Sistema de Detecção de Intrusão Confiável Baseado em Aprendizagem por Fluxo

Eduardo K. Viegas[1], **Altair O. Santin**[1], **Roger R. dos Santos**[1], **Vilmar Abreu**[1]

[1]Programa de Pós-Graduação em Informática (PPGIa)
Pontifícia Universidade Católica do Paraná (PUCPR)
80.215-901 - Curitiba - PR

{eduardo.viegas, santin, robson.roger, vilmar.abreu}@ppgia.pucpr.br

Abstract. *Intrusion detection systems through machine learning techniques have been extensively used in the literature. However, although the promising reported results, due to the lack of reliability in the accuracy of the system, such techniques are hardly used in production. In this paper, we propose a reliable intrusion detection model through stream learning algorithms. The system reliability is provided through the classification confidence assessment. Experiments have shown the proposal feasibility, which maintained its accuracy while classifying new attacks and services, autonomously updating its system.*

Resumo. *Sistemas de detecção de intrusão baseados em aprendizagem de máquina são amplamente propostos na literatura. Porém, apesar dos resultados promissores reportados, devido a falta de confiabilidade na acurácia dos sistemas, tais técnicas raramente são utilizadas em produção. Neste artigo, propomos um sistema de detecção de intrusão confiável baseado em algoritmos de aprendizagem por fluxo. A confiabilidade do sistema é provida através da avaliação da confiança da classificação efetuada pelo sistema. Experimentos demonstraram a viabilidade de proposta, que manteve a sua acurácia mesmo durante a classificação de novos ataques e serviços, atualizando o sistema de modo autônomo.*

1. Introdução

Atualmente, as abordagens baseadas em aprendizagem de máquina têm sido amplamente utilizadas para o desenvolvimento de sistemas de detecção de intrusão em rede (*Network-based Intrusion Detection System*, NIDS) [Sommer e Paxson 2010]. Porém, apesar dos resultados promissores reportados na literatura, os NIDS baseados em aprendizagem de máquina raramente são utilizados em ambientes de produção [Viegas et al. 2017b]. Isso ocorre porque as acurácias reportadas na literatura não são evidenciadas quando os sistemas propostos são usados em produção. Consequentemente, as técnicas não se tornam confiáveis para sua devida utilização, tornando-se apenas um tópico de pesquisa, raramente utilizado em produção.

Em NIDS, aprendizagem de máquina é geralmente efetuada através de técnicas de reconhecimento de padrões [Gates e Taylor 2007]. Nesse caso, um modelo é construído por meio de uma base de treinamento, em geral, composta por fluxos de rede. Os dados de treinamento compreendem as atividades normais e maliciosas do ambiente, que são esperadas do ambiente de produção. Como resultado, o modelo obtido é capaz de classificar

novos eventos, desde que apresentem um comportamento semelhante ao observado na base de treinamento [Viegas et al. 2017b]. Por outro lado, o tráfego de rede é altamente variável, portanto, a construção de uma base de treinamento real não é uma tarefa facilmente alcançada [Peng et al. 2016]. Além disso, mesmo que a base de treinamento seja construída de modo adequado, com o passar do tempo, novos ataques serão descobertos e novos serviços serão fornecidos. Portanto, o modelo de aprendizagem de máquina construído se tornará ineficaz, ou seja, obsoleto, pois o comportamento evidenciado na base de treinamento não será mais observado no ambiente de produção [Sommer e Paxson 2010].

Na literatura, devido a descoberta de novos ataques de rede, os trabalhos geralmente assumem que atualizações periódicas do modelo são efetuadas [Peng et al. 2016]. No entanto, a atualização do modelo não é facilmente alcançada, pois uma nova base de treinamento deve ser construída para cada atualização do modelo. Essa tarefa requer a coleta de novos dados de rede, a rotulagem prévia de maneira adequada (ou seja, a classificação de um evento como normal ou ataque) e a execução de um processo computacional custoso para o treinamento do modelo. Como resultado, a tarefa de retreinamento do modelo pode exigir vários dias ou até mesmo semanas antes que um modelo atualizado esteja disponível [Sommer e Paxson 2010]. Esse atraso na atualização do modelo deixa o sistema desprotegido e acarreta no aumento das taxas de erro do sistema. Portanto, um NIDS baseado em aprendizagem de máquina deve ser capaz de executar classificações confiáveis, mesmo na presença de um comportamento de rede desconhecido [Gates e Taylor 2007].

Porém, na literatura, os NIDS baseados em aprendizagem de máquina normalmente buscam apenas melhorar sua acurácia em um determinado conjunto de dados [Tavallaee et al. 2010]. Consequentemente, as técnicas propostas não são confiáveis para ambientes de produção, por diminuírem sua acurácia ao longo do tempo, exigindo o retreino periódico do modelo. Em geral, na literatura, para garantir a confiabilidade das classificações do modelo de aprendizagem de máquina, os autores geralmente recorrem a técnicas de rejeição [Loganathan et al. 2018]. Abordagens baseadas em rejeição avaliam a confiança da classificação de um evento realizado por um modelo antes que uma decisão possa ser aceita ou não pelo sistema. No entanto, em ambientes de rede, devido às mudanças de comportamento ao longo do tempo, a métrica de confiança dos classificadores é ineficaz para ser usada com segurança em ambientes de produção [Peng et al. 2016]. Essa limitação ocorre porque os valores de confiança são calculados de acordo com o comportamento evidenciado na base de treinamento, que devido as mudanças de comportamento do ambiente ao longo do tempo, não representa o comportamento atual do ambiente de produção [Yin et al. 2017]. Além disso, mesmo que a confiabilidade das classificações possa ser garantida, o retreino periódico do modelo ainda é necessário, devido as constantes mudanças do comportamento nos ambientes de rede [Sommer e Paxson 2010] [Viegas et al. 2017b] [Gates e Taylor 2007].

Neste contexto, este trabalho propõe um NIDS confiável baseado em técnicas de aprendizagem de fluxo (*stream learning*), visando a confiabilidade das classificações mesmo na ocorrência de tráfego de rede desconhecido ao modelo ao longo do tempo, assim como a facilidade nas atualizações do modelo de aprendizagem de máquina. Para atingir esse objetivo, nossa proposta é efetuada em duas etapas. Primeiramente, nossa proposta trata o desafio de prover classificações confiáveis mesmo na ocorrência de tráfego

de rede desconhecido ao modelo através do uso de algoritmos de aprendizagem de fluxo baseado em anomalia, assim, garantindo que apenas classificações conhecidas ao modelo, e portanto, confiáveis, são aceitas pelo sistema. Posteriormente, nossa proposta realiza a detecção confiável de intrusões por meio de algoritmos de aprendizagem de fluxo, aliado a abordagem de detecção de anomalia proposta, permitindo assim a fácil atualização do sistema através da atualização incremental dos modelos utilizados. Como resultado, nossa proposta é capaz de garantir a confiabilidade das classificações, mesmo quando o tráfego de rede é desconhecido ao modelo, ou seja, não presente na base de treinamento, atualizando de forma incremental o modelo de aprendizagem de fluxo existente. Portanto, incorporando de maneira confiável o novo comportamento em nosso sistema.

Em resumo, este artigo apresenta uma técnica de aprendizagem de fluxo baseado em anomalia que garante a confiabilidade das classificações ao longo do tempo mesmo sem a atualização dos modelos. A abordagem proposta é capaz de avaliar as classificações realizadas pelo modelo existente de aprendizagem de máquina e incorporar um novo comportamento de tráfego de rede de modo incremental. Adicionalmente, este artigo apresenta um modelo confiável de detecção de intrusão baseado em técnicas de aprendizagem por fluxo capaz de incorporar de maneira confiável novos comportamentos de tráfego de rede, facilitando a tarefa de atualização do modelo. A abordagem proposta é capaz de efetuar classificações confiáveis, mesmo para comportamento de tráfego desconhecido, enquanto também facilita a inclusão de novos comportamentos para detecção de novos ataques ou serviços no tráfego de rede ao longo do tempo.

O restante deste artigo esta organizado da seguinte maneira. A seção 2 apresenta o estado da arte em NIDS e aprendizagem por fluxo. A seção 3 aborda os trabalhos relacionados. A seção 4 trata dos detalhes da proposta, enquanto a seção 5 avalia o método proposto. Por fim, a seção 6 conclui o trabalho.

2. Estado da Arte

2.1. *Network-based Intrusion Detection*

Um NIDS objetiva encontrar atividades maliciosas em um ambiente de rede [Gates e Taylor 2007]. Nos últimos anos, várias abordagens de detecção de intrusão foram propostas para efetuar essa tarefa, nas quais as técnicas de aprendizagem de máquina reportaram resultados promissores [Sommer e Paxson 2010]. Para atingir esse objetivo, um NIDS baseado em aprendizagem é tipicamente composto por quatro módulos sequenciais, sendo *Aquisição de Dados*, *Extração de Características*, *Classificação* e *Alerta* [Kugler et al. 2020]. Primeiro, o módulo de *Aquisição de dados* reúne os dados da rede do ambiente, por exemplo, os pacotes de rede. Então, o módulo de *Extração de Características* extrai um conjunto de características para compor uma descrição comportamental do evento, ou seja, vetor de características. Em geral, em NIDS, o comportamento do evento é representado como um fluxo de rede, que compreende os dados trocados entre as entidades da rede em um determinado período de tempo. Através do vetor de características extraído, o módulo de *Classificação* aplica um modelo de aprendizagem de máquina para classificar a entrada como normal ou maliciosa. Por fim, se um evento malicioso for encontrado, o módulo *Alerta* o reporta [Abreu et al. 2017].

Várias abordagens baseadas em aprendizagem de máquina foram propostas para a tarefa de classificação, nas quais as técnicas de reconhecimento de padrões são geral-

mente utilizadas produzindo resultados promissores [Viegas et al. 2017b]. Em reconhecimento de padrões, um modelo de aprendizagem de máquina é construído de acordo com o comportamento extraído de um conjunto de dados de treinamento. Consequentemente, o conjunto de dados de treinamento deve incluir o comportamento esperado do ambiente de produção. No entanto, a construção de um conjunto de dados de treinamento adequado no NIDS não é facilmente alcançada. Isso ocorre porque o comportamento do tráfego de rede é altamente variável e tem muitas mudanças com o tempo [Peng et al. 2016]. Como resultado, mesmo que um conjunto de dados de treinamento "*perfeito*" seja construído, ele falharia ao considerar o tráfego de rede como estático [Viegas et al. 2019]. Portanto, as técnicas de reconhecimento de padrões para o NIDS exigem atualizações periódicas do modelo, o que também não é facilmente viável em ambientes de produção.

2.2. Aprendizagem de fluxo para detecção de intrusão

Nos últimos anos, várias técnicas de aprendizagem de fluxo (*stream learning*) foram propostas para cenários em que o comportamento muda com o tempo [He et al. 2011]. Uma abordagem de aprendizagem de fluxo se contrasta das técnicas tradicionais de reconhecimento de padrões, permitindo que o modelo existente seja atualizado gradualmente ao longo do tempo, de modo incremental, reaproveitando o modelo atual. Como resultado, a tarefa de atualização do modelo é significativamente aprimorada, pois o modelo atual não é descartado [Peng et al. 2016]. Ou seja, novos comportamentos podem ser incorporados ao modelo de forma incremental, diminuindo significativamente os requisitos computacionais e de tempo demandados para a atualização do sistema [dos Santos et al. 2020].

No entanto, as técnicas tradicionais de aprendizagem de fluxo assumem um cenário supervisionado, no qual o rótulo do evento sempre encontra-se disponível [He et al. 2011]. Em outras palavras, exige que o rótulo do evento atual seja conhecido para incorporar o novo comportamento no modelo existente [Sovilj et al. 2020]. Por outro lado, em NIDS, o rótulo apropriado do evento nem sempre está disponível, uma vez que geralmente requer assistência especializada humana. Como consequência, apenas um pequeno subconjunto de eventos devidamente rotulados pode ser fornecido ao longo do tempo [Sommer e Paxson 2010], tornando assim a aplicabilidade das técnicas de aprendizagem de fluxo em NIDS de difícil uso.

3. Trabalhos Relacionados

Nos últimos anos, várias abordagens baseadas em aprendizagem de máquina foram propostas para NIDS [Sommer e Paxson 2010]. Em geral, aprendizagem de máquina em NIDS é realizada através de técnicas de reconhecimento de padrões [Gates e Taylor 2007]. Por exemplo, [Tobi e Duncan 2019] propõe o uso de vários classificadores, cada um com um limite de classe adaptável para lidar com o tráfego de rede. Os autores mostraram que, devido à variabilidade do tráfego de rede, os modelos de aprendizagem de máquina utilizados devem ser otimizados de acordo. No entanto, os autores não abordaram a tarefa de atualização do modelo, nem a confiabilidade das classificações em face a novos tráfegos. Por outro lado, [P.Singh e Venkatesan 2018] constrói um classificador de *random forest* aliado a um algoritmo de agrupamento, o *k-means*, para identificar novos ataques. Em seu trabalho, os autores aplicam a abordagem de agrupamento para identificar novos comportamentos, enquanto treinam o classificador para detectá-los. Porém, eles não tratam o desafio das atualizações dos modelos nem da confiabilidade da

classificação quando um novo comportamento de tráfego de rede é detectado. Como alternativa, alguns autores recorrem a abordagens baseadas em agrupamento para identificar novos comportamentos. Por exemplo, [Peng et al. 2018] aplica uma técnica baseada em agrupamento usada com uma abordagem para redução de recursos para classificar ataques desconhecidos. No entanto, sua abordagem só pode ser aplicada em um ambiente supervisionado, ou seja, com o conhecimento prévio do rótulo dos eventos, além disso, sua técnica é computacionalmente cara e não pode ser aplicada em tempo real.

Para superar o desafio de aplicar técnicas de aprendizagem de máquina em ambientes com comportamento conhecido, em geral, os autores recorrem a técnicas de aprendizagem de fluxo [Peng et al. 2016] [He et al. 2011]. Tais abordagens têm sido amplamente utilizadas em outros campos, como o setor financeiro, de comunicação e até em jogos [He et al. 2011]. No entanto, sua aplicabilidade em NIDS ainda está em seu início [Gates e Taylor 2007]. Por exemplo, em [Viegas et al. 2017a] o classificador Hoeffding Tree é aplicado para NIDS. Em seu trabalho, supõe-se que o rótulo do evento seja sempre conhecido e possa ser usado para atualizar o modelo conforme desejado. Por outro lado, [Muallem et al. 2019] propõe uma abordagem de aprendizagem de fluxo distribuído para detecção de intrusão. Da mesma forma, os autores assumem que o rótulo do evento é conhecido anteriormente e as atualizações do modelo podem ser executadas conforme necessário pelo administrador.

Portanto, de acordo com o nosso conhecimento, o presente trabalho é o primeiro a abordar a tarefa de atualização do modelo, assim como a confiabilidade das classificações em ambiente de produção. Em outras palavras, a abordagem proposta no presente trabalho assume que apenas um subconjunto de rótulos dos eventos é fornecido, e o modelo deve permanecer confiável, independentemente de sua atualização.

4. Um Sistema de Detecção de Intrusão Confiável Baseado em Aprendizagem por Fluxo

Nesta seção, o modelo confiável de detecção de intrusão baseado em aprendizagem por fluxo para tratar o desafio de classificação de comportamento de tráfego de rede desconhecido é detalhado. O modelo proposto é composto por três principais componentes, a *Classificação*, o *Avaliador de Confiabilidade* e *Atualização* (conforme mostrado na Figura 1). Ele se concentra na avaliação da confiabilidade das classificações ao longo do tempo, mesmo quando um comportamento de rede desconhecido é enfrentado pelo sistema. Além disso, facilita a tarefa de atualização do modelo, por meio de técnicas de aprendizado de fluxo.

O módulo de *Classificação* executa a classificação do evento de entrada (*Fluxo de Rede*). Para atingir esse objetivo, nossa proposta usa classificadores de aprendizagem de fluxo para permitir a atualização incremental do sistema, facilitando o processo de atualização dos modelos. O resultado da classificação é utilizado de duas maneiras. Primeiro, um alerta é gerado se um evento malicioso for encontrado. Segundo, o sistema encaminha a saída para o módulo *Avaliador de Confiabilidade*, que permite a avaliação da confiabilidade da classificação.

O módulo *Avaliador de confiabilidade*, por sua vez, avalia a confiabilidade da classificação por meio de um conjunto de detectores de anomalias de fluxo. Cada detector é atualizado de forma incremental, de acordo com sua classe relacionada. Por exemplo, um

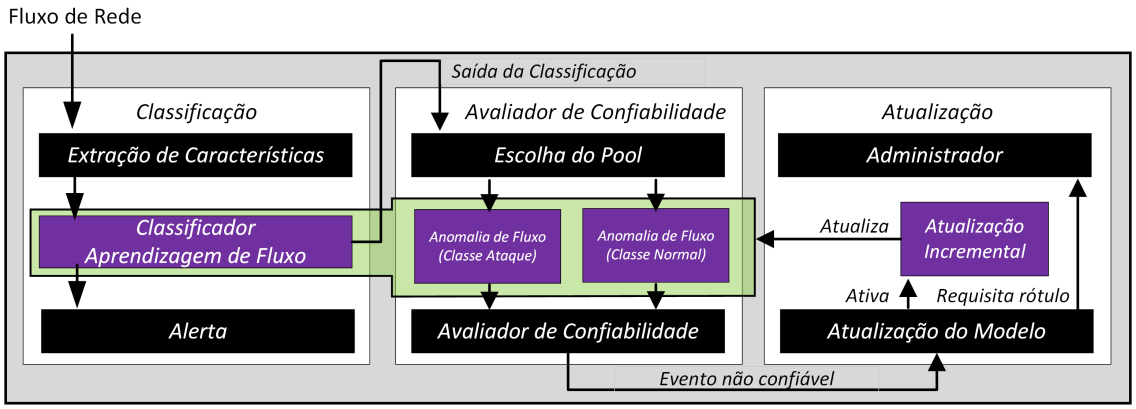

Figura 1. Arquitetura confiável de detecção de intrusão com Stream Learning.

detector de anomalia para a classe normal e outro para a classe de ataque. O objetivo de cada detector de anomalia de fluxo é permitir uma avaliação adequada das classificações, também de forma incremental e atualizada. Assim, é atualizado de forma incremental juntamente com os classificadores de aprendizagem de fluxo. Cada detector de anomalia de fluxo permite avaliar adequadamente se o evento classificado atualmente apresenta um comportamento semelhante ao usado para o treinamento. Se um evento desconhecido for encontrado (uma anomalia para uma determinada classe de eventos classificados), ele será encaminhado para o módulo de *Atualização*.

Por fim, o módulo de *Atualização*, por sua vez, utiliza os eventos classificados de uma maneira não confiável para adaptar de forma incremental os modelos do sistema, facilitando significativamente o processo de atualização do modelo. Para tanto, o módulo solicita o rótulo apropriado do evento a um administrador, que por sua vez pode avaliar manualmente o evento ou usar ferramentas de segurança conhecidas para identificá-lo adequadamente. Como o rótulo do evento é conhecido, o módulo atualiza de maneira incremental o classificador de aprendizado de fluxo e o detector de anomalia relacionado, mantendo o sistema confiável à medida que novos comportamentos de rede são enfrentados ao longo do tempo.

As próximas subseções descrevem cada uma dessas três etapas.

4.1. Classificação

Em NIDS baseado em aprendizagem de máquina, para classificar a atividade de rede, é necessário aplicar um modelo de aprendizagem de máquina através de um vetor de características que descreve o comportamento do evento. No entanto, as técnicas usadas atualmente não conseguem lidar com o comportamento desconhecido do tráfego de rede, considerando o tráfego de rede do ambiente de produção. Como consequência, as técnicas propostas exigem atualizações periódicas e constantes do modelo.

Para enfrentar o desafio de atualização do modelo, nossa proposta recorre a classificadores de aprendizagem de fluxo, o que permite a facilidade no processo de atualização do modelo, permitindo incorporar novos comportamentos no modelo atual de modo incremental. O processo de classificação começa com um evento de rede a ser classificado (*fluxo de rede*), que é encaminhado ao módulo de extração de características para gerar um vetor de características a ser classificado. O evento classificado é usado de duas ma-

neiras. Primeiro, se um evento malicioso for encontrado, ele gera um alerta relacionado. Segundo, ele é encaminhado para o módulo de *Avaliador de Confiabilidade*, que avaliará a confiabilidade da classificação, para definir se um processo de atualização do modelo de modo incremental é necessário ou não.

Portanto, através dos classificadores de aprendizagem de fluxo, nosso modelo é capaz de melhorar significativamente o processo de atualização. Isso ocorre porque somos capazes de atualizar incrementalmente o modelo de aprendizagem de máquina, em vez de executar todo o processo computacionalmente caro da tarefa de retreino do modelo. Como resultado, nosso modelo é capaz de facilitar significativamente o processo de fornecimento de modelos de classificação atualizados.

4.2. Avaliador de Confiabilidade

Independentemente do comportamento atual do ambiente, o modelo existente de aprendizagem de máquina irá executar uma decisão. Como resultado, assim que o comportamento do ambiente muda, as classificações se tornam não confiáveis, aumentando as taxas de erro é resultando em alarmes nos quais o operador não confia mais. Portanto, o objetivo do módulo *Avaliador de confiabilidade* é duplo. Primeiro, ele garante que a classificação realizada seja confiável. Em outras palavras, avalia se a classificação foi feita sobre um comportamento conhecido ou se é um novo comportamento não confiável. Segundo, define se o evento atual deve ser usado pelo nosso sistema para adaptá-lo de forma incremental. Assim, permitindo que nosso sistema incorpore de forma confiável um novo comportamento.

Para tanto, o módulo recebe como entrada as classificações executadas pelo módulo *Classificação*. Em seguida, o módulo de *escolha de pool* verifica o rótulo do evento atribuído, a fim de encaminhá-lo adequadamente para o detector de anomalias de comportamento correspondente. O módulo contém um conjunto de detectores de anomalias de comportamento, nos quais cada um é treinado para cada rótulo relacionado, isto é, um para normal e outro para ataque. Como consequência, cada detector de anomalias de comportamento é capaz de avaliar o grau de anomalia do evento em relação ao seu comportamento correspondente e conhecido, ou seja, utilizado para treinamento. Portanto, o módulo pode determinar se o evento é conhecido (usado para fins de treinamento) ou desconhecido. Por fim, de acordo com o grau de anomalia, o módulo define se um evento é confiável ou não.

Eventos não confiáveis podem ser usados para fins de atualização pelo módulo *Atualização* e possuir o seu alerta correspondente suprimido. Portanto, mantendo a confiabilidade do sistema pelo operador, que acredita que apenas eventos confiáveis devem gerar alertas ao longo do tempo. Esse processo permite determinar os eventos que devem ser usados para fins de atualização, em vez de exigir um novo processo de treinamento completo.

4.3. Atualização

O comportamento do tráfego em ambientes de rede muda com o tempo, exigindo que os modelos de aprendizagem de máquina construídos sejam atualizados. As atualizações do modelo podem levar dias ou até semanas para serem concluídas, pois a construção de uma nova base de dados de treinamento não é facilmente viável.

Portanto, o objetivo do módulo *Atualização* é atualizar o sistema proposto de forma incremental e confiável. Para atingir esse objetivo, o módulo recebe como entrada os eventos não confiáveis, conforme estabelecido pelo módulo *Avaliador de Confiabilidade*. Em seguida, o rótulo correto de evento não confiável é solicitado a um administrador. O administrador pode inspecionar manualmente o evento ou usar ferramentas conhecidas para a tarefa de rotulagem. O evento rotulado corretamente é usado para acionar a atualização do modelo, que atualiza incrementalmente o classificador de aprendizagem de fluxo (*Classificação*, Seção 4.1) e os detectores de anomalia de fluxo (*Avaliador de Confiabilidade*, Seção 4.2).

Consequentemente, o procedimento de atualização é capaz de garantir que apenas eventos rotulados corretamente sejam incorporados ao conhecimento do sistema. Portanto, nosso modelo é capaz de se adaptar de forma incremental e confiável às mudanças de comportamento do ambiente ao longo do tempo.

4.4. Discussão

O modelo de sistema de detecção de intrusão confiável baseado em aprendizagem por fluxo proposto têm como objetivo fazer uso das técnicas de aprendizagem de fluxo para facilitar o processo de atualização. Através dos detectores de anomalia de fluxo, podemos avaliar se o comportamento atual do ambiente é conhecido pelo nosso modelo existente de aprendizagem de máquina. Isso permite que nosso modelo avalie a confiabilidade da classificação, considerando que comportamentos desconhecidos são potencialmente erros de classificação. Além disso, através de técnicas de aprendizado de fluxo, podemos incorporar comportamentos desconhecidos encontrados em nosso modelo, facilitando significativamente o processo de atualização do modelo de maneira confiável. Como resultado, nosso modelo é capaz de se adaptar às mudanças no comportamento do ambiente de produção, mantendo sua confiabilidade para o administrador do sistema.

5. Avaliação

As próximas subseções descrevem o conjunto de dados utilizado, o processo de criação do modelo e sua acurácia no conjunto de dados utilizado.

5.1. Ambiente de Testes

Para avaliação da nossa proposta utilizamos o dataset *Fine-grained Intrusion Dataset* (FGD) de [Viegas et al. 2017b]. A base de dados utilizada compreende uma série de comportamentos (fluxos) de rede que podem aparecer nas variações de rede ao longo do tempo, em que os modelos de aprendizagem de máquina devem ser capazes de lidar nos ambientes de produção. As propriedades são definidas independentemente das alterações inerentes ao comportamento do tráfego de rede ao longo do tempo. Em outras palavras, o conjunto de dados FGD compreende a variabilidade natural do tráfego da rede, devido às limitações da reprodução do tráfego de rede real em um ambiente controlado. Isso permite a avaliação adequada dos modelos de aprendizagem de máquina de acordo com cada variação possível do comportamento da rede no ambiente de produção, incluindo conteúdo do serviço, tipo de serviço e ataque. O conjunto de dados contém eventos em três situações: *conhecido* (comportamentos usados no treinamento), *similar* (comportamento semelhante aos dados de treinamento) e *novo* (não disponível durante o tempo de treinamento, portanto, com um comportamento diferente). A semelhança é definida

de acordo com o critério do administrador da rede, considerando que ele opera a rede e compreende as possíveis variações de tráfego. Para fornecer esse controle de dados de granularidade fina, o FGD foi criado em um ambiente controlado, por meio de um *Honeypot* e através de ferramentas de geração de tráfego de ataque conhecidas, descritas em mais detalhes em [Viegas et al. 2017b]. Além disso, cinco serviços foram utilizados para gerar tráfego normal. É importante ressaltar que os comportamentos do cliente e do invasor variam significativamente durante o período de monitoramento do ambiente de rede [Viegas et al. 2017b].

A base de dados construída permite a avaliação detalhada dos esquemas de detecção de NIDS baseados em aprendizagem de máquina propostos em relação à sua confiabilidade ao enfrentar as propriedades do ambiente de produção. o ambiente de testes foi executado por 10 horas, durante as quais, 100 nós foram usados como clientes para gerar tráfego benigno, enquanto 10 nós foram usados como maliciosos para gerar tráfego de ataque. Durante a execução do ambiente de testes, o comportamento do cenário atual foi variado de *conhecido*, *similar* e *novo* em um intervalo de janela de 30 minutos, conforme descrito em [Viegas et al. 2017b]. No total, a base de dados gerada compreende 165 GB de dados, com 560 milhões de pacotes de rede, dos quais 740 mil deles são ataques. O algoritmo de extração de características agrupou os eventos em intervalos de 2 segundos enquanto extraiu 49 características de acordo com o fluxo das comunicações sobre a base de dados do trabalho [Viegas et al. 2017b].

5.2. Construção do modelo

Devido ao tráfego gerado ser desbalanceado (apenas aproximadamente 2% dos fluxos de rede são classificados como ataques), um processo de sub amostragem aleatória foi realizada nos dados de treinamento, para produzir uma distribuição uniforme entre as classes. Para avaliar adequadamente o impacto do comportamento do tráfego de rede nos modelos de aprendizagem de máquina, os classificadores foram treinados através do comportamento *Conhecido*, enquanto foram avaliados durante todo o restante do conjunto de dados FGD. Para a avaliação do nosso modelo proposto, utilizados o classificador de aprendizado de fluxo *Hoeffding Tree* no módulo de *Classificação*. Por outro lado, o módulo de *Avaliador da Confiabilidade* contém dois detectores de anomalia de fluxo, um para a classe normal e outro para a classe de ataque. O detector de anomalia de fluxo foi implementado através do classificador *half-space tree one-class* [Tan et al. 2011], com 25 árvores, uma profundidade de 15 e um limite de anomalia de 0,5, valores que foram definidos de modo empírico após uma rodada de testes. Os classificadores de aprendizagem de fluxo foram implementados usando a API MOA [Moa 2020]. Em cada classificador foi avaliado a quantidade de taxas de *falso-negativo* (FN) e *falso-positivo* (FP). Os algoritmos de aprendizado de fluxo foram treinados apenas com os dados do cenário conhecido, enquanto os testes foram efetuados sobre todo o conjunto de dados do FGD, com ou sem o módulo de *Atualização* em execução.

5.3. Avaliação da Confiabilidade

Inicialmente avaliamos a acurácia obtida pelo algoritmo tradicional no FGD, usando o mesmo classificador de aprendizagem de fluxo o *Hoeffding Tree*, com o mesmo conjunto de parâmetros em todo o conjunto de dados. Na Figura 2 é possível observar um aumento significativo nas taxas de FP e FN ao classificar os comportamentos *similares* e *novos*.

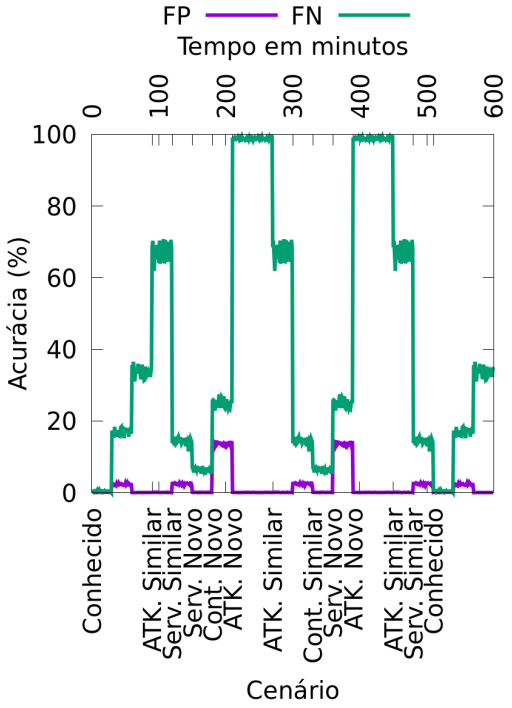

Figura 2. Acurácia do Hoeffding Tree ao longo do tempo.

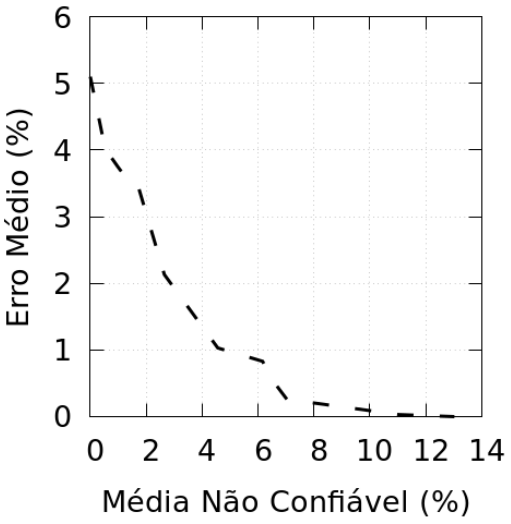

Figura 3. Curva de rejeição e erro através do *Avaliador de Confiabilidade*.

A presente avaliação possui como objetivo responder a três perguntas de pesquisa: (*P*1) *A técnica de avaliação proposta é capaz de detectar classificações não confiáveis?* (*P*2) *A técnica de avaliação da confiabilidade proposta ajuda a melhorar a acurácia do sistema?* (*P*3) *A técnica de atualização proposta é capaz de incorporar comportamentos desconhecidos no modelo existente?*

A primeira avaliação da nossa proposta objetiva responder à pergunta *P*1 e avalia se a técnica proposta de *Avaliador de Confiabilidade* é capaz de detectar classificações não confiáveis, ou seja, detectar erros de classificação. Para isso, selecionamos um ponto de operação de confiabilidade em 0,5% da taxa de erro nos cenários *Conhecido* e *Similar*.

A figura 3 apresenta a relação entre as classificações não confiáveis e a taxa de erro, quando apenas classificações confiáveis são consideradas. Usando o ponto de operação selecionado (0,5% da taxa de erro, alcançado em 10% de classificações não confiáveis), aplicamos nossa técnica proposta *Avaliador de Confiabilidade* em todo o conjunto de dados FGD sem atualizações incrementais do modelo. A figura 4 mostra a taxa de classificação não confiável ao longo do tempo, quando o ponto operacional selecionado é usado no FGD. Nosso modelo é capaz de garantir de maneira adequada a confiabilidade do modelo, considerando que sua taxa de classificação não confiável aumenta quando o comportamento de tráfego desconhecido é encontrado.

Para responder à pergunta *P*2, avaliamos a taxa de erro obtida, quando apenas classificações confiáveis são consideradas. A figura 5 mostra a taxa de erro relacionada, sem atualizações incrementais do modelo, quando apenas classificações confiáveis são consideradas para o cálculo da acurácia. É possível notar que nosso modelo proposto é capaz de manter a confiabilidade do sistema (taxas de acurácia), mesmo quando enfrenta

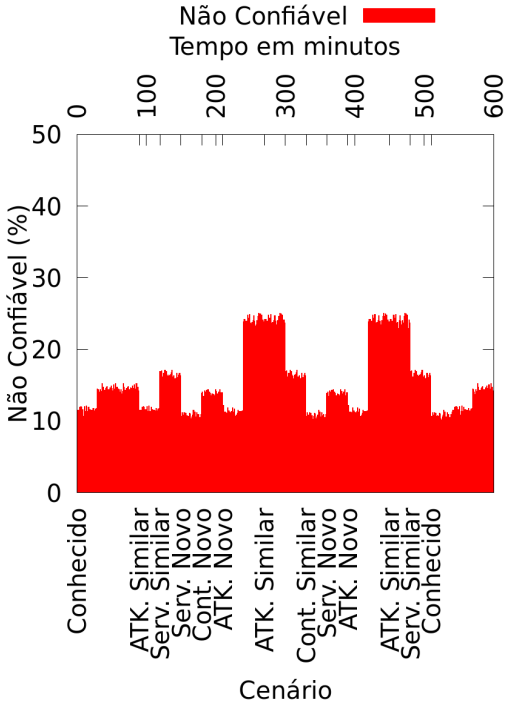

Figura 4. Rejeição ao longo do tempo sem atualização incremental.

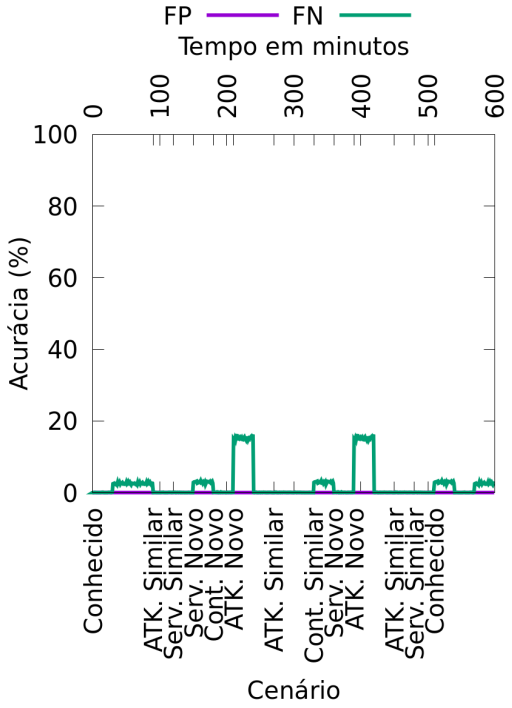

Figura 5. Acurácia ao longo do tempo sem atualização incremental.

um comportamento de tráfego desconhecido, apesar de uma maior taxa de classificação não confiável.

5.4. Stream Learning

Por fim, para responder à pergunta *P*3, atualizamos incrementalmente nosso modelo utilizando as instâncias não confiáveis. A figura 6 exibe a acurácia obtida, enquanto a figura 7 mostra a taxa de eventos não confiáveis quando atualizações incrementais do modelo são efetuadas. É possível notar que nosso modelo proposto foi capaz de incorporar o comportamento desconhecido no modelo existente. No entanto, quando atualizações incrementais do modelo são realizadas, foi possível diminuir consideravelmente a taxa de rejeição do sistema, demonstrando que o modelo proposto é capaz de facilitar o processo de atualização do sistema de maneira confiável.

6. Conclusão

As abordagens atuais baseadas em aprendizagem de máquina para NIDS não são capazes de detectar novos comportamentos de rede sem exigir um processo computacional custoso de retreinamento, assim como assistência de um especialista. O modelo proposto neste artigo abordou a confiabilidade das classificações, ao enfrentar um novo comportamento de tráfego de rede, além de facilitar as atualizações do modelo. Nosso modelo proposto usou técnicas de aprendizagem de fluxo para incorporar de forma incremental novos comportamentos de tráfego no modelo existente. Para garantir a confiabilidade das classificações de modo incremental, avaliamos a confiabilidade da classificação através de detectores de anomalia de fluxo. O modelo proposto foi capaz de detectar classificações não confiáveis,

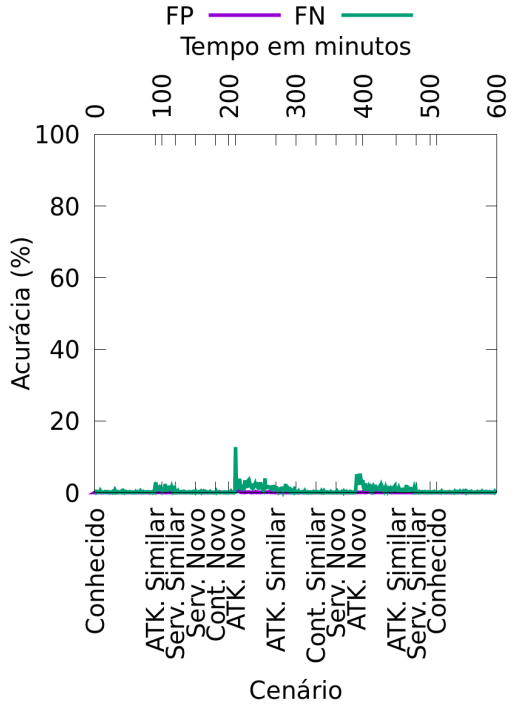

Figura 6. Acurácia ao longo do tempo com atualização incremental.

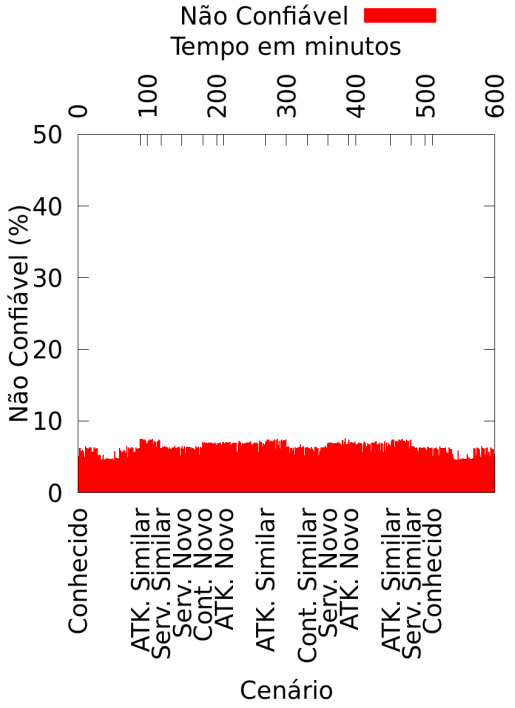

Figura 7. Rejeição ao longo do tempo com atualização incremental.

mantendo a acurácia do sistema, além de facilitar o processo de atualização do modelo, fazendo uso das técnicas de aprendizagem de fluxo. Como trabalho futuro, focaremos na escalabilidade do nosso modelo em um ambiente distribuído, mantendo a confiabilidade do sistema.

Referências

Abreu, V., Santin, A. O., Viegas, E. K., e Stihler, M. (2017). A multi-domain role activation model. In *2017 IEEE International Conference on Communications (ICC)*. IEEE.

dos Santos, R. R., Viegas, E. K., Santin, A., e Cogo, V. V. (2020). A long-lasting reinforcement learning intrusion detection model. In *Advanced Information Networking and Applications*, pages 1437–1448. Springer International Publishing.

Gates, C. e Taylor, C. (2007). Challenging the anomaly detection paradigm: A provocative discussion. pages 21–29. Proc. 2006 Work. New Secur. Paradig.

He, H., Chen, S., Li, K., e Xu, X. (2011). Incremental learning from stream data. pages 1901–1914. IEEE Trans. Neural Netw. 22.

Kugler, E., Santin, A. O., Cogo, V. V., e Abreu, V. (2020). A reliable semi-supervised intrusion detection model: One year of network traffic anomalies. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. IEEE.

Loganathan, G., Samarabandu, J., e Wang, X. (2018). Real-time intrusion detection in network traffic using adaptive and auto-scaling stream processor. In *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE.

Moa (2020). Moa. disponível em: https://moa.cms.waikato.ac.nz/.

Muallem, A., Shetty, S., Hong, L., e Pan, J. (2019). Tddeht: Threat detection using distributed ensembles of hoeffding trees on streaming cyber datasets. pages 219–224. Proc. - IEEE Mil. Commun. Conf. MILCOM.

Peng, J., Choo, K.-K. R., e Ashman, H. (2016). User profiling in intrusion detection: A review. volume 72, pages 14–27. Elsevier BV.

Peng, K., Leung, V., e Huang, Q. (2018). Clustering approach based on mini batch kmeans for intrusion detection system over big data. pages 11897–11906. IEEE Access.

P.Singh e Venkatesan, M. (2018). Hybrid approach for intrusion detection system. pages 1–5. Proc. 2018 Int. Conf. Curr. Trends Towar. Converging Technol. ICCTCT.

Sommer, R. e Paxson, R. (2010). Outside the closed world: On using machine learning for network intrusion detection. pages 305–316. IEEE Symp. Secur. Priv.s.

Sovilj, D., Budnarain, P., Sanner, S., Salmon, G., e Rao, M. (2020). A comparative evaluation of unsupervised deep architectures for intrusion detection in sequential data streams. volume 159, page 113577. Elsevier BV.

Tan, S., Ting, K., e Liu, T. (2011). Fast anomaly detection for streaming data. pages 1511–1516. IJCAI International Joint Conference on Artificial Intelligence, vol. 22.

Tavallaee, M., Stakhanova, N., e Ghorbani, A. A. (2010). Toward credible evaluation of anomaly-based intrusion-detection methods. pages 516–524. IEEE Trans. Syst. Man Cybern. 5.

Tobi, A. e Duncan, I. (2019). Improving intrusion detection model prediction by threshold adaptation. pages 1–42. Information.

Viegas, E., Santin, A., e Abreu, N. N. A. (2017a). A resilient stream learning intrusion detection mechanism for real-time analysis of network traffic. page 978–983. IEEE Glob. Telecommun. Conf. GLOBECOM.

Viegas, E., Santin, A., Bessani, A., e Neves, N. (2019). BigFlow: Real-time and reliable anomaly-based intrusion detection for high-speed networks. *Future Generation Computer Systems*, 93:473–485.

Viegas, E. K., Santin, A. O., e Oliveira, L. S. (2017b). Toward a reliable anomaly-based intrusion detection in real-world environments. *Computer Networks*, 127:200–216.

Yin, C., Xia, L., Zhang, S., Sun, R., e Wang, J. (2017). Improved clustering algorithm based on high-speed network data stream. volume 22, pages 4185–4195. Springer Science and Business Media LLC.