeu a longevidade e diminuiu variação da precisão ao longo do tempo de maneira significativa.

Por fim, foi avaliado a comparação entre as acurácias das técnicas tradicionais, com as acurácias da proposta em todo o conjunto de dados MAWIFlow [9]. E pode-se observar que o modelo proposto fornece maior precisão na classificação na maioria dos meses, comparado com as técnicas tradicionais.

Capítulo 7

Conclusão

Novas abordagens para detecção de intrusão por meio de técnicas de aprendizado de máquina foram amplamente propostas na literatura científica nos últimos anos, enquanto apenas algumas foram implantadas em produção. Foi mostrado no trabalho que os pesquisadores adotam incorretamente as suposições de aprendizado de máquina tradicionais no domínio de detecção de intrusão (por exemplo, o tráfego de rede era de natureza estática, o que significa que não muda com o tempo). Contudo, o tráfego de rede sofre com a mudança de comportamento diariamente, isto ocorre devido novos conjuntos de ataques de rede que surgem e novos serviços disponíveis.

Este trabalho propôs a pesquisa dessas suposições de detecção de intrusão em relação ao comportamento do tráfego de rede. Logo, o trabalho motivou a pesquisa relacionada à longevidade, à precisão e à variabilidade dos modelos de detecção de intrusão em trabalhos encontrados na literatura. Mostrando que a grande maioria não se preocupa em avaliar estes três problemas encontrados na detecção de intrusão em ambientes de produção. Para avaliar, buscou-se artigos na literatura que tentam tratar as mudanças de comportamento, a atualização do modelo, a variabilidade da acurácia ao longo do tempo, modelos de detecção de intrusão desatualizados e que utilizam conjuntos de dados de ambientes de produção.

Entretanto, a grande maioria dos trabalhos emprega técnicas que permitem atingir uma alta taxa de acurácia em um conjunto de dados, enquanto outros trabalhos utilizam técnicas para escolher os melhores conjuntos de características de fluxos de rede para criar os modelos mais rápidos e com melhor eficácia. Também existem trabalhos que utilizam técnicas promissoras em conjuntos de dados muito antigos, que não representam a variabilidade do tráfego de rede

atual, tornando as técnicas mal avaliadas e sem a utilização em ambientes de produção. Poucos trabalhos na literatura se preocupam com a atualização do modelo, a grande maioria acredita que os ambientes de detecção de intrusão sempre iram possuir rótulo do evento disponível para atualização. Todavia, em um ambiente produção, o rótulo não existe, normalmente é necessário auxílio humano, tornando muito custoso a atualização de um ambiente que gera milhões ou bilhões de eventos de rede diariamente. Poucos trabalhos da literatura avaliam o tráfego de rede ao longo do tempo, normalmente quando avaliam, são períodos muito pequenos que não consegue aferir corretamente a técnica ao longo do tempo, impossibilitando a utilizando em ambientes de produção. Outro problema discutido no trabalho é o fato de que não é simples saber que um modelo de detecção de intrusão está desatualizado, é necessário que um especialista verifique. Todavia, enquanto um novo modelo está sendo criado, a organização fica impossibilitada de utilizar o modelo atual e descobrir ataques de rede que podem surgir a qualquer momento.

Este trabalho avaliou os problemas de longevidade do modelo, sua precisão e variabilidade na classificação dos eventos de rede com algoritmos tradicionais da literatura *Decision Tree* (DT), *Random Forest* (RF), *Adaboosting* (Ada) e *Gradient Boosting* (GBT) no conjunto de dados MAWIFlow [9] (ver seção 2.3) que contempla quatro anos de tráfego de rede (2016 – 2019). Para tal avaliação, o trabalho buscou responder duas questões de pesquisa. (RQ1) Qual é o comportamento das abordagens baseadas em aprendizagem de máquina amplamente utilizadas em termos de precisão ao longo do tempo quando nenhuma atualização do modelo é realizada? (RQ2) Qual é o impacto das atualizações periódicas do modelo na precisão das abordagens baseadas em aprendizagem de máquina amplamente utilizadas?

A questão RQ1 utilizou o mês de janeiro de 2016 para treinamento dos modelos, foi testada ao longo do tempo em todo o conjunto de dados do MAWIFlow [9]. Os testes mostraram que em média o intervalo interquartil médio dos algoritmos aumenta em 3,4% e 4,0% para as taxas de FP e FN a cada seis meses de longevidade do modelo, aumentando significativamente a variação da precisão das classificações ao longo do tempo. Ou seja, as técnicas tradicionais não conseguem lidar com o comportamento do tráfego de rede ao longo do tempo quando os modelos não são atualizados periodicamente. Além disso, a avaliação das acurácias sem atualizações pode mostrar que as técnicas logo nos meses iniciais já se tornam ineficazes para utilização em ambientes de produção, caso não sejam atualizadas.

A questão RQ2 considerou a longevidade do modelo de 6 meses, ou seja, atualizando os modelos tradicionais semestralmente. Neste caso, pode-se observar que as acurácias e a variabilidade das classificações melhoram significativamente, diminuindo a variação da precisão ao longo do tempo.

Para resolver os problemas dos algoritmos os problemas de longevidade do modelo, precisão do modelo e variabilidade na classificação dos eventos de rede dos algoritmos tradicionais, este trabalho apresentou uma nova abordagem de detecção de intrusão de rede utilizando a técnica de aprendizagem por reforço, aplicando o algoritmo de *Q-Learning*. A proposta do trabalho respondeu 3 questões de pesquisa, sendo elas: (RQ3) Como o modelo proposto se comporta sem as atualizações do modelo ao longo do tempo? (RQ4) Qual é o impacto das atualizações do modelo na proposta? (RQ5) O modelo proposto pode fornecer maior confiança e menor variabilidade do que as técnicas tradicionais?

A questão RQ3 mostrou que a abordagem proposta foi capaz de manter a longevidade e a confiança do modelo por períodos mais extensos, mantendo as taxas de FP mais próximas da fase de treinamento do classificador ao longo do tempo. Em média, há um aumento de 4,2% e 0,3% nas taxas de FP e FN respectivamente. Além disso as taxas de acurácia ficaram semelhantes às dos algoritmos tradicionais. Desta forma, o modelo mostrou que é capaz de fornecer uma alta taxa de precisão na detecção de intrusão, mesmo sem utilizar nenhuma atualização do modelo. Ainda, quanto à variação da classificação ao longo do tempo, ela é significativamente menor que as variações das abordagens tradicionais, consequentemente a abordagem é capaz de aumentar a longevidade do modelo.

A questão RQ4 utilizou a atualização periódica do modelo semestralmente, contudo foi comprovado que o modelo é capaz convergir durante o treinamento muito mais rapidamente quando aproveita o conhecimento de um modelo desatualizado, também consegue atualizar utilizando apenas os dados de 1 semana. Desta forma, a proposta utiliza muito menos recursos computacionais ou assistência humana para rotular os dados.

Além disso, foi comparado as taxas de acurácia e precisão ao longo do tempo com as das técnicas tradicionais. Foi comprovado que o modelo proposto aumenta significativamente a confiança da detecção de intrusão ao longo do tempo, estendendo a longevidade do modelo e diminuindo a variação da precisando ao longo do tempo.

A questão RQ5 utilizando o modelo atualizado semestralmente, comparou os resultados com as abordagens tradicionais da literatura em todo conjunto de dados. A proposta do trabalho

é capaz de fornecer na maior precisão na classificação, na maioria das vezes comparada as técnicas tradicionais de literatura.

O conjunto de dados utilizado em todos os experimentos deste trabalho está disponível publicamente para download em https://secplab. ppgia.pucpr.br/reinforcemawiflow.

7.1 Trabalhos Futuros

Este trabalho apresenta os seguintes trabalhos futuros:

- Avaliar outros algoritmos de aprendizagem de máquina da literatura para gerar diferentes políticas de aprendizagem por reforço e estabelecer diferentes diretrizes de longevidade aos modelos de aprendizagem por reforço, visto que a técnica apresentada no trabalho, não depende especificamente de uma rede neural, mas sim de qualquer algoritmo capaz de gerar probabilidades de ações em um ambiente simulado de rede. Desta forma é possível aplicar outras políticas e avaliar o desempenho e a longevidade do modelo;
- Avaliar outras métricas de confiança para criação das ações do agente no ambiente simulado, permitindo menor variabilidade nos resultados das classificações. Durante o processo de criação dos modelos, foi utilizado a métrica de distância da classe real, seria interessante encontrar outras métricas que permitam ser utilizadas durante o treinamento do agente no ambiente simulado, possibilitando maior exatidão nos modelos de aprendizagem por reforço;
- Desenvolver um novo método para avaliar a vida útil dos modelos, a fim de estabelecer se um modelo deve ser reconstruído ou não, é muito complicado identificar em um ambiente de produção quando um modelo se torna ineficaz, logo é importante uma técnica que seja capaz de avaliar quando o modelo se tornou ineficaz e deve ser criado um novo modelo para detecção de intrusão.

Referências

[1] Kaspersky Lab., "Kaspersky security bulletin 2019. statistics," 2019. [Online]. Disponível em: https://securelist.com/kaspersky-securitybulletin-2019-statistics/95475/bbb

[2] Cisco, "Cisco internet report (2018–2023)," 2020. [Online]. Disponível em: https://www.cisco.com/c/en/us/solutions/collateral/executiveperspectives/annual-internet-report/white-paper-c11-741490.html

[3] B. Molina-Coronado, U. Mori, A. Mendiburu, e J. Miguel-Alonso, "Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process," IEEE Trans. on Network and Service Management, vol. 17, no. 4, pp. 2451–2479, Dez. 2020.

[4] N. Hubballi e V. Suryanarayanan, "False alarm minimization techniques in signature-based intrusion detection systems: A survey," Computer Communications, vol. 49, pp. 1–17, Ago. 2014. [Online]. Disponível em: https://doi.org/10.1016/j.comcom.2014.04.012

[5] R. Sommer e V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in 2010 IEEE Symposium on Security and Privacy. IEEE, 2010. [Online]. Disponível em: https://doi.org/10.1109/sp.2010.25

[6] C. Gates e C. Taylor, "Challenging the anomaly detection paradigm: A provocative discussion," in Proc. of the Workshop on New Security Paradigms (NSPW), pp. 21–29, 2006.

[7] M. Lopez-Martin, B. Carro, e A. Sanchez-Esguevillas, "Application of deep reinforcement learning to intrusion detection for supervised problems," Expert Systems with Applications, vol. 141, p. 112963, Mar. 2020. [Online]. Disponível em: https://doi.org/10.1016/j.eswa.2019.112963

[8] M. Injadat, A. Moubayed, A. B. Nassif, e A. Shami, "Multi-stage optimized machine learning framework for network intrusion detection," IEEE Transactions on Network and Service Management, pp. 1–1, 2020. [Online]. Disponível em: https://doi.org/10.1109/tnsm.2020.3014929

[9] E. Viegas, A. Santin, A. Bessani, e N. Neves, "Bigflow: Real-time and reliable anomaly-based intrusion detection for high-speed networks," Future Generation Computer Systems, vol. 93, pp. 473–485, 2019.

[10] G. Folino, F. S. Pisani, e L. Pontieri, "A GP-based ensemble classification framework for time-changing streams of intrusion detection data," Soft Computing, vol. 24, no. 23, pp. 17541–17560, Ago. 2020.

[11] M. Tavallaee, N. Stakhanova, e A. A. Ghorbani, "Toward credible evaluation of anomaly-based intrusion-detection methods," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 40, no. 5, pp. 516–524, Set. 2010. [Online]. Disponível em: https://doi.org/10.1109/tsmcc.2010.2048428

[12] F. Maggi, W. Robertson, C. Kruegel, e G. Vigna, "Protecting a moving target: Addressing web application concept drift," in Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 21–40, 2009.

[13] F. Pinage, E. M. dos Santos, e J. Gama, “A drift detection method´ based on dynamic classifier selection,” Data Mining and Knowledge Discovery, vol. 34, no. 1, pp. 50–74, Oct. 2019. [Online]. Disponível em: https://doi.org/10.1007/s10618-019-00656-w

[14] E. Kim, J. Lee, H. Shin, H. Yang, S. Cho, S. kwan Nam, Y. Song, J. a Yoon, e J. il Kim, “Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning,” Expert Systems with Applications, vol. 128, pp. 214–224, Ago. 2019. [Online]. Disponível em: https://doi.org/10.1016/j.eswa.2019.03.042

[15] V. Dremin, Z. Marcinkevics, E. Zherebtsov, A. Popov, A. Grabovskis, H. Kronberga, K. Geldnere, A. Doronin, I. Meglinski, e A. Bykov, “Skin complications of diabetes mellitus revealed by polarized hyperspectral imaging and machine learning,” IEEE Transactions on Medical Imaging, pp. 1–1, 2021. [Online]. Disponível em: https://doi.org/10.1109/tmi.2021.3049591

[16] J. Memon, M. Sami, R. A. Khan, e M. Uddin, “Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR),” IEEE Access, vol. 8, pp. 142642–142668, 2020. [Online]. Disponível em: https://doi.org/10.1109/access.2020.3012542

[17] S. Wassermann, T. Cuvelier, P. Mulinka, e P. Casas, “Adaptive and reinforcement learning approaches for online network monitoring and analysis,” IEEE Trans. on Net. and Service Management, pp. 1–1, 2020.

[18] A. Géron "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow". O'Reilly, edição 2, 2019.

[19] M. Tavallaee, E. Bagheri, W. Lu, e A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 dataset,” in 2009 IEEE Symposium on Computational Intelligence for Security e Defense Applications. IEEE, Jul. 2009. [Online]. Disponível em: https: //doi.org/10.1109/cisda.2009.5356528

[20] MAWI, “MAWI Working Group Traffic Archive - Samplepoint F,” 2021. [Online]. Disponível em: https://mawi.wide.ad.jp/mawi/

[21] R. Fontugne, P. Borgnat, P. Abry, e K. Fukuda, “MAWILab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking,” in Proc. of the 6th Int. Conf. on emerging Networking Experiments and Technologies (CoNEXT), 2010.

[22] L. Koc, T. A. Mazzuchi, and S. Sarkani, “A network intrusion detection system based on a hidden naıve bayes multiclass classifier,” in Expert Systems with Applications, vol. 39, 2012, pp. 13492 – 13500.

[23] S. Longari, D. H. N. Valcarcel, M. Zago, M. Carminati, e S. Zanero, “Cannolo: An anomaly detection system based on lstm autoencoders for controller area network,” IEEE Transactions on Network and Service Management, pp. 1–1, 2020.

[24] S. Otoum, B. Kantarci, e H. Mouftah, “Empowering reinforcement learning on big, sensed data for intrusion detection,” in IEEE International Conference on Communications (ICC), 2019, pp. 1–7.

[25] Akashdeep, I. Manzoor, e N. Kumar, “A feature reduced intrusion detection system using ann classifier,” Expert Systems with Applications, vol. 88, pp. 249–257, 2017.

[26] T. Hamed, R. Dara, e S. C. Kremer, “Network intrusion detection system based on recursive feature addition and bigram technique,” in Computers & Security, vol. 73, pp. 137 – 155, 2018.

[27] K. Yang, J. Liu, C. Zhang e Y. Fang, "Adversarial Examples Against the Deep Learning Based Network Intrusion Detection Systems," MILCOM 2018 - 2018 IEEE Military Communications Conference

(MILCOM), Los Angeles, CA, pp. 559-564, 2018.

[28] Nguyen Thanh Van, Tran Ngoc Thinh e Le Thanh Sach, "An anomaly-based network intrusion detection system using Deep learning," 2017 International Conference on System Science and Engineering (ICSSE), Ho Chi Minh City, pp. 210-214, 2017.

[29] Farnaaz, N., Jabbar, M. A. Random Forest Modeling for Network Intrusion Detection System.Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016), 2016.

[30] Sukumar, J. V. A., Pranav, I., Neetish, MM, Narayanan, J. Network Intrusion Detection Using Improved Genetic k-means Algorithm. 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018.

[31] Van, N., Thinh, T., e Sach, L. An anomaly-based network intrusion detection system using deep learning. In 2017 International Conference on System Science and Engineering (ICSSE), 2017, pp 210–214.

[32] Gupta, D., Singhal, S., Malik, S., e Singh, A. Network intrusion detection system using various data mining techniques. InInternational Conference on Research Advances in Integrated Navigation Systems (RAINS - 2016), 2016.

[33] Elisa, N., Yang, L., Fu, X, Naik, N. Dendritic Cell Algorithm Enhancement Using Fuzzy Inference System for Network Intrusion Detection. 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2019.

[34] E. Seo, H. M. Song e H. K. Kim, "GIDS: GAN based Intrusion Detection System for In-Vehicle Network," 2018 16th Annual Conference on Privacy, Security and Trust (PST), pp. 1-6, 2018, doi: 10.1109/PST.2018.8514157.

[35] V. Hajisalem, S. Babaie, A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection, Computer Networks, Volume 136, pp 37-50, 2018, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2018.02.028.

[36] F. Farahnakian e J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system," 2018 20th International Conference on Advanced Communication Technology (ICACT), pp. 178-183, 2018, doi: 10.23919/ICACT.2018.8323688.

[37] W. Choi, K. Joo, H. J. Jo, M. C. Park e D. H. Lee, "Voltage IDS: Low-Level Communication Characteristics for Automotive Intrusion Detection System," in IEEE Transactions on Information Forensics and Security, vol. 13, no. 8, pp. 2114-2129, Ago. 2018, doi: 10.1109/TIFS.2018.2812149.

[38] S. U. Jan, S. Ahmed, V. Shakhov e I. Koo, "Toward a Lightweight Intrusion Detection System for the Internet of Things," in IEEE Access, vol. 7, pp. 42450-42471, 2019, doi: 10.1109/ACCESS.2019.2907965.

[39] H. Alazzam, A. Sharieh, K. E. Sabri, A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer In: Expert Systems with Applications, Volume 148, 2020, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.113249.

[40] E. K. Viegas, A. O. Santin, V. V. Cogo, e V. Abreu, “A reliable semi-supervised intrusion detection model: One year of network traffic anomalies,” in ICC 2020 - 2020 IEEE International Conference on Communications (ICC). IEEE, jun. 2020.

[41] G. Caminero, M. Lopez-Martin, e B. Carro, “Adversarial environment reinforcement learning algorithm for intrusion detection,” Computer Networks, vol. 159, pp. 96–109, 2019.

[42] D. Upadhyay, J. Manero, M. Zaman, e S. Sampalli, “Gradient boosting feature selection with machine learning classifiers for intrusion detection on power grids,” IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 1104–1116, 2021.

[43] J. Zhang, F. Li, H. Wu, e F. Ye, “Autonomous model update schemefor deep learning-based network traffic classifiers,” in 2019 IEEE Global Communications Conference (GLOBECOM). IEEE, Dez. 2019.

[44] J. Liang e M. Ma, “Co-maintained database based on blockchain for idss: A lifetime learning framework,” IEEE Transactions on Network and Service Management, pp. 1–1, 2021.

[45] Sethi, K., Sai Rupesh, E., Kumar, R. et al. A context-aware robust intrusion detection system: a reinforcement learning-based approach. Int. J. Inf. Secur. 19, pp 657–678, 2020.

[46] M. Mazini, B. Shirazi e I. Mahdavi, “Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms” In: Journal of King Saud University - Computer and Information Sciences, Volume 31, Issue 4, pp. 541-553, 2019.

[47] Nguyen Thanh Van, Tran Ngoc Thinh e Le Thanh Sach, "An anomaly-based network intrusion detection system using Deep learning," 2017 International Conference on System Science and Engineering (ICSSE), Ho Chi Minh City, pp. 210-214, 2017.

[48] C. J. C. H. Watkins e P. Dayan, “Q-learning,” Machine Learning, vol. 8, no. 3-4, pp. 279–292, May 1992. [Online]. Disponível em: https://doi.org/10.1007/bf00992698

[49] OpenAI, “OpenAI - Gym,” 2021. [Online]. Disponível em: https://gym. openai.com/

[50] TensorFlow, “TensorFlow API,” 2021. [Online]. Disponível em: https://www.tensorflow.org/