UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Arnon Afonso Vieira Carrasco

Aprendizado de máquina aplicado a detecção de botnets utilizando aprendizado ativo

Uberlândia, Brasil

2022

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Arnon Afonso Vieira Carrasco

Aprendizado de máquina aplicado a detecção de botnets utilizando aprendizado ativo

Trabalho de conclusão de curso apresentado à Faculdade de Engenharia Mecatrônica da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Engenheiro Mecatrônico.

Orientador: Prof. Dr. Rodrigo Sanches Miani

Universidade Federal de Uberlândia – UFU

Faculdade de Engenharia Mecânica

Bacharelado em Engenharia Mecatrônica

Uberlândia, Brasil

2022

Arnon Afonso Vieira Carrasco

Aprendizado de máquina aplicado a detecção de botnets utilizando aprendizado ativo

Trabalho de conclusão de curso apresentado à Faculdade de Engenharia Mecatrônica da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Engenheiro Mecatrônico.

Trabalho aprovado. Uberlândia, Brasil, 08 de Julho de 2022:

Prof. Dr. Rodrigo Sanches Miani
Orientador

Prof. Dr. Marcelo Zanchetta do Nascimento
Professor

Profa. Dra. Elaine Ribeiro de Faria
Professor

Uberlândia, Brasil
2022

Resumo

Segurança de redes, Aprendizado de máquina, Fluxo contínuo de dados, Aprendizado ativo

Este trabalho consiste na análise de algoritmos de classificação, voltados para a detecção de ataques em fluxos de dados, com o objetivo de melhorar a segurança de redes de computadores. Para isto, os algoritmos foram aplicados na base de dados CTU-13, que consiste em um conjunto de tráfico de *botnets* previamente capturado e tratado. Com o intuito de analisar os diferentes comportamentos dos classificadores ao utilizar estratégias de atraso de rotulação e de percentual de rotulagem, aliados com o aprendizado ativo. Ao utilizar essas técnicas, foi possível perceber uma variação na desempenho do classificador utilizado, principalmente quando analisadas as métricas obtidas dos testes relacionados a alteração do percentual de rotulagem juntamente com o aprendizado ativo, no qual os valores de revocação e precisão diminuíram.

Lista de ilustrações

Figura 1 – Arquitetura de uma *botnet*. Adaptado de (KHATTAK; RAMAY; KHAYAM,2014). 14
Figura 2 – Esquema de obtenção de rótulos em fluxo contínuo de dados (OLIMPIO et al., 2021). 24
Figura 3 – Representação do aprendizado ativo. (OLIMPIO et al., 2021) 25

Lista de tabelas

Tabela 1 – Tabela com processamento tradicional de dados 16
Tabela 2 – Quantidade de dados em cada cenário do CTU-13 21
Tabela 3 – Distribuição de classes em cada cenário do CTU-13 21
Tabela 4 – Resultados de classificação do CTU-13 utilizando atraso na rotulação dos fluxos 26
Tabela 5 – Resultados de classificação do CTU-13 utilizando a técnica de aprendizado ativo, juntamente com a alteração do percentual de rotulagem . 28
Tabela 6 – Resultados de classificação do CTU-13 utilizando a combinação de estratégias de percentual de rotulagem com o atraso de rotulagem 29
Tabela 7 – Resultados de classificação do CTU-13 utilizando a combinação de estratégias de percentual de rotulagem com o atraso de rotulagem. Continuação da Tabela 6 30
Tabela 8 – Resultados de classificação do CTU-13 utilizando a combinação de estratégias de percentual de rotulagem com o atraso de rotulagem. Continuação da Tabela 7 31

Lista de abreviaturas e siglas

ASHT	*Adaptive Size Hoeffding Tree*
DDoS	*Distribuited Denial of Service*
CTU	*Czech technical university*
FAR	*False Alarm Rate*
IDS	*Intrusion Detection Systems*
IP	*Internet Protocol*
MOA	*Massive Online Analysis*
N/A	*Not Available*

Sumário

1 INTRODUÇÃO 8
1.1 Contextualização e motivação 8
1.2 Objetivos 10
1.3 Organização da monografia 10

2 REVISÃO BIBLIOGRÁFICA 11
2.1 Segurança da informação 11
2.2 Sistemas de detecção de intrusão 12
2.3 Botnets 13
2.4 Aprendizado de máquina 15
2.5 Mineração de fluxo 15
2.6 Aprendizado ativo 17
2.7 Trabalhos relacionados 18
2.8 Considerações Finais 19

3 DESENVOLVIMENTO 20
3.1 MOA 20
3.2 CTU-13 20
3.3 Métricas avaliadoras de desempenho 21
3.4 Experimentos 23
3.4.1 Validação de Experimentos 23
3.5 Resultados 26
3.6 Comparação com a literatura 28

4 CONCLUSÕES 33

REFERÊNCIAS 35

1 Introdução

A modernização e popularização das Tecnologias de Informação e Comunicação na última década foram responsáveis pelo aumento no volume de dados que trafegam na rede mundial de computadores. Com a digitalização de serviços financeiros essenciais, juntamente com o crescimento massivo de usuários das diversas redes sociais existentes atualmente, tornou-se comum o envio de dados pessoais para estes serviços com o intuito de usufruí-los da melhor maneira possível. Entretanto, muitos desses dados são de caráter sigiloso, não podendo serem expostos de qualquer maneira pois, ao faze-lo, pode trazer problemas de segurança aos envolvidos. Por esta razão, questões como cibersegurança e segurança da informação são amplamente debatidos atualmente. Segundo o relatório da Accenture (2022) sobre cibersegurança, em uma pesquisa analisando mais de 500 companhias nos EUA, houve um aumento de 31% no número de ataques em 2021 em relação ao ano anterior. Desta forma, com o desenvolvimento e popularização da Internet, a necessidade de garantir a proteção e segurança dos dados que trafegam nessa rede também aumenta e, por isso, a área de pesquisa em segurança da informação torna-se cada vez mais ativa e necessária (LI et al., 2012).

1.1 Contextualização e motivação

Os ataques possuem diversos propósitos, além de terem características distintas e não apresentarem sintomas óbvios na maioria das vezes, em muitos casos o ataque é detectado quando uma funcionalidade da aplicação já foi comprometida. Uma espécia de ataque, muito utilizado atualmente, são os *botnets*, que consistem em redes formadas por computadores comprometidos por *malwares* apelidados de *bots* (SILVA et al., 2013). *Botnets* são utilizadas para uma variedade de atividades maliciosas como *fraud click, phishing, spamming*, entrega de *malware* e (*Distributed Denial Of Service*) DDoS (GAONKAR et al., 2020). Alguns exemplos de *botnets* utilizadas em ataques incluem Mirai, Hajime, Aidra, Bashlite, Dofloo e Tsunami (BEZERRA et al., 2018).

Pela necessidade de uma maior segurança no fluxo de dados que chega em redes locais, surgiram os IDS (sigla em inglês para *Intrussion Detection System*), que consistem em *hardwares* ou *softwares* que monitoram continuamente a rede buscando identificar e alertar os administradores de segurança sobre tentativas de intrusão. Os IDS carregam as funções de coleta, análise e armazenamento de informações contidos no tráfego da rede em questão. Frequentemente, dois métodos são utilizados para a detecção de comportamentos maliciosos, os baseados em assinaturas e os baseados em detecção de anomalias (BHUYAN; BHATTACHARYYA; KALITA, 2017). Métodos baseados em assinaturas tem

a função de comparar o tráfego em análise com uma base de dados, de ataque, previamente catalogada. Se por acaso, algum ataque presente na base de dados em questão, seja identificado no tráfego, o IDS irá categorizá-lo como malicioso. A maior questão destes sistemas por assinaturas é que a identificação é feita baseando-se em um comportamento anteriormente reconhecido como não-malicioso, desta forma, caso alguma identificação saia do padrão pré-analisado, ela será identificada como um possível ataque.

A respeito da coleta de dados, eles podem ser apresentados de forma individual, em pacotes, ou em grupos na forma de fluxo de pacotes. O fluxo é uma agregação de pacotes de redes transmitidos que compartilham algumas características em comum, tais como: o mesmo endereço de *Internet Protocol* (IP) de origem e destino, porta de origem e destino e protocolo de transporte dentro de uma janela de tempo (RING et al., 2019).

Ao contrário dos algoritmos tradicionais de aprendizado de máquina, os quais são baseados em aprendizado a partir de conjuntos estáticos de dados (ou lotes), os algoritmos de mineração de fluxos contínuos são projetados para lidar com aprendizado incremental sem armazenar todos os dados, processando os dados como um fluxo contínuo e infinito (COSTA et al., 2018). Durante o aprendizado para classificar os dados de um fluxo contínuo, obter os rótulos verdadeiros das instâncias pode requerer um esforço maior e até um custo excessivo. O aprendizado ativo é uma sub-área do aprendizado de máquina, cujo objetivo é alcançar maior desempenho utilizando o menor número possível de instâncias rotuladas (SETTLES, 2010). Sendo assim, o custo de se obter dados rotulados é minimizado (ALBERT; BERNHARD; GEOFF, 2011).

De acordo com Ribeiro, Paiva e Miani (2020), a aplicação da mineração de fluxos contínuos de dados mostra-se uma alternativa promissora para a construção de novos modelos de detecção de *botnet*. Contudo, ela ainda precisa ser analisada sob uma ótica mais abrangente, seja utilizando diferentes tipos de algoritmos de classificação e conjuntos de dados e investigando outras estratégias para avaliação dos modelos, como o número de exemplos rotulados. Outros pontos relevantes envolvem (i) buscar a maximização do desempenho de classificação minimizando o número de instâncias rotuladas, (ii) entender o impacto da distribuição das classes dos conjuntos de dados nos resultados gerados pelos modelos, e (iii) utilizar métricas relacionadas ao aprendizado incremental para comparação dos resultados.

Em cenários com fluxos contínuos, os dados podem chegar rapidamente e a partir de múltiplas fontes de dados (GROSSI; TURINI, 2012). Isso impõe uma restrição de que os algoritmos de mineração de dados verifiquem os dados apenas uma vez. Logo, é desafiador ter um modelo de classificação dos dados eficiente e que mantenha o controle dos dados que já passaram no fluxo (GROSSI; TURINI, 2012).

1.2 Objetivos

Este trabalho tem o objetivo de colaborar com o desenvolvimento de um IDS que atenda as necessidades cotidianas de problemas de detecção de intrusão, como por exemplo: geração contínua de dados e constante atualização do modelo de detecção de intrusão. Para isto, este trabalho tem o intuito de comparar e analisar o impacto do atraso da entrega de instâncias rotuladas para a atualização do modelo e, por fim, analisar o impacto da seleção de um conjunto reduzido de amostras para receberem o rótulo para a atualização do modelo de classificação.

Foi tomado como base o trabalho de Olimpio et al. (2021), que realizou 3 tipos diferentes de experimentos para teste de diferentes classificadores na base de dados *CICIDS2017*. Os experimentos feitos por Olimpio et al. (2021) consistiram em primeiramente realizar um teste de atraso na rotulação dos dados, por meio do método de classificação, e assim variar o tempo de atraso. O segundo teste foi feito com aprendizado ativo, porém variando a porcentagem de rótulos entregues ao classificador. E por fim, o terceiro teste combinou as duas estratégias anteriores em um único teste. Neste trabalho, seus testes foram replicados porém na base de dados CTU-13, invés de *CICIDS2017*. Os testes também foram realizados somente no classificador *OzaBagASHT*.

1.3 Organização da monografia

Este trabalho está dividido da seguinte forma. O Capítulo 2 trata da revisão bibliográfica, com intuito de fundamentar a teoria. O Capítulo 3 descreve em detalhes o *framework* utilizado, a base de dados analisada, os testes realizados, os resultados obtidos em cada experimento, e por fim, comparação com estudos atuais sobre o tema. Após isto, o Capítulo 4 conclui este trabalho relatando o que foi observado nos resultados dos experimentos, analisando as métricas estatísticas aplicadas em cada uma das estratégias, e desta forma, propõe próximos passos a serem seguidos em possíveis trabalhos futuros.

2 Revisão Bibliográfica

Neste capítulo será feita uma revisão bibliográfica dos principais temas correlacionados a cibersegurança, para um melhor entendimento da base teórica.

2.1 Segurança da informação

Segundo o Glossário de Segurança da Informação (2019), publicado pelo Gabinete de Segurança Institucional da presidência da república, segurança da informação constitui-se em ações que objetivam viabilizar e assegurar a disponibilidade, a integridade, a confidencialidade e a autenticidade das informações. A segurança de informações visa garantir a integridade, confidencialidade, autenticidade e disponibilidade das informações processadas pela instituição (TCU, 2012).

Os três conceitos que formam a base da segurança de computadores são detalhados a seguir (STALLINGS, 2015):

- Confidencialidade: assegurar que informação privada ou confidencial não esteja disponível ou seja divulgada a indivíduos não autorizados (confidencialidade de dados). Além disso, garantir que os indivíduos controlem ou influenciem quais informações relacionadas a eles podem ser coletadas e armazenadas, e por quem e para quem essas informações podem ser divulgadas (privacidade). Uma perda de confidencialidade é a divulgação não autorizada de informações;
- Integridade: garantir que os programas e informações são alterados apenas de forma especificada e autorizada (integridade de dados). Também deve garantir que um sistema execute sua função pretendida de maneira intacta, livre de manipulação deliberada, inadvertida ou não autorizada (integridade do sistema). Uma perda de integridade é a modificação não autorizada ou destruição de informações;
- Disponibilidade: assegurar que os sistemas funcionem prontamente e o serviço não seja negado a usuários autorizados. Uma perda de disponibilidade é a interrupção do acesso ou uso de informações ou de um sistema de informações.

A abertura da Internet para uso comercial no início dos anos 90 aumentou a importância das políticas de segurança para transações remotas. Dados sensíveis (como senhas, informações das transações e dos próprios usuários, etc.) precisaram ser protegidos em trânsito, e os nós da Internet tiveram que ser protegidos contra acessos indesejáveis (GOLLMANN, 2010).

Ainda na questão de segurança de dados, segundo o TCU (2012), é importante zelar pela segurança das informações pois a informação é um ativo muito importante para qualquer instituição, podendo ser considerado até mesmo o recurso patrimonial mais crítico. Informações adulteradas, não disponíveis, sob conhecimento de pessoas de má-fé, ou de concorrentes podem comprometer significativamente, não somente a imagem da instituição perante terceiros, como também o andamento dos próprios processos institucionais.

2.2 Sistemas de detecção de intrusão

Segundo Viegas, Santin e Oliveira (2017), um típico IDS é composto de quatro módulos:

(i) Coletador do evento: módulo responsável por coletar eventos do ambiente para monitoramento posterior, neste caso, analisando pacotes de rede de uma placa de rede.

(ii) Pré-processamento: este módulo performa o processamento necessário antes que a ferramenta de detecção seja executada nos eventos coletados, no caso, seria a extração de um conjunto de funcionalidades.

(iii) Detecção: módulo baseado no evento pré-processado que a ferramenta de detecção uso para decidir quando um evento é considerado normal ou uma tentativa de intrusão.

(iv) Alerta: Por fim, se um evento é considerado uma tentativa de intrusão, este módulo gera um alerta.

De acordo com Stallings (2014), intrusões podem ser divididas em 2 classes, passivas e ativas. Sendo ataques passivos aqueles que não envolvem qualquer alteração dos dados. Como por exemplo, o vazamento do conteúdo de uma mensagem importante para determinada organização ou análise de tráfego cujas mensagens não estejam criptografadas. Enquanto isso, ataques ativos são categorizados da seguinte forma: disfarce, repasse, modificação de mensagens e negação de serviço.

1. Um disfarce ocorre quando uma entidade finge ser outra diferente. Um ataque de disfarce normalmente inclui uma das outras formas de ataque ativo. Por exemplo, sequências de autenticação podem ser capturadas e reproduzidas depois que houver uma delas, válida, permitindo assim que uma entidade autorizada com poucos privilégios obtenha alguns extras, personificando uma que os tenha

2. Repasse envolve a captura passiva de uma unidade de dados e sua subsequente retransmissão para produzir um efeito não autorizado.

3. Modificação de mensagens simplesmente significa que alguma parte de uma mensagem legítima é alterada, ou que as mensagens são adiadas ou reordenadas, para produzir um efeito não autorizado. Por exemplo, uma mensagem significando "Permitir que John Smith leia o arquivo confidencial contas" é modificada para "Permitir que Fred Brown leia o arquivo confidencial contas".

4. A negação de serviço impede ou inibe o uso ou gerenciamento normal das instalações de comunicação. Esse ataque pode ter um alvo específico; por exemplo, uma entidade a suprimir todas as mensagens dirigidas para determinado destino (por exemplo, o serviço de auditoria de segurança). Outra forma de negação de serviço é a perturbação de uma rede inteira, seja desativando-a ou sobrecarregando-a com mensagens, a fim de prejudicar seu desempenho.

Uma outra característica vital de um IDS é a forma com que as detecções são realizadas. Existem dois sistemas principais de detecção de intrusão: os sistemas baseados em assinaturas e os sistemas baseados em anomalias (BHUYAN; BHATTACHARYYA; KALITA, 2017).

Quando um ataque é identificado pela comunidade de segurança, é desenvolvido um conjunto de regras chamadas de assinaturas, e que podem ser usadas por IDSs para identificar uma ameaça através de comparação. A abordagem baseada em assinatura é conhecida por ter um baixo número de falsos positivos e falsos negativos, pois a sua assinatura é conhecida pela base de ataques. Porém, no reconhecimento de ataques baseado em assinaturas, não há como distinguir novos ataques até que a base seja atualizada e distribuída na Internet (GARCíA-TEODORO et al., 2009).

Já as técnicas baseadas em anomalias, visam caracterizar uma alteração no padrão normal de comportamento da rede. Esse padrão é estabelecido anteriormente, analisando o fluxo de pacotes normais e criando assim, uma referência que será utilizada para a identificação do desvio de comportamento da rede. Esse tipo de técnica é também conhecida como detecção baseada em comportamento (BHUYAN; BHATTACHARYYA; KALITA, 2017). Geralmente, para construir a base de comportamento normal e ataques, são utilizadas técnicas estatísticas ou de aprendizado de máquina (ZARPELãO et al., 2017). Elas podem identificar ataques desconhecidos, apesar de poderem atingir uma alta taxa de falsos positivos e falsos negativos (MAHONEY; CHAN, 2001).

2.3 Botnets

Uma *botnet* é um conjunto de *bots*, que são dispositivos interconectados via internet, infectados por *malware*, e comandados por um criador, o *botmaster* (NOGUEIRA, 2016). Através de ataques coordenados, *botnets* são usadas em várias atividades ilegais;

por exemplo, lançando ataques distribuídos de negação de serviços (*Distributed Denial Of Service* - DDoS), e obtendo lucro com exploração de informação de usuários normais (NOGUEIRA, 2016). A preparação de um ataque envolve recrutar, explorar e infectar os dispositivos alvo para que se tornem parte de uma *botnet* (MIRKOVIC; REIHER, 2004).

A Figura 1 mostra a arquitetura típica de um *botnet*. Em tal cenário, existem os elementos funcionais que são: o *botmaster*, entidade que tem a capacidade de gerenciar toda a estrutura de rede maliciosa; seus *bots*, que compreendem os dispositivos comprometidos e integrantes da botnet, mas que pode ser definido também como um programa de computador instalado em uma máquina a qual permite que um atacante (*botmaster*) execute comandos arbitrários no sistema infectado; a infraestrutura de comando e controle (C & C) (elemento crítico na arquitetura de uma *botnet* (CERON, 2010) que permite que comandos sejam enviados às máquinas comprometidas; e um protocolo de comunicação que é usado para enviar mensagens para os *bots*, por exemplo, HTTP, IRC e P2P (CERON, 2010).

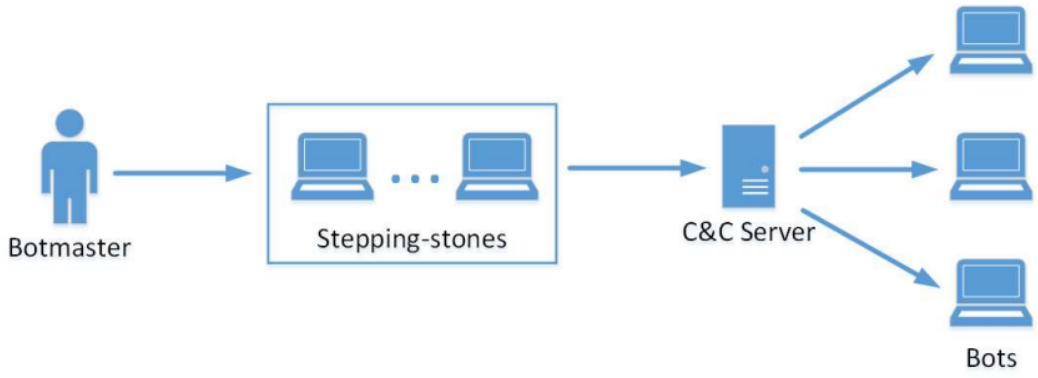

Figura 1 – Arquitetura de uma *botnet*. Adaptado de (KHATTAK; RAMAY; KHAYAM,2014).

Cada máquina comprometida apresenta um conjunto de funcionalidades pré-programadas que podem ser invocadas remotamente pelo *botmaster*. Além disso, os *bots* possuem, na maioria dos casos, a opção de atualização remota. Com isso, novas funcionalidades podem ser incorporadas à rede sem a necessidade de renovação da estrutura ou do comprometimento de novas máquinas (CERON, 2010).

Segundo Silva et al. (2013), a detecção de *botnets* deve ser uma das primeiras ações que devem ser tomadas para combater ameaças a segurança de rede. Dado o potencial das *botnets* em conduzir diferentes atividades maliciosas, as técnicas de detecção tem um importante papel neste processo. As técnicas de detecção podem ser classificadas em duas categorias principais: com base em configuração de *honeynets*, ou com uso de Sistemas de Detecção de Intrusão. A categoria de IDSs foi subdivida em sistemas baseados em

assinatura e sistemas baseados em anomalia (SILVA et al., 2013), (KHATTAK et al., 2014) e (GARCIA et al., 2014).

Entretanto, as novas gerações de *botnets* aproveitam dos avanços nas tecnologias e comunicação, aumentando sua velocidade de adaptação para protocolos de comunicação, políticas de segurança, e também seus alvos. Esses fatores criam vários desafios para prevenir e detectar a ameaça das *botnets*. *Botnets* são consideradas alvos em movimento, o que significa que todos os aspectos relacionados a *botnets* incluindo detecção, mitigação e resposta, estão mudando ao longo do tempo. Não existem técnicas de mitigação ou detecção que ofereçam uma solução permanente. De forma similar, diferentes tipos de *stakeholders*, por exemplo, empresas, governos, redes, e ISPs, empregam diferentes formas e objetivos para endereçar o problema das *botnets* (KARIM et al., 2014). Portanto, estudar novas formas de detectar e mitigar *botnets* assim como melhorar a compreensão acerca da aplicabilidade das técnicas já existentes são relevantes problemas de pesquisa.

2.4 Aprendizado de máquina

Segundo Mitchell (1997), a área de aprendizado de máquina se preocupa em questionar em como construir programas de computador que evoluem automaticamente com sua experiência. Aprendizado de máquina também pode ser definido como o processo de solucionar um problema prático por 1) obter um conjunto de dados, e 2) construir um modelo estatístico, a partir de algoritmos, baseado naquele conjunto de dados. Assume-se que o modelo estatístico deva, de alguma forma, resolver o problema prático em questão (BURKOV, 2019).

A utilização de aprendizado de máquina, na maioria dos casos tem com o intuito de realizar uma classificação de um conjunto de dados. Essa classificação consiste em atribuir categorias predefinidas a dados (CARVALHO et al., 2011),(TAN; KUMAR, 2006). Formalmente, um problema de classificação pode ser definido da seguinte maneira. Dado um conjunto de exemplos de treinamento composto por pares (x_i, c_j), no qual x_i representa um vetor de atributos de entrada que descrevem um exemplo, e c_j sua classe associada, deve-se encontrar uma função que mapeie cada x_i para sua classe associada c_j, tal que i = 1, 2, ..., n, em que n é o número de exemplos de treinamento, e j = 1, 2, ..., m, em que m é o número de classes do problema (EMBRAPA, 2017).

2.5 Mineração de fluxo

Um fluxo contínuo de dados pode ser visto como um sequência de itens de dados que chegam à medida que o tempo passa (KRAWCZYK et al., 2017), ou como um processo estocástico em que eventos ocorrem continuamente e independentemente um do outro

	Tradicional	Fluxo Contínuo
Número de execuções	Múltiplo	Único
Tempo de processamento	Ilimitado	Restrito
Uso de memória	Ilimitado	Restrito
Tipo de resultado	Preciso	Aproximado
Distribuído	Não	Sim

Tabela 1 – Tabela com processamento tradicional de dados

(GAMA, 2010). As diferenças mais relevantes que fazem que o processamento de fluxos contínuos seja diferente dos modelos relacionais convencionais são (GAMA, 2010):

- Os elementos de dados em um fluxo contínuo chegam de forma contínua;
- O sistema não possui controle sobre a ordem em que os elementos de dados chegam;
- Fluxos contínuos são potencialmente ilimitados em tamanho;
- Uma vez que um elemento de um fluxo contínuo foi processado ele é descartado ou arquivado.

De acordo com Domingos e Hulten (2000), (GAMA, 2010), (KRAWCZYK et al., 2017), mineração de fluxo é um segmento do aprendizado de máquina que visualiza os dados como um possível fluxo infinito, sendo que desta forma, os algoritmos são especialmente formulados para aprender de uma forma *online*, uma instância por vez. Adicionalmente, limitação de memória e tempo são estudados especificamente neste área, para que os algoritmos estejam páreos para estes desafios (DOMINGOS; HULTEN, 2000), (GAMA, 2010), (KRAWCZYK et al., 2017).

A Tabela 1 mostra de forma resumida, as diferenças entre o processamento tradicional de dados e o processamento de fluxos contínuos. Adaptado de (GAMA, 2010).

Vários cenários se beneficiam da análise de fluxos contínuos porque a geração automática de dados se tornou bem comum. Alguns exemplos são análise de redes, predição de dados financeiros, controle de tráfego de rede, processamento de sensores de medição, computação ubíqua, GPS, rastreamento de dispositivos móveis, mineração de *logs* de clique de usuários, análise de opinião (KRAWCZYK et al., 2017). Dentre tantos outros tópicos que possam beneficiar, em diversos aspectos, uma determinada organização, principalmente no financeiro.

Os algoritmos para classificação de fluxos contínuos de dados precisam constantemente atualizar o modelo de decisão a Ąm de tratar os fenômenos de mudança e evolução de conceito. Muitos desses algoritmos atualizam seus modelos de forma supervisionada e, para isso, supõem que o rótulo de todas as instâncias estará disponível para atualização

do modelo. Assim, eles consideram que após uma instância ser classificada, o seu rótulo estará imediatamente disponível para atualização do modelo de classificação (OLIMPIO et al., 2021).

2.6 Aprendizado ativo

Em geral, a rotulação das instâncias de um fluxo é feita por um especialista de domínio. Por conta do alto volume de dados em um tráfego, apresenta-se inviável rotular todas as instâncias presentes. Por se tratar de uma tarefa com alto custo de mão de obra e que necessita de uma grande disponibilidade de tempo dos especialistas para avaliar milhões de instâncias.

O aprendizado ativo é uma alternativa para a solução de muitos problemas modernos de aprendizado de máquina, em que dados não rotulados podem ser abundantes, mas os rótulos são difíceis, demorados ou caros de se obter (SETTLES, 2009). A ideia principal por trás do aprendizado ativo é que um algoritmo de aprendizado de máquina pode alcançar maior precisão com menos instâncias de treinamento rotuladas mas, para isso, é necessário escolher as melhores instâncias para o seu treinamento (SETTLES, 2009).

Aprendizado ativo (do inglês, *active learning*) é uma sub-área do aprendizado de máquina. A ideia principal é a de que um algoritmo de aprendizado de máquina pode alcançar alto desempenho com menos instâncias rotuladas para treinamento, sendo bem aplicado em problemas em que dados sem rótulo podem ser abundantes e obtidos facilmente, mas rótulos são difíceis, demorados, ou caros de se obter (SETTLES, 2010). Aprendizado ativo estuda como rotular seletivamente ao invés de requisitar por todos os rótulos verdadeiros dos dados (ALBERT; BERNHARD; GEOFF, 2011).

A principal diferença entre aprendizado ativo online e aprendizado ativo em fluxos contínuos está nas expectativas com relação às mudanças (ALBERT; BERNHARD; GEOFF, 2011). Aprendizado ativo online tipicamente estabelece um limite (como um limite de incerteza)e requisita o verdadeiro rótulo se o limite é excedido. Em fluxos contínuos o relacionamento entre os dados de entrada e os rótulos podem mudar, e estas mudanças podem acontecer em qualquer posição no espaço das instâncias. Assim, as estratégias de aprendizado ativo existentes podem nunca requerer instâncias em algumas regiões e assim podem nunca saber que mudanças estão acontecendo e então nunca se adaptar. Além disso, em fluxos contínuos não é possível manter o limite de decisão ou uma região de incerteza fixos, pois eventualmente o sistema pode parar de aprender e falhar em reagir às mudanças. Finalmente, o aprendizado ativo com fluxos contínuos deve preservar a distribuição dos dados que chegam na medida que mudanças podem ser detectadas assim que acontecem (ALBERT; BERNHARD; GEOFF, 2011).

2.7 Trabalhos relacionados

Visto que a questão de segurança da informação tenha recebido um aumento de atenção significável durante os últimos anos, vários trabalhos foram publicados com a intenção de criar e melhorar sistemas de IDSs que estejam mais adaptados aos desafios atuais. No trabalho de Olimpio et al. (2021) foi proposto comparar e analisar o uso de fluxos de pacotes e pacotes individuais através da análise do desempenho preditivo de classificadores de fluxos contínuo de dados, além de analisar o impacto que o atraso na rotulação das instâncias causa nos classificadores. Os experimentos realizados mostraram que inspecionar os pacotes individualmente tem um desempenho similar ao inspecionar os fluxos durante a detecção de intrusão. Também mostrou que o desempenho dos classificadores sofreu queda à medida que se aumentava o atraso na entrega dos rótulos verdadeiros.

Apesar dos avanços nas pesquisas relacionadas a IDSs, Ribeiro, Paiva e Miani (2020) apontaram a questão de alto custo e esforço praticado pelos profissionais da área de segurança para obter dados rotulados. Com isso, o objetivo do seu trabalho foi avaliar o uso de algoritmos de mineração de fluxos contínuos de dados para a detecção de *botnets*, porém utilizando cenários mais próximos do cotidiano real, como por exemplo:

- os fluxos de dados estão constantemente chegando;
- novos ataques podem surgir e tais ataques não estão presentes no modelo de decisão;
- poucos fluxos são rotulados;
- a avaliação da qualidade do classificador deve ser feita atentando-se para o momento em que os fluxos chegam, em particular aqueles em que novos ataques chegam.

Os resultados de seu trabalho demonstraram que é possível minimizar a quantidade de instâncias rotuladas apresentadas ao classificador, e ainda mantendo um bom desempenho.

Um trabalho bem interessante foi o de Viegas et al. (2018), onde foi implementado o algoritmo denominado de *Big flow* que aplica um método de verificação que checa se o resultado de determinado classificador é válido e provém confiança, isso tudo aplicado a redes de altas velocidades e, consequentemente, grandes pacotes de dados. Experimentos utilizando um conjunto de dados trafegando pela rede, com abrangência de um ano, demostraram que o *Bigflow* mantém sua acurácia ao longo do tempo. A partir dos testes foi comprovado que o *Bigflow* é escalável, lidando com uma rede de 10 Gbps em um *Cluster* de 40 núcleos.

Como já mencionado, os IDS atualmente necessitam estar cada vez mais alinhados com os ataques praticados cotidianamente a grandes instituições do mercado, com

essa premissa, Horchulhack, Viegas e Santin (2022) também propuseram um IDS baseado em aprendizado de fluxo, com atrasos na atualização do modelo. Seu modelo, primeiramente, mantém a acurácia da detecção de intrusão. Seguidamente, as instâncias rejeitadas são armazenadas por longos períodos e utilizadas para atualizações no modelo. Em seus experimentos, o modelo proposto pode manter sua acurácia de classificações por longos períodos sem necessitar de atualizações do modelo, melhorando as taxas de falso positivos em 12% e de rejeição de instâncias em 8% em relação as técnicas de detecção mais tradicionais.

2.8 Considerações Finais

Neste capítulo, foram abordados conceitos importantes para o entendimento deste trabalho e o que ele propõe. Foi introduzido o conceito de segurança da informação e a importância da segurança dos dados. Logo em sequência, foi apresentado o conceito e o funcionamento dos sistemas de detecção de intrusão. Em seguida, foi explanado o conceito de *botnets* e seu funcionamento. Também foram apresentados os conceitos de aprendizado de máquina, além da explicação de como esse pode estar aliado a mineração de fluxo de dados. A partir disso, foi feita a explicação do aprendizado ativo e de como este pode auxiliar em situações com alto número de rótulos. Por fim, foram revistos outros trabalhos que abordam o mesmo tema.

3 Desenvolvimento

Neste capítulo, serão apresentados as ferramentas e os métodos utilizados neste trabalho para realizar os experimentos, além da apresentação e análise dos experimentos em si. Na seção 3.1 será apresentado e explicado o software MOA utilizado para análise da base de dados, cuja qual é explicada na seção 3.2 onde será explicado o conteúdo da base CTU-13, para que assim, partir para a seção 3.3 com os experimentos detalhados, para depois apresentar os resultados na seção 3.4. E com isso, comparar a metodologia dos experimentos, e seus resultados, com outros trabalhos na seção 3.6.

3.1 MOA

O framework (*Massive Online Analysis*) MOA, desenvolvido em Java por Bifet et al. (2010), foi utilizado neste trabalho para executar os experimentos. Uma das vantagens do MOA é que ele concentra e disponibiliza um conjunto de algoritmos de fluxos contínuos de dados para diversos cenários de aprendizado de máquina. O MOA também permite que algoritmos externos desenvolvidos pela comunidade cientifica sejam incorporados ao seu escopo e podem ser executados tanto em sua interface gráfica quando em linha de comando.

Outra vantagem do MOA é que ele permite utilizar o recurso de atraso na rotulação em suas estratégias, juntamente com a possibilidade do classificador utilizar uma janela inicial de treinamento. Desta forma, é possível comparar os diversos efeitos destes parâmetros nas bases de dados analisadas, e desta forma, concluir qual possui um efeito maior nos classificadores.

3.2 CTU-13

O conjunto de dados CTU-13 (GARCíA et al., 2014) consiste em conjunto de tráfego de *botnets* disponibilizado abertamente para a comunidade cientifica e que possui tráfego real de botnets combinados com tráfego normal, além do *background*. O CTU-13 apresenta treze cenários reais baseados em ataques de *botnets*, contendo diferentes tipos de protocolos de rede e *malwares*, sendo que cada cenário possui um número diferente de instâncias pertencentes a cada classe do problema. Os rótulos de instância são divididos em: Normal (o tráfego corresponde a determinados filtros relacionados a computadores de rede conhecidos), *Botnet* (tráfego originado ou direcionado para endereços IP infectados conhecidos) e *Background* (tráfego entre os outros endereços IP de computadores que não são conhecidos nem infectados) (GARCíA et al., 2014). As Tabelas 2 e 3 apre-

sentam a quantidade de dados e a distribuição de classes em cada cenário do CTU-13, respectivamente.

Id	**Duração (h)**	**# Pacotes**	**# Fluxos**	**Tamanho**	**Tipo de *botnet***	***# Bots***
1	6,15	71.971.482	2.824.637	52 GB	Neris	1
2	4,21	71.851.300	1.808.123	60 GB	Neris	1
3	66,85	167.730.395	4.710.639	121 GB	Rbot	1
4	4,21	62.089.135	1.121.077	53 GB	Rbot	1
5	11,63	4.481.167	129.833	37,6 GB	Virut	1
6	2,18	38.764.357	558.920	30 GB	Menti	1
7	0,38	7.467.739	144.078	5,8 GB	Sogou	1
8	19,5	155.207.799	2.954.231	123 GB	Murlo	1
9	5,18	115.415.321	2.753.885	94 GB	Neris	10
10	4,75	90.389.782	1.309.792	73 GB	Rbot	10
11	0,26	6.337.202	107.252	5,2 GB	Rbot	3
12	1,21	13.212.268	325.472	8,3 GB	NSIS.ay	3
13	16,36	50.888.256	1.925.150	34 GB	Virut	1

Tabela 2 – Quantidade de dados em cada cenário do CTU-13

Id	***Background***	***Botnet***	**Normal**
1	10.124.854 (95,40%)	94.972 (0,89%)	392.433 (3,69%)
2	6.071.419 (95,59%)	54.433 (0,85%)	225.336 (3,54%)
3	14.381.899 (94,60%)	75.891 (0,49%)	744.270 (4,89%)
4	3.895.469 (91,91%)	6466 (0,15%)	336.103 (7,93%)
5	419.267 (91,37%)	2129 (0,46%)	37.144 (8,15%)
6	2.031.967 (94,12%)	4927 (0,22%)	121.854 (5,64%)
7	425.611 (93,47%)	293 (0,06%)	28.270 (6,22%)
8	11.451.205 (95,47%)	12.063 (0,10%)	530.666 (4,42%)
9	6.881.228 (90,22%)	383.441 (6,24%)	321.917 (6,21%)
10	4.535.493 (87,54%)	323.441 (6,24%)	321.917 (6,21%)
11	119.933 (29,33%)	277.892 (67,97%)	11.010 (2,69%)
12	119.933 (29,33%)	277.892 (67,97%)	11.010 (2,69%)
13	1.218.140 (93,76%)	21.760 (1,67%)	59.190 (4,55%)

Tabela 3 – Distribuição de classes em cada cenário do CTU-13

Vale ressaltar que foram analisados os conjuntos de dados pós-processados de Ribeiro (2020), para que fosse possível utilizar o *framework* MOA nas análises. Os dados de cada conjunto do CTU foram analisados, com exceção do conjunto 7; pois o mesmo não possui uma quantidade de fluxos suficientes para a análise.

3.3 Métricas avaliadoras de desempenho

De acordo com Ribeiro, Paiva e Miani (2020), o objetivo dos avaliadores de desempenho preditivo é calcular medidas que sejam capazes de proporcionar a comparação entre

diferentes classificadores. É importante destacar que nem todas as métricas utilizadas em aprendizado de máquina são interessantes quando se trabalha com IDSs. Ribeiro, Paiva e Miani (2020) ainda destaca, que isso acontece pois o tráfego de rede é naturalmente muito desbalanceado. Por essas razões, é importante ressaltar que nem todas as métricas comumente utilizadas em aprendizado de máquina possuem uma aplicação, ao trabalhar com IDSs.

Normalmente, para tarefas de classificação binária, é bastante utilizado a matriz de confusão para o cálculo dos avaliadores de desempenho. As linhas da matriz indicam a classe prevista enquanto que as colunas mostram a classe real da instância. Portanto, quando uma instância é executada, uma das células da matriz de confusão é atualizada. No caso dos problemas de classificação de intrusão, as instâncias rotuladas como Ataque são interpretadas como instâncias positivas enquanto que, instâncias rotuladas como normal, são interpretadas como negativas.

- Verdadeiro Positivo (VP), quando o modelo executa corretamente a classificação de um pacote de ataque;
- Verdadeiro Negativo (VN), quando o modelo executa corretamente a classificação de um pacote normal;
- Falso Positivo (FP), quando o modelo classifica erroneamente uma instância como um ataque, quando na verdade é um pacote normal;
- Falso Negativo (FN), quando o modelo classifica erroneamente uma instância como um pacote normal, quando na verdade é um ataque.

As medidas mais utilizados para IDSs são: revocação, precisão e taxa de alarmes falsos, do inglês *False Alarm Rate* (FAR).

A revocação, apresentada na Equação 1, mede o quão bem o classificador identificou comportamentos anormais como ataques.

$$REV = \frac{VP}{VP + FN} \tag{3.1}$$

Enquanto que a precisão, dada Equação 2, verifica o quanto de instâncias de Ataque foram classificadas corretamente dentro daquelas que foram classificadas como Ataque.

$$PREC = \frac{VP}{VP + FP} \tag{3.2}$$

Por fim, a Taxa de Alarme Falso (FAR), Apresentada na Equação 3, indica a proporção das observações de instâncias do tipo Normal incorretamente sinalizadas como Ataque.

$$FAR = \frac{FP}{FP + VN} \tag{3.3}$$

3.4 Experimentos

Nesta subseção, serão detalhados os experimentos realizados neste trabalho. Foi utilizado o algoritmo *OzaBagASHT*, com parâmetros *default*, por ter sido o classificador com melhor desempenho no trabalho de Olimpio et al. (2021). Os experimentos são divididos nos seguintes grupos:

- Experimento 1: Classificação de fluxos maliciosos usando o algoritmo *OzaBagASHT* com a estratégia de atraso na rotulação. Testes realizados com os parâmetros de atraso iguais a 10.000, 50.000 e 100.000 instâncias. Desta forma, é possível simular eventos onde o classificador toma um determinado período de tempo para receber os rótulos e classificar os fluxos. Algo que possa simular um atraso na transmissão de pacotes de uma rede local. Este experimento foi feito com base ao que foi realizado por (OLIMPIO et al., 2021), detalhado na seção 5.5.2;
- Experimento 2: Classificação de fluxos maliciosos usando o algoritmo *OzaBagASHT* com a estratégia de porcentagem de rotulação. Testes realizados com parâmetros de porcentagem iguais a 1%, 5%, 10% e 50%. Com isto, é possível simular eventos onde o classificador não possui todos rótulos necessários para analisar os fluxos de dados. Portanto, verifica-se que os resultados do classificador quando este possui uma quantidade de dados limitada para trabalhar. Este experimento foi feito com base ao que foi realizado na seção 5.5.3 do trabalho de (OLIMPIO et al., 2021);
- Experimento 3: Por fim, o terceiro teste consiste em utilizar uma estratégia composta pela união dos parâmetros das estratégias 1 e 2, ou seja, a classificação de fluxos maliciosos usando atraso na entrega de rótulos e aprendizado ativo. Logo, o classificador é testado de uma forma que simule melhor um cenário mais próximo do real. Da mesma forma que foi feita por (OLIMPIO et al., 2021) na seção 5.5.3.1 de seu trabalho;

3.4.1 Validação de Experimentos

O experimento 1 irá Avaliar o algoritmo de classificação em um cenário mais próximo do real, onde há um atraso para se obter as instâncias rotuladas, e assim, atualizar o modelo. Portanto será avaliado o impacto no desempenho do classificador quando há atraso na entrega dos rótulos. A Figura 2 ilustra a rotulagem aplicada a um fluxo contínuo de dados. Assume-se que as instâncias chegam continuamente e o classificador executa a predição para cada instância de tempo.

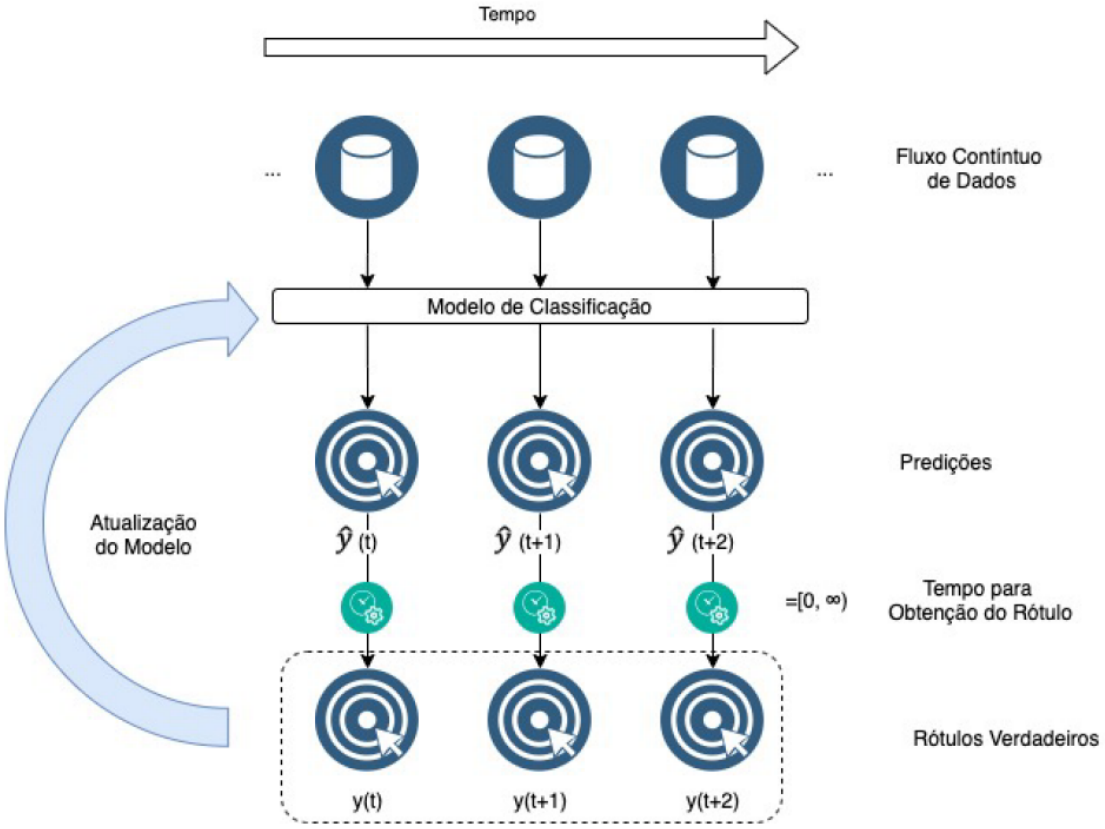

Figura 2 – Esquema de obtenção de rótulos em fluxo contínuo de dados (OLIMPIO et al., 2021).

No cenário de atraso dos rótulos, o classificador não receberá o rótulo imediatamente após a classificação. O rótulo verdadeiro não será entregue ao classificador até o tempo máximo, previamente configurado, ser atingido. Após isso, o rótulo da instância é enviado ao classificador para realizar o treinamento do modelo, e assim ter sua atualização.

Para o experimento 2, foi aplicado o aprendizado ativo em fluxos maliciosos utilizando estratégia aleatória. Uma hipótese proposta por se aproximar de um ambiente real, onde não é possível o especialista rotular todas as instâncias do fluxo de dados. Nessa estratégia, os rótulos a serem entregues para o classificador são selecionados aleatoriamente, porém sempre obedecendo o limite de percentual previamente configurado.

Para o cenário de fluxos contínuos de dados, onde novos dados chegam de modo contínuo e o modelo está sendo constantemente atualizado, foi utilizado a técnica de sequenciamento preditivo, que consiste em utilizar cada exemplo individualmente para teste antes de fazer o treinamento do modelo. A Figura 3 ilustra o funcionamento do aprendizado ativo para a seleção de rótulos para a atualização do modelo de classificação.

Após a predição feita pelo classificador, antes dos rótulos verdadeiros serem enviados para atualização, é feita uma seleção aleatória destes rótulos a partir do percentual previamente estabelecido. Para cada rótulo selecionado, o percentual é decrementado.

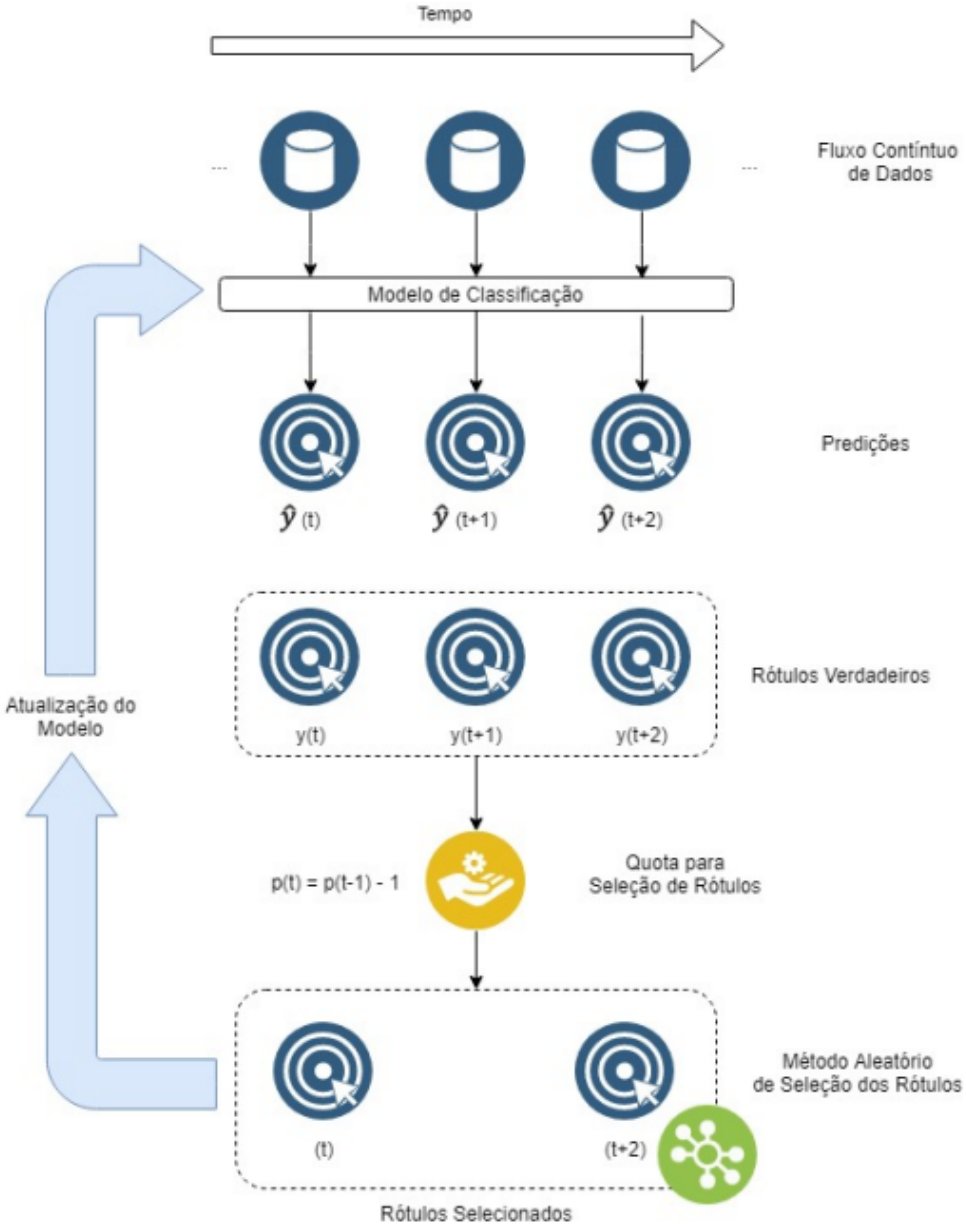

Figura 3 – Representação do aprendizado ativo. (OLIMPIO et al., 2021)

Após o rótulo ser selecionado, ele é por fim enviado ao classificador para executar a atualização do modelo.

Importante ressaltar que os diferentes grupos na base CTU-13, descritos da subseção 3.2, são desbalanceados, ou seja, os grupos tentam espelhar da melhor forma o tráfego de rede presente em ambientes reais de produção onde somente uma pequena parcela do tráfego é realmente malicioso. Ao final de cada teste, foram geradas as matrizes de confusão que foram posteriormente utilizadas para realizar os cálculos de precisão, revocação e FAR de cada conjunto do CTU-13. Para fins comparativos, foram calculados e inseridos os valores de *Baseline*, que representa o modelo de classificação treinado com o mesmo algoritmo porém sem nenhum atraso na rotulação ou método de seleção de instâncias

rotuladas.

3.5 Resultados

CTU	Atraso	Revocação(%)	Precisão(%)	FAR
Baseline		88,97	97,64	0,02
	10.000	74,13	95,98	0,03
1	50.000	71,34	88,47	0,04
	100.000	58,05	95,02	0,03
Baseline		78,71	94,40	0,06
	10.000	55,20	91,51	0,07
2	50.000	37,36	87,35	0,07
	100.000	34,76	87,15	0,07
Baseline		98,59	99,99	0,00
	10.000	80,95	100,00	0,00
3	50.000	53,93	99,96	0,00
	100.000	35,16	100,00	0,00
Baseline		92,38	97,22	0,00
	10.000	86,02	99,34	0,13
4	50.000	63,52	94,98	0,00
	100.000	67,16	91,34	0,01
Baseline		75,00	91,82	0,05
	10.000	33,23	82,99	0,05
5	50.000	1,67	10,61	0,10
	100.000	0,51	3,52	0,00
Baseline		95,35	99,96	0,00
	10.000	79,34	100,00	0,00
6	50.000	59,59	100,00	0,00
	100.000	58,20	100,00	0,00
Baseline		90,94	95,41	0,24
	10.000	60,46	98,48	0,00
8	50.000	46,77	97,04	0,00
	100.000	50,16	99,35	0,00
Baseline		90,94	95,41	0,24
	10.000	88,40	94,12	0,30
9	50.000	86,55	92,97	0,36
	100.000	83,45	93,98	0,29
Baseline		99,78	99,99	0,00
	10.000	97,71	100,00	0,13
10	50.000	87,70	99,98	0,00
	100.000	87,66	99,98	0,00
Baseline		99,31	99,93	0,01
	10.000	72,85	99,97	0,00
11	50.000	0,00	0,00	0,00
	100.000	0,00	0,00	0,00
Baseline		57,03	92,34	0,04
	10.000	16,51	75,77	0,04
12	50.000	3,52	99,21	0,00
	100.000	0,35	89,29	0,00
Baseline		89,54	93,44	0,15
	10.000	57,99	88,47	0,18
13	50.000	41,19	80,89	0,23
	100.000	36,21	75,90	0,27

Tabela 4 – Resultados de classificação do CTU-13 utilizando atraso na rotulação dos fluxos

Os resultados presentes na Tabela 4, referentes ao primeiro experimento, mostram que em todos os conjuntos do CTU, os valores de Revocação foram menores que o *baseline*. Isto demonstra que o atraso no fornecimento dos rótulos influencia na capacidade do algoritmo de classificar fluxos como Ataque de forma correta. Em destaque para os

conjuntos 5 e 11 do CTU onde os valores de precisão também foram menores que o *baseline*. Isso pode estar relacionado a menor quantidade fluxos presentes nestes conjuntos em relação aos demais, vide 2. Já que os conjuntos 9 e 10 mantiveram valores bem próximos ao *baseline*, sendo os conjuntos com maior número de *botnets*, como também mostrado na Tabela 2.

A Tabela 5 apresenta os resultados referentes ao experimento 2, de classificação do CTU-13 utilizando a técnica de Aprendizado ativo, juntamente com a alteração do percentual de rotulagem. A ideia principal deste teste foi de verificar se uma limitação no percentual de rotulagem iria alterar drasticamente os resultados.

De acordo com a Tabela 5 os valores de revocação e precisão ficam abaixo do *baseline* na maioria dos conjuntos do CTU-13. Percebe-se que os valores se aproximam do *baseline* conforme o percentual de rotulação é aumentado. Comportamento ainda mais visível no conjunto 5 no qual provavelmente seu baixo número de fluxos, aliada ao baixo percentual de rotulagem, deixou o classificador com um número bem limitado de dados para processar. Assim como o teste 1, os conjuntos 9 e 10 do CTU obtiveram valores de precisão e revocação próximos do *baseline*. Algo que pode até então indicar que um maior número de *botnets* seja mais perceptível para o classificador.

Para concluir, foi realizado o 3º teste de combinação da estratégia de atraso de rotulagem com a estratégia de percentual de rotulagem das amostras. O intuito deste teste é avaliar a desempenho do classificador, além de expô-lo a circunstâncias diversas. Com isto, é emulado um ambiente mais realista. Os resultados seguem nas Tabelas 6, 7 e 8.

A partir de uma análise das métricas presentes nas Tabelas 6, 7 e 8, pode-se perceber o efeito que ambas as estratégias possuem na desempenho do classificador, principalmente nos conjuntos 5, 11 e 12 do CTU. Com uma atenção maior para o conjunto 5, onde houve praticamente uma falha total do classificador de diferenciar os fluxos normais dos fluxos com dados maliciosos, valores exibidos como N/A representam que o classificador não conseguiu retornar a métrica pois o denominador da equação equivalia a zero. Ou seja, valores de Verdadeiros positivos e Falsos Positivos iguais a zero. O que deve estar relacionado, novamente, com o menor número de fluxos presentes nestes conjuntos. Ao ser comparado com os resultados das Tabelas 4 e 5, é notável que o percentual de rotulagem tem um efeito maior no classificador quando adicionado o atraso no fornecimento dos rótulos, principalmente nos conjuntos 2 e 13 que apresentaram uma maior discrepância em relação aos testes 1 e 2. Enquanto que nos conjuntos 9 e 10, que possuem um maior número de *botnets*, os valores das métricas foram bem próximos do *baseline*.

CTU	Percentual de Rotulagem	Revocação(%)	Precisão(%)	FAR
Baseline		90,42	98,17	0,02
1	1%	71,57	88,90	0,13
	5%	77,16	94,63	0,06
	10%	85,33	96,63	0,04
	50%	89,97	98,15	0,03
Baseline		76,60	96,81	0,03
2	1%	51,18	55,05	0,49
	5%	64,30	79,81	0,19
	10%	57,51	78,86	0,18
	50%	71,29	94,12	0,05
Baseline		98,34	99,99	0,00
3	1%	93,60	99,96	0,00
	5%	97,08	99,90	0,00
	10%	97,69	99,99	0,00
	50%	98,78	99,98	0,00
Baseline		94,57	99,92	0,00
4	1%	48,84	95,96	0,00
	5%	63,84	99,34	0,00
	10%	83,22	96,02	0,01
	50%	94,65	97,10	0,01
Baseline		78,69	90,43	0,06
5	1%	25,31	68,06	0,08
	5%	30,85	51,29	0,20
	10%	73,14	80,27	0,13
	50%	83,57	83,57	0,11
Baseline		96,03	99,96	0,00
6	1%	86,91	98,22	0,01
	5%	93,11	99,95	0,00
	10%	94,88	99,91	0,00
	50%	95,90	95,10	0,04
Baseline		87,02	99,64	0,00
8	1%	61,76	100,00	0,00
	5%	41,64	93,61	0,01
	10%	57,94	80,88	0,03
	50%	91,51	99,54	0,00
Baseline		91,33	95,92	0,38
9	1%	93,18	81,80	2,02
	5%	91,58	89,95	0,99
	10%	90,11	93,17	0,64
	50%	91,49	94,89	0,48
Baseline		99,78	100,00	0,00
10	1%	98,65	99,90	0,01
	5%	99,42	99,91	0,01
	10%	99,60	99,95	0,00
	50%	99,78	99,95	0,00
Baseline		99,51	99,93	0,01
11	1%	91,33	100,00	0,00
	5%	98,46	99,95	0,00
	10%	99,60	99,95	0,00
	50%	99,64	99,94	0,01
Baseline		48,89	93,15	0,02
12	1%	0,46	10,99	0,03
	5%	5,67	80,39	0,01
	10%	99,60	99,95	0,00
	50%	42,02	91,93	0,02
Baseline		89,41	94,68	0,11
13	1%	61,33	82,06	0,28
	5%	85,41	74,80	0,61
	10%	87,77	79,73	0,47
	50%	91,90	93,91	0,13

Tabela 5 – Resultados de classificação do CTU-13 utilizando a técnica de aprendizado ativo, juntamente com a alteração do percentual de rotulagem

3.6 Comparação com a literatura

Nesta subseção, serão analisados trabalhos com temas relacionados a análise de fluxo e aprendizado ativo de dados. A ideia é comparar as conclusões de seus experimentos e resultados com os deste trabalho, para que assim, possa se ter uma melhor noção de padrões dos classificadores, assim como comparar técnicas de classificação.

CTU	Percentual de Rotulagem	Atraso	Revocação	Precisão	FAR
Baseline			93,34	97,68	0,03
1	1%	10.000	70,22	86,63	0,16
		50.000	63,74	82,34	0,20
		100.000	59,18	80,87	0,21
	5%	10.000	71,41	94,19	0,06
		50.000	81,63	73,96	0,61
		100.000	61,28	93,69	0,06
	10%	10.000	82,12	96,07	0,05
		50.000	75,10	95,67	0,05
		100.000	46,65	68,14	0,46
	50%	10.000	81,83	96,99	0,04
		50.000	73,17	96,84	0,04
		100.000	66,84	97,71	0,02
Baseline			78,95	96,11	0,04
2	1%	10.000	41,64	49,90	0,49
		50.000	18,95	31,32	0,49
		100.000	18,70	31,54	0,48
	5%	10.000	52,15	75,61	0,20
		50.000	52,15	75,61	0,20
		100.000	52,15	75,61	0,20
	10%	10.000	43,98	74,53	0,18
		50.000	15,23	47,99	0,19
		100.000	12,54	43,11	0,19
	50%	10.000	55,26	91,32	0,06
		50.000	55,26	91,32	0,06
		100.000	29,69	79,96	0,09
Baseline			99,16	100,00	0,00
3	1%	10.000	0,49	99,96	0,00
		50.000	83,00	99,96	0,00
		100.000	79,62	99,97	0,00
	5%	10.000	93,06	99,88	6,94
		50.000	93,06	99,88	0,00
		100.000	72,75	99,87	0,00
	10%	10.000	91,06	99,98	0,00
		50.000	69,77	99,98	0,00
		100.000	66,14	99,98	0,00
	50%	10.000	81,83	96,99	0,04
		50.000	89,28	99,96	0,00
		100.000	68,65	99,96	0,00
Baseline			79,36	90,62	0,06
4	1%	10.000	35,81	94,19	0,01
		50.000	17,52	91,13	0,00
		100.000	1,16	71,43	0,00
	5%	10.000	55,62	98,97	0,00
		50.000	55,62	98,97	0,00
		100.000	6,43	94,86	0,00
	10%	10.000	71,20	95,28	0,01
		50.000	0,00	0,00	0,08
		100.000	9,57	76,23	0,01
	50%	10.000	82,21	96,94	0,01
		50.000	82,21	96,94	0,01
		100.00	13,10	84,29	0,01

Tabela 6 – Resultados de classificação do CTU-13 utilizando a combinação de estratégias de percentual de rotulagem com o atraso de rotulagem

Este trabalho utilizou as mesmas métricas trabalhadas por Olimpio et al. (2021), cujo seu trabalho teve o intuito de verificar se o atraso nos rótulos de instância em classificadores de fluxo contínuo de dados levaria em algum impacto nas análises, da mesma forma, também foi analisado se, a partir da técnica de aprendizado ativo, o desempenho do classificador iria se manter estável ao forçar o classificador a somente rotular um subconjunto de instâncias.

Em seus resultados, Olimpio et al. (2021) chegou a mesma conclusão de que a entrega atrasada dos rótulos para atualização do modelo de decisão, conforme esperado, causou uma queda no desempenho preditivo dos classificadores testados. Além de que

CTU	Percentual de Rotulagem	Atraso	Revocação	Precisão	FAR
Baseline			97,34	99,96	0,00
5	1%	10.000	16,32	50,00	0,11
		50.000	0,00	N/A	100,00
		100.000	0,00	N/A	0,00
	5%	10.000	10,43	28,40	0,19
		50.000	10,43	28,40	0,19
		100.000	0,00	N/A	0,00
	10%	10.000	40,51	53,44	0,25
		50.000	83,39	99,23	0,01
		100.000	0,00	N/A	0,00
	50%	10.000	19,76	52,82	0,12
		50.000	19,76	52,82	0,12
		100.000	0,00	N/A	0,00
Baseline			95,12	99,71	0,00
6	1%	10.000	82,70	94,75	0,04
		50.000	77,32	98,16	0,01
		100.000	67,56	97,90	0,01
	5%	10.000	89,18	99,95	0,00
		50.000	89,18	99,95	0,00
		100.000	72,42	100,00	0,00
	10%	10.000	91,17	99,95	0,00
		50.000	83,39	99,23	0,01
		100.000	72,76	100,00	0,00
	50%	10.000	87,15	93,19	0,05
		50.000	87,15	93,19	0,05
		100.000	71,17	95,37	0,03
Baseline			95,12	99,71	0,00
8	1%	10.000	59,91	100,00	0,00
		50.000	58,29	100,00	0,00
		100.000	59,66	100,00	0,00
	5%	10.000	11,02	76,27	0,01
		50.000	11,02	76,27	0,01
		100.000	25,14	90,01	0,01
	10%	10.000	37,48	73,43	0,03
		50.000	39,62	75,23	0,03
		100.000	41,58	78,95	0,02
	50%	10.000	68,01	98,53	0,00
		50.000	68,01	98,53	0,00
		100.000	69,36	99,23	0,00
Baseline			95,64	94,10	0,58
9	1%	10.000	92,97	81,93	2,00
		50.000	91,82	82,35	1,91
		100.000	90,29	82,82	1,82
	5%	10.000	89,05	92,77	0,68
		50.000	87,09	93,05	0,63
		100.000	84,57	92,98	0,62
	10%	10.000	89,05	92,77	0,68
		50.000	87,09	93,05	0,63
		100.000	84,57	92,98	0,62
	50%	10.000	88,96	93,77	0,57
		50.000	88,96	93,77	0,57
		100.000	83,92	93,25	0,59

Tabela 7 – Resultados de classificação do CTU-13 utilizando a combinação de estratégias de percentual de rotulagem com o atraso de rotulagem. Continuação da Tabela 6

rotular somente uma parte das instâncias causa um impacto muito menor do que atrasar os mesmos.

O trabalho de Ribeiro (2020) envolveu a realização de testes com diversos algoritmos de classificação, porém não só em classificações binárias mas também em multi-classe, juntamente com a estratégia de aprendizado ativo. Seus resultados mostram que é possível diminuir a quantidade de instâncias rotuladas e manter o bom desempenho de detecção de fluxos maliciosos. Além disso, Ribeiro (2020) mostrou que a avaliação multi-classe demonstra quais classes são decisivas no desempenho do classificador. Sua pluralidade de parâmetros são motivacionais para a realização de um novo trabalho envolvendo o

CTU	Percentual de Rotulagem	Atraso	Revocação	Precisão	FAR
Baseline			99,83	99,99	0,00
10	1%	10.000	10,89	10,86	0,00
		50.000	57,76	99,83	0,01
		100.000	29,47	99,79	0,01
	5%	10.000	93,02	99,93	0,01
		50.000	90,86	89,71	1,01
		100.000	29,52	99,80	0,01
	10%	10.000	93,57	99,97	0,00
		50.000	60,47	99,97	0,00
		100.000	29,50	99,89	0,00
	50%	10.000	88,96	93,77	0,57
		50.000	96,20	99,94	0,01
		100.000	29,53	99,69	0,01
Baseline			99,52	99,93	0,01
11	1%	10.000	26,74	100,00	0,00
		50.000	0,00	N/A	0,00
		100.000	0,00	N/A	0,00
	5%	10.000	36,46	99,83	0,01
		50.000	93,02	99,93	0,01
		100.000	0,00	N/A	0,00
	10%	10.000	38,30	99,43	0,02
		50.000	0,00	N/A	0,00
		100.000	0,00	N/A	0,00
	50%	10.000	45,19	99,92	0,00
		50.000	96,20	99,94	0,01
		100.000	0,00	N/A	0,00
Baseline			50,65	92,42	0,03
12	1%	10.000	0,05	1,25	0,02
		50.000	0,00	0,00	0,02
		100.000	0,00	0,00	0,03
	5%	10.000	0,60	27,08	0,01
		50.000	36,46	99,83	0,01
		100.000	0,00	0,00	0,02
	10%	10.000	26,34	52,15	0,16
		50.000	1,94	8,57	0,14
		100.000	1,61	9,75	0,10
	50%	10.000	18,17	84,19	0,02
		50.000	5,21	59,16	0,02
		100.000	1,34	27,88	0,02
Baseline			94,19	93,53	0,14
13	1%	10.000	57,55	79,27	0,32
		50.000	48,09	76,25	0,32
		100.000	36,91	73,01	0,29
	5%	10.000	81,63	73,96	0,61
		50.000	81,63	73,96	0,61
		100.000	52,47	65,95	0,58
	10%	10.000	84,13	78,26	0,50
		50.000	65,08	74,14	0,48
		100.000	46,65	68,14	0,46
	50%	10.000	74,80	91,69	0,14
		50.000	48,89	87,77	0,14
		100.000	37,96	86,42	0,13

Tabela 8 – Resultados de classificação do CTU-13 utilizando a combinação de estratégias de percentual de rotulagem com o atraso de rotulagem. Continuação da Tabela 7

CTU-13.

Comparando os resultados obtidos neste trabalho com os da seção 3.5 do trabalho de Olimpio et al. (2021), é possível verificar um padrão nos valores de revocação e precisão ao aplicar as técnicas de aprendizado ativo e atraso na rotulação. Os valores das métricas de revocação e precisão diminuem a partir do aumento do atraso dos rótulos, enquanto que os valores de FAR aumentam. O mesmo ocorre para o aumento no percentual de rotulagem em aprendizado ativo. O que mostra uma piora no desempenho do classificador em ambos os trabalhos. Principalmente ao comparar os resultados das Tabelas 5, 6 e 7; do experimento 3, com os resultados obtidos por (OLIMPIO et al., 2021) ao aplicar as

duas estratégias em um único teste.

4 Conclusões

Neste trabalho, foi possível analisar os resultados dos testes de classificação e de aprendizado ativo do conjunto CTU-13, alinhado com as estratégias de atraso na rotulação e limitação no percentual da rotulação, respectivamente. Utilizando o *framework* MOA, foi possível configurar as estratégias com diferentes parâmetros e realizar os testes em 12 grupos do conjunto CTU-13. A partir dos testes realizados, foi concluído que o atraso na rótulos dos fluxos afetou o desempenho do classificador, o que pode ser visto nas métricas de precisão, revocação principalmente e FAR. Algo que já era esperado, porém os testes 1, 2 e 3 também mostraram que mesmo em situações adversas, o classificador manteve um desempenho consideravelmente bom. Principalmente nos conjuntos que possuem um maior número de pacotes, como por exemplo o conjunto 1 (CTU-1). Mesmo com um percentual de 10% e um atraso de 10.000 instâncias, apresentou valores de 82% e 96% de revocação e precisão, respectivamente. Valores bem próximos do seu *baseline*.

Nos testes onde foram combinados as estratégias de atraso de rotulação, alteração na porcentagem dos rótulos fornecidos, foi possível averiguar como condições adversas podem influenciar no desempenho do modelo, podendo torná-lo totalmente incapaz de distinguir os dados fornecidos a ele, como foi averiguado nas Tabelas 7 e 8. Uma possível causa dos baixos valores de precisão do modelo nos conjuntos 05 e 11 seria o fato de serem conjuntos menores que os demais, como foi detalhado na Tabela 1. Pode-se ver também que o classificador manteve bons níveis de revocação e precisão nos conjuntos 09 e 10, que são os conjuntos com maior número de *botnets* presentes nos fluxos. Com destaque no conjunto 9 da Tabela 7, onde que com somente 1% dos rótulos, manteve valores de precisão bem próximos do *baseline*, com menos de 10% de diferença.

Portanto, os resultados mostram que o classificador *OzabagASHT* mantém o seu desempenho na classificação, ainda que submetido a atrasos na rotulação dos fluxos presentes na análise, além da restrição da porcentagem dos rótulos fornecidos. Relevando também de que os cenários do tipo *baseline*, tratados neste trabalho, são cenários ideias e pouco prováveis de ocorrer em sistemas rais rodando em produção. Pois não levam em conta diversos fatores, como o atraso de rotulação e indisponibilidade dos rótulos, que foram discutidos neste trabalho.

Para trabalhos futuros, há a possibilidade de realizar os testes com outros algoritmos e compará-los ao *OzabagASHT*. Outra proposta seria aplicar as estratégias utilizadas neste trabalho em outros conjuntos de dados, para averiguar uma diversidade maior de dados maliciosos. Um proposta mais prática seria estudar opções de como colocar o estudo em um ambiente em produção, e assim simular um real ataque a uma rede, em um

ambiente controlado, para analisar o real funcionamento do classificador.

Para a realização deste trabalho, utilizei dos conhecimentos adquiridos nas disciplinas de Arquitetura de Redes de Computadores (FACOM49070) para entender o funcionamento de uma rede local, Algoritmos e Programação de computadores (FACOM49010) na qual, foi aprendida a base do funcionamento de algoritmos. Por fim, também foram utilizados os conhecimentos obtidos na disciplina de Estatística (FAMAT49021), para melhor análisa das métricas utilizadas.

Referências

ACCENTURE. State of cybersecurity resilience 2021, how aligning security and the business creates cyber resilience. 2022. Disponível em: <https://www.accenture.com/_acnmedia/PDF-165/Accenture-State-Of-Cybersecurity-2021.pdf>. Acesso em: 10/03/2022. Citado na página 8.

ALBERT, B.; BERNHARD, P.; GEOFF, H. Active learning with evolving streaming data. p. 597–612, 2011. Disponível em: <https://doi.org/10.1007/978-3-642-23808-6_39>. Citado 2 vezes nas páginas 9 e 17.

BEZERRA et al. Providing iot host-based datasets for intrusion detection research. in: Anais principais do xviii simpósio brasileiro em segurança da informação e de sistemas computacionais. SBC, p. 15028, 2018. Disponível em: <https://sol.sbc.org.br/index.php/sbseg/article/view/4240>. Acesso em: 10/03/2022. Citado na página 8.

BHUYAN, M. H.; BHATTACHARYYA, D. K.; KALITA, J. K. Network traffic anomaly detection techniques and systems. Springer, p. 115–169, 2017. Citado 2 vezes nas páginas 8 e 13.

BIFET, A. et al. Moa: Massive online analysis, a framework for stream classification and clustering. *Journal of Machine Learning Research (JMLR) Workshop and Conference Proceedings.*, v. 11, 2010. Citado na página 20.

BURKOV, A. *The Hundred-Page Machine Learning Book*. [S.l.: s.n.], 2019. Citado na página 15.

CARVALHO, A. et al. Inteligência artificial–uma abordagem de aprendizado de máquina. *Rio de Janeiro: LTC*, p. 45, 2011. Citado na página 15.

CERON, J. M. Arquitetura distribuída e automatizada para mitigação de botnet baseada em análise dinâmica de malwares. 2010. Disponível em: <http://www.lume.ufrgs.br/handle/10183/70238>. Citado na página 14.

COSTA, V. et al. Online detection of botnets on network flows using stream mining. SBC, Porto Alegre, RS, Brasil, p. 225–238, 2018. ISSN 2177-9384. Disponível em: <https://sol.sbc.org.br/index.php/sbrc/article/view/2418>. Citado na página 9.

DOMINGOS, P.; HULTEN, G. Mining high-speed data streams. In: *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2000. p. 71–80. Citado na página 16.

EMBRAPA. Empresa brasileira de pesquisa agropecuária. 2017. Citado na página 15.

GAMA, J. Knowledge discovery from data streams. CRC Press, 2010. Citado na página 16.

GAONKAR, S. et al. A survey on botnet detection techniques. p. 1–6, 2020. Citado na página 8.

GARCIA, S. et al. An empirical comparison of botnet detection methods. *computers & security*, Elsevier, v. 45, p. 100–123, 2014. Citado na página 15.

GARCíA, S. et al. An empirical comparison of botnet detection methods. *Computers & Security*, v. 45, p. 100–123, 2014. ISSN 0167-4048. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167404814000923>. Citado na página 20.

GARCíA-TEODORO, P. et al. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, v. 28, n. 1, p. 18–28, 2009. ISSN 0167-4048. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167404808000692>. Citado na página 13.

GOLLMANN, D. Computer security. *Wiley Interdisciplinary Reviews: Computational Statistics*, Wiley Online Library, v. 2, n. 5, p. 544–554, 2010. Disponível em: <https://doi.org/10.1002/wics.10>. Citado na página 11.

GROSSI, V.; TURINI, F. Stream mining: a novel architecture for ensemble-based classification. *Knowledge and information systems*, Springer, v. 30, n. 2, p. 247–281, 2012. Disponível em: <https://doi.org/10.1007/s10115-011-0378-4>. Citado na página 9.

HORCHULHACK; VIEGAS; SANTIN. Toward feasible machine learning model updates in network-based intrusion detection. Elsevier, 2022. Citado na página 19.

KARIM, A. et al. Botnet detection techniques: review, future trends, and issues. *Journal of Zhejiang University SCIENCE C*, Springer, v. 15, n. 11, p. 943–983, 2014. Disponível em: <https://doi.org/10.1631/jzus.C1300242>. Citado na página 15.

KHATTAK, S. et al. A taxonomy of botnet behavior, detection, and defense. *IEEE Communications Surveys Tutorials*, v. 16, n. 2, p. 898–924, 2014. Citado na página 15.

KRAWCZYK, B. et al. Ensemble learning for data stream analysis: A survey. *Information Fusion*, v. 37, p. 132–156, 2017. ISSN 1566-2535. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1566253516302329>. Citado 2 vezes nas páginas 15 e 16.

LI et al. An efficient intrusion detection system based on support vector machines and gradually feature removal method. *Expert Systems with Applications*, Elsevier, v. 39, p. 424–430, 2012. Citado na página 8.

MAHONEY, M. V.; CHAN, P. K. Phad: Packet header anomaly detection for identifying hostile network traffic. 2001. Disponível em: <http://hdl.handle.net/11141/94>. Citado na página 13.

MIRKOVIC, J.; REIHER, P. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, ACM New York, NY, USA, v. 34, n. 2, p. 39–53, 2004. Citado na página 14.

MITCHELL, T. M. *Machine Learning*. [S.l.: s.n.], 1997. Citado na página 15.

NOGUEIRA, M. Anticipating moves to prevent botnet generated ddos flooding attacks. *arXiv preprint arXiv:1611.09983*, 2016. Citado 2 vezes nas páginas 13 e 14.

OLIMPIO, G. et al. Intrusion detection over network packets using data stream classification algorithms. In: IEEE. *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*. [S.l.], 2021. p. 985–990. Citado 6 vezes nas páginas 10, 17, 18, 23, 29 e 31.

RIBEIRO. Detecção de botnets utilizando classificação de fluxos contínuos de dados. 2020. Citado 2 vezes nas páginas 21 e 30.

RIBEIRO, G. H.; PAIVA, E. R. de F.; MIANI, R. S. A comparison of stream mining algorithms on botnet detection. In: *Proceedings of the 15th International Conference on Availability, Reliability and Security*. [s.n.], 2020. p. 1–10. Disponível em: <https://doi.org/10.1145/3407023.3407053>. Citado 4 vezes nas páginas 9, 18, 21 e 22.

RING, M. et al. A survey of network-based intrusion detection data sets. *Computers & Security*, Elsevier, v. 86, p. 147–167, 2019. Citado na página 9.

SETTLES, B. Active learning literature survey. University of Wisconsin-Madison Department of Computer Sciences, 2010. Citado 2 vezes nas páginas 9 e 17.

SILVA, S. S. et al. Botnets: A survey. *Computer Networks*, v. 57, n. 2, p. 378–403, 2013. ISSN 1389-1286. Botnet Activity: Analysis, Detection and Shutdown. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1389128612003568>. Citado 3 vezes nas páginas 8, 14 e 15.

STALLINGS, W. *Criptografia e segurança de redes: Princípios e práticas*. [S.l.: s.n.], 2014. Citado na página 12.

STALLINGS, W. Computer security third edition, 3rd editio. United States of America: Pearson Education, 2015. Citado na página 11.

TAN, M. S. P.-N.; KUMAR, V. Introduction to data mining. v. 1, 2006. Citado na página 15.

TCU. Boas práticas em segurança da informação. Tribunal de Contas da União, v. 4, 2012. Citado 2 vezes nas páginas 11 e 12.

VIEGAS, E. et al. Bigflow: Real-time and reliable anomaly-based intrusion detection for high-speed networks. *Future Generation Computer Systems*, v. 93, p. 473–485, 2018. ISSN 0167-739X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167739X18307635>. Citado na página 18.

VIEGAS, E. K.; SANTIN, A. O.; OLIVEIRA, L. S. Toward a reliable anomaly-based intrusion detection in real-world environments. *Computer Networks*, Elsevier, v. 127, p. 200–216, 2017. Disponível em: <https://www.sciencedirect.com/science/article/pii/S138912861730322>. Citado na página 12.

ZARPELãO, B. B. et al. A survey of intrusion detection in internet of things. *Journal of Network and Computer Applications*, v. 84, p. 25–37, 2017. ISSN 1084-8045. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804517300802>. Citado na página 13.