Universidade Federal de Pernambuco

Centro de Informática

Engenharia da Computação

Isolation forest como um modelo alternativo para o primeiro estágio de uma abordagem multiestágio para detecção de intrusão

Zenio Angelo Oliveira Neves

Trabalho de Graduação

Recife, Pernambuco

Outubro, 2024

Universidade Federal de Pernambuco
Centro de Informática

Zenio Angelo Oliveira Neves

Isolation forest como um modelo alternativo para o primeiro estágio de uma abordagem multiestágio para detecção de intrusão

Trabalho apresentado ao Curso de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia da Computação.

Área de concentração: Cibersegurança, Machine Learning, Sistemas de Detecção de Intrusão.
Orientador: Paulo Freitas de Araújo Filho.

Recife, Pernambuco
Outubro, 2024

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Neves, Zenio Angelo Oliveira.
Isolation forest como um modelo alternativo para o primeiro estágio de uma abordagem multiestágio para detecção de intrusão / Zenio Angelo Oliveira Neves. - Recife, 2024.
33p : il., tab.

Orientador(a): Paulo Freitas de Araújo Filho
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Engenharia da Computação - Bacharelado, 2024.
Inclui referências.

1. Sistemas de Detecção de Intrusão. 2. Cibersegurança. 3. Machine learning. 4. Abordagens multiestágio. 5. Isolation forest. I. Araújo Filho, Paulo Freitas de. (Orientação). II. Título.

000 CDD (22.ed.)

ZENIO ANGELO OLIVEIRA NEVES

Isolation forest como um modelo alternativo para o primeiro estágio de uma abordagem multiestágio para detecção de intrusão

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia da Computação do Centro de Informática na Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de bacharel em Engenharia da Computação.

Aprovado em: 16/10/2024

BANCA EXAMINADORA

_____

Prof. Dr. Paulo Freitas de Araújo Filho (Orientador)

Universidade Federal de Pernambuco

_____

Prof. Dr. Divanilson Rodrigo de Sousa Campelo (Examinador Interno)

Universidade Federal de Pernambuco

Dedico este trabalho a Deus, Nossa Senhora e minha família.

Agradecimentos

Agradeço especialmente:

- À minha família, pelo apoio e incentivo durante toda a jornada acadêmica.
- À minha namorada, pelo apoio e incentivo durante o desenvolvimento deste trabalho.
- Aos amigos que fiz durante o curso, pela colaboração e pela amizade.
- Ao meu orientador, Professor Paulo Freitas, pelas correções e liberdade para eu desenvolver este projeto no meu próprio ritmo.
- À UFPE, pelos recursos disponibilizados no Centro de Informática (CIn) e pela oportunidade de aprendizado.

"Só porque alguma coisa não faz o que você planejou que ela fizesse, não quer dizer que ela seja inútil"

—Thomas Edison – inventor

Resumo

O uso da Internet aumentou significativamente na última década, levando ao crescimento de cibercrimes. Para identificar intrusões, existem sistemas de detecção de intrusão, ou intrusion detection systems (IDSs) em inglês, que utilizam duas abordagens principais: a detecção baseada em assinatura, que é precisa para ataques conhecidos, e a detecção baseada em anomalias, que pode identificar novos ataques utilizando técnicas de machine learning (ML).

Entre os algoritmos de ML, existem decision tree, que classifica dados com um conjunto de regras, e random forest, que utiliza múltiplas decision trees. Ambos possuem uma complexidade de treinamento quadrática e, geralmente, tempo de inferência mais altos na identificação de dados benignos em comparação com modelos de classificação binária, como os modelos one-class. Para contornar isso, abordagens multiestágio foram desenvolvidas. A abordagem de [3] utiliza um estágio inicial com um autoencoder (AE) para identificar dados benignos e encaminhar os maliciosos para random forest, reduzindo os tempos de inferência de dados benignos e treinamento. A abordagem de [8] acrescenta um terceiro estágio para corrigir decisões do primeiro estágio e utiliza One-Class Support Vector Machine (OCSVM) no primeiro estágio.

Quanto mais rápido um sistema de detecção de intrusão identifica uma ameaça, mais cedo pode ser feita uma intervenção. As abordagens propostas em [3] e [8] apresentaram tempos de inferência de 7 a 8 segundos, enquanto random forest levou 1,5 segundo. Por isso, este trabalho propõe uma abordagem multiestágio de três estágios, que utiliza o algoritmo isolation forest no primeiro estágio, visando um tempo de inferência menor em comparação com os modelos propostos em [3] e [8].

Palavras-chave: IDSs, machine learning, random forest, abordagem multiestágio, isolation forest, tempo de inferência.

Abstract

Internet usage has increased significantly in the last decade, leading to the growth of cybercrimes. To identify intrusions, there are intrusion detection systems (IDSs) that use two main approaches: signature-based detection, which is accurate for known attacks, and anomaly-based detection, which can identify new attacks using machine learning (ML) techniques.

Among the ML algorithms, there are decision tree, which classify data with a set of rules, and random forest, which use multiple decision trees. Both have a quadratic training complexity and generally higher inference times in identifying benign data compared to binary classification models, such as one-class models. To overcome this, multistage approaches have been developed. [3]'s approach uses an initial stage with an autoencoder (AE) to identify benign data and forward malicious data to random forests, reducing the inference times for benign data and training. The [8] approach adds a third stage to correct decisions from the first stage and uses One-Class Support Vector Machine (OCSVM) in the first stage.

The faster an intrusion detection system identifies a threat, the sooner an intervention can be made. The approaches proposed in [3] and [8] presented inference times of 7 to 8 seconds, while random forest took 1.5 second. Therefore, this work proposes a three-stage multistage approach, which uses the isolation forest algorithm in the first stage, aiming at a shorter inference time compared to the models proposed in [3] and [8].

Keywords: IDSs, machine learning, random forest, multistage approach, isolation forest, inference time.

Lista de Tabelas

Lista de Figuras

Sumário

Capítulo 1

Introdução

O uso da Internet, a rede global de computadores interconectados, cresceu significativamente na última década. Em 2022, 82% das residências urbanas em todo o mundo utilizavam a rede global [6]. Com o aumento do uso da Internet, os cibercrimes também se expandem, incluindo atividades como chantagem, invasão de privacidade, roubo de identidade e negação de serviço (DoS), entre outras ilegalidades [6]. O prejuízo financeiro causado subiu de dois trilhões de dólares em 2015 para seis trilhões de dólares em 2016 [6].

Sistemas de detecção de intrusão, ou intrusion detection systems (IDSs) em inglês, monitoram eventos a fim de identificarem indícios de intrusão [7]. Existem duas principais abordagens para sistemas de detecção de intrusão: a detecção baseada em assinatura e a detecção baseada em anomalias [7]. Os sistemas baseados em assinatura utilizam um banco de dados de assinaturas, que são identificadores de dados maliciosos, para verificar se a assinatura dos dados investigados corresponde a alguma das assinaturas registradas. Porém, em caso de variantes de ataques já conhecidos ou ataques novos, esses sistemas podem não detectar a ameaça, o que leva à adoção da segunda abordagem. A detecção baseada em anomalias, por sua vez, envolve o uso de técnicas a fim de definir um comportamento para um conjunto de amostras. Dessa forma, é possível identificar dados que não se comportem de acordo com o comportamento estabelecido [7]. IDSs baseados em anomalias podem utilizar diversas técnicas de machine learning (ML), incluindo supervisionada, que requer treinamento com dados rotulados; semi-supervisionada, que utiliza dados parcialmente rotulados; e não supervisionada, que opera com dados não rotulados [6].

É possível combinar dois modelos de ML, como, por exemplo, na estratégia multiestágio com dois estágios: um para detecção de anomalias e outro para classificação de anomalias

[3]. No primeiro estágio, um modelo one class novelty é treinado com dados benignos e o segundo estágio, com modelo random forest, é treinado somente com dados maliciosos. Assim, o primeiro estágio envia dados maliciosos para o segundo estágio classificar. Esta abordagem permite a detecção mais rápida de dados benignos, reduz a quantidade de dados usados no treinamento da random forest e, consequentemente, reduz o custo computacional [3]. E, para melhorar ainda mais o desempenho, em termos de acurácia e acurácia balanceada, [8] propõe uma abordagem de três estágios. Neste modelo, o terceiro estágio é responsável por corrigir as decisões do primeiro estágio, o que promove um aprimoramento de desempenho em relação ao modelo proposto em [3].

Quanto mais rápido um sistema de detecção de intrusão identifica uma ameaça, mais cedo pode ser feita uma intervenção. Assim, o tempo de inferência de um modelo de detecção de intrusão é um critério crucial, especialmente em cenários de tempo real. Os modelos apresentados em [8] e [3] tiveram um tempo de inferência cerca de 8 e 7 segundos, respectivamente, enquanto que o tempo da random forest foi de 1,5 segundo [8].

Assim, este trabalho de graduação propõe uma abordagem multiestágio como a proposta em [8], mas com uso do algoritmo isolation forest no primeiro estágio, a fim de reduzir o tempo de detecção em comparação com os IDSs propostos em [8] e [3].

Capítulo 2

Fundamentação Teórica

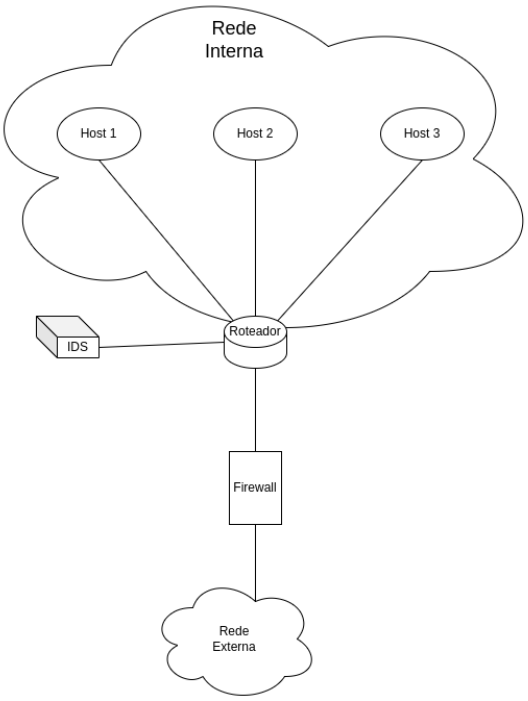

Figura 2.1: IDS numa topologia de rede.

A detecção de intrusão é o processo de monitorar e analisar o tráfego de rede (ver figura 2.1) em busca de sinais de intrusão [1]. Um dos critérios para classificar um sistema de detecção de intrusão é a abordagem: baseada em assinatura ou em anomalias [1]. Os IDSs baseados em assinatura utilizam um banco de dados de assinaturas de ameaças conhecidas. Quando uma assinatura de um dado de entrada corresponde a uma assinatura presente no banco, um alerta é gerado. A principal vantagem desses sistemas é a alta precisão na detecção de ameaças conhecidas. Contudo, eles não conseguem identificar ataques desconhecidos. Por outro lado, IDSs baseados em anomalia empregam técnicas, como machine learning (ML), a

fim de estabelecer um comportamento para um conjunto de amostras e identificar desvios em relação a esse comportamento [5]. Esses sistemas têm a vantagem de detectar ataques novos, mas geralmente apresentam falsos positivos.

Machine learning, ou aprendizagem de máquina, é uma área da inteligência artificial (IA) que visa treinar computadores para tomar decisões e fazer predições utilizando um conjunto de dados [6]. ML pode ser classificada em três tipos: supervisionada, não supervisionada e semi-supervisionada [6]. A principal diferença entre esses tipos está na rotulação dos dados usados no treinamento [6]. O aprendizado supervisionado utiliza dados totalmente rotulados, o aprendizado não supervisionado utiliza dados não rotulados, e o aprendizado semi-supervisionado utiliza dados parcialmente rotulados [6]. Dados rotulados são vantajosos porque ajudam no treinamento de modelos que fornecem predições com rótulos.

Entre as técnicas de aprendizado supervisionado, está a decision tree (ver figura 2.2), que realiza a classificação de dados por meio de um conjunto de regras [5]. O algoritmo de decision tree pode eliminar características irrelevantes ou redundantes através de um processo de seleção de características [5]. A técnica de random forest (ver figura 2.3) utiliza múltiplas decision trees [5]. A complexidade temporal de treinamento de uma decision tree é $O(f \cdot n^2 \cdot \log(n))$, onde f representa o número de características dos dados e n é o número de dados de entrada. Assim, a complexidade de treinamento de uma random forest é $O(a \cdot f \cdot n^2 \cdot \log(n))$, onde a é o número de decision trees na random forest. Devido a sua complexidade de treinamento, a random forest pode apresentar um tempo de treinamento elevado quando o tamanho do conjunto de dados é grande.

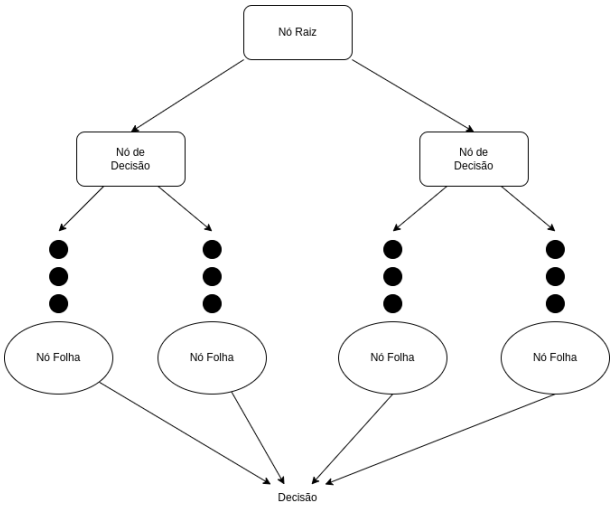

Figura 2.2: Decision tree.

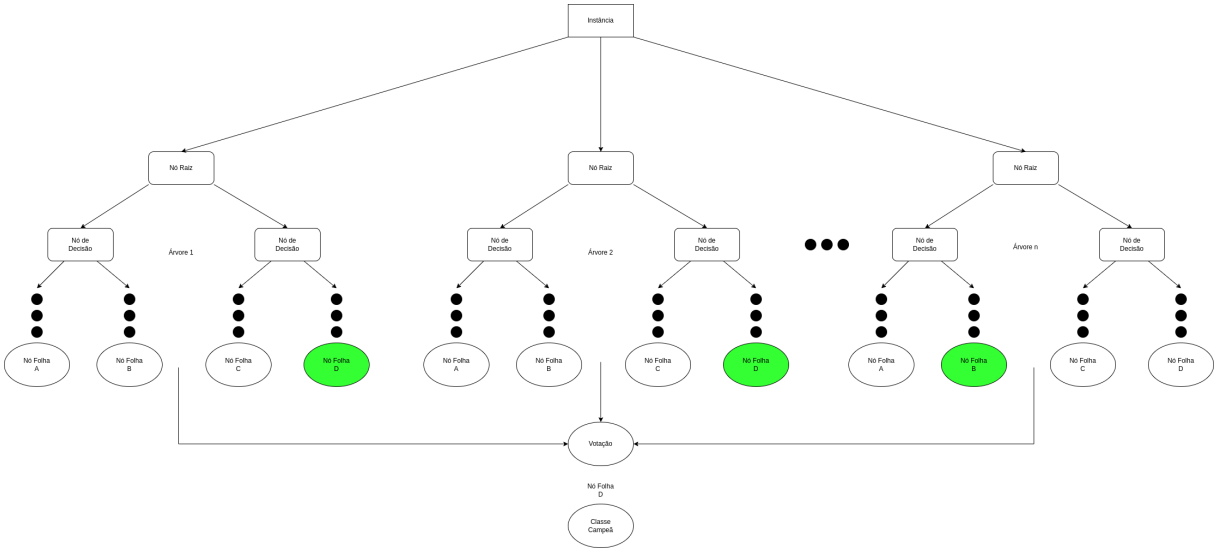

Figura 2.3: Random forest.

Entre as técnicas de aprendizado não supervisionado, está o autoencoder (AE). Um AE possui dois componentes, um encoder e um decoder [5] (ver figura 2.4). O encoder extrai as características dos dados brutos, e o decoder reconstrói os dados a partir das características extraídas [5]. Durante o treinamento, a divergência entre a entrada do encoder e a saída do decoder é gradualmente reduzida [5]. Quando o decoder consegue reconstruir os dados por meio das características extraídas, isso significa que as características extraídas pelo encoder representam a essência dos dados [5]. O deep autoencoder (DAE) é um AE com múltiplas camadas de encoders e decoders [3] (ver figura 2.5) e o Modality2-DAE (M2-DAE) é um DAE com separação das características em numérica e categóricas [3] (ver figura 2.6). A complexidade de treinamento de um AE é dada por $O(n \cdot e \cdot p)$, onde e é o número de épocas e p é o número de parâmetros da rede neural. A complexidade de treinamento de um DAE também é dada por $O(n \cdot e \cdot p)$, mas o valor de p aumenta à medida que a quantidade de camadas de encoders e decoders aumenta. A complexidade de treinamento de um M2-DAE é dada pela soma

$$O(n \cdot e \cdot p_{\text{numérico}}) + O(n \cdot e \cdot p_{\text{categórico1}}) + \ldots + O(n \cdot e \cdot p_{\text{categórico}k}),$$

onde k é a quantidade de características categóricas dos dados de entrada.

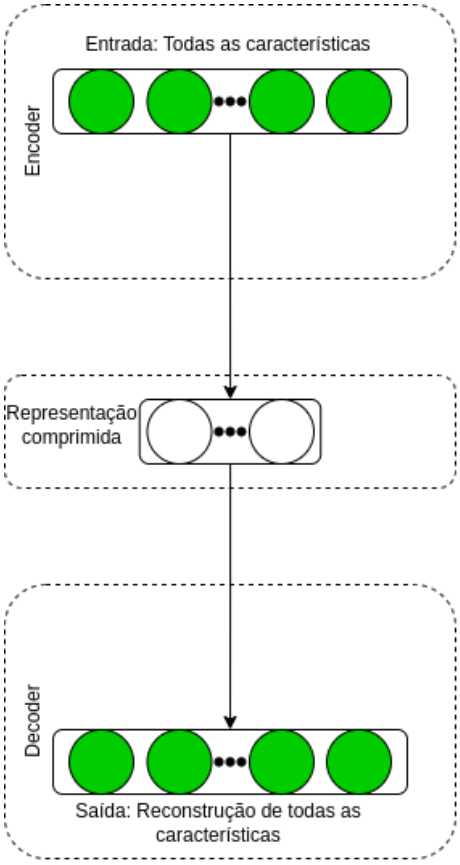

Figura 2.4: Autoencoder.

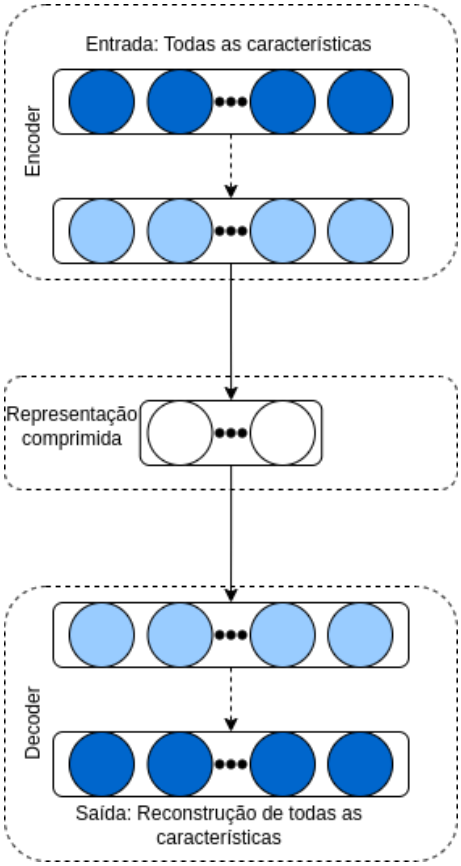

Figura 2.5: Deep autoencoder.

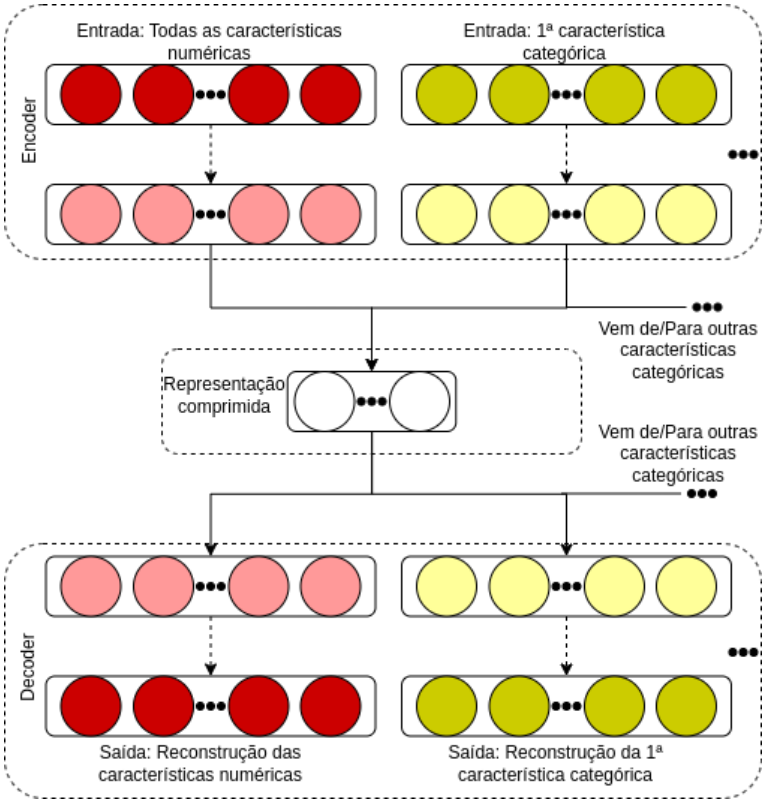

Figura 2.6: Modality2 - Deep autoencoder.

Support Vector Machine (SVM) (ver figura 2.7) é uma técnica de aprendizado supervisionado que visa encontrar um hiperplano de separação com a maior margem possível. As SVMs podem obter bons resultados mesmo com conjuntos de dados pequenos, pois o hiperplano de separação é definido por um número reduzido de vetores de suporte [5]. O One-Class SVM (OCSVM) (ver figura 2.8) é uma técnica de detecção de anomalias que, em vez de usar um hiperplano, utiliza uma hiperesfera para isolar dados normais de anômalos [8]. Nesse contexto, é possível determinar se um dado está dentro da hiperesfera calculando a distância entre o dado e todos os vetores de suporte. Uma abordagem one-class novelty é voltada para a classificação de dados em apenas dois tipos: anômalos e normais. A complexidade temporal de treinamento de um OCSVM varia de $O(n^2)$ a $O(n^3)$, dependendo da implementação e dos parâmetros utilizados. Para a inferência, a complexidade é de $O(m)$, onde m representa o número de vetores de suporte.

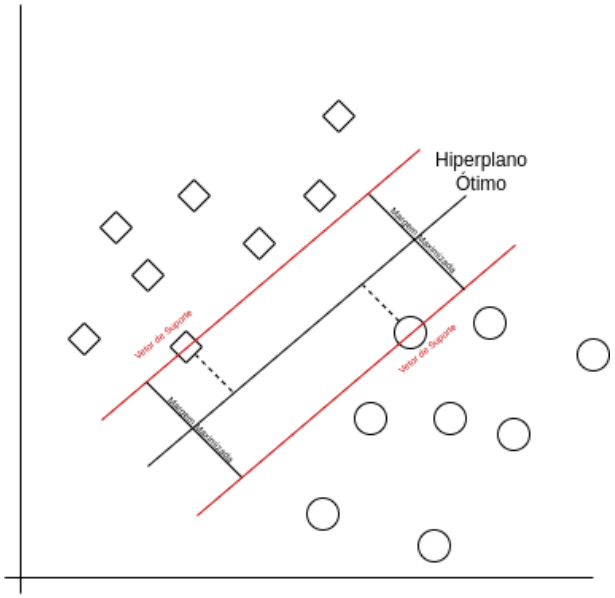

Figura 2.7: Support Vector Machine (SVM).

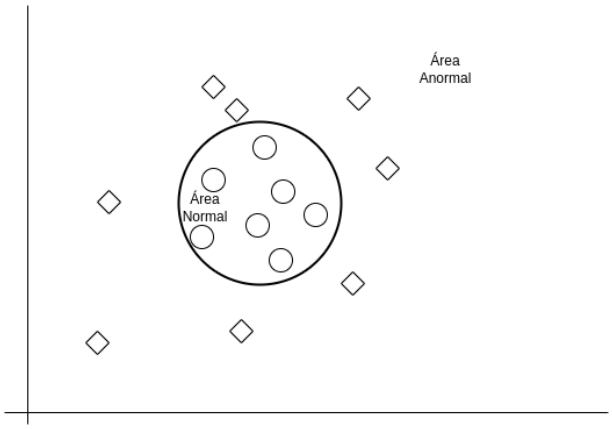

Figura 2.8: One-Class Support Vector Machine (OCSVM).

O algoritmo isolation tree (ver figura 2.9) consiste em isolar uma amostra em relação às outras através da construção de uma árvore [4]. A técnica se baseia na hipótese de que amostras anômalas são isoladas mais facilmente. Assim, após a construção de uma isolation tree, as anomalias tendem a aparecer em nós menos profundos. Quando várias isolation trees são combinadas, a estrutura resultante é chamada de isolation forest (ver figura 2.10). A isolation forest analisa a profundidade de uma amostra em cada isolation tree para determinar se a amostra é anômala [4]. A complexidade temporal para treinar uma isolation forest é $O(n \cdot \log(n))$, enquanto a complexidade para a inferência é $O(a \cdot r)$, onde r é o número médio de nós nas isolation trees.

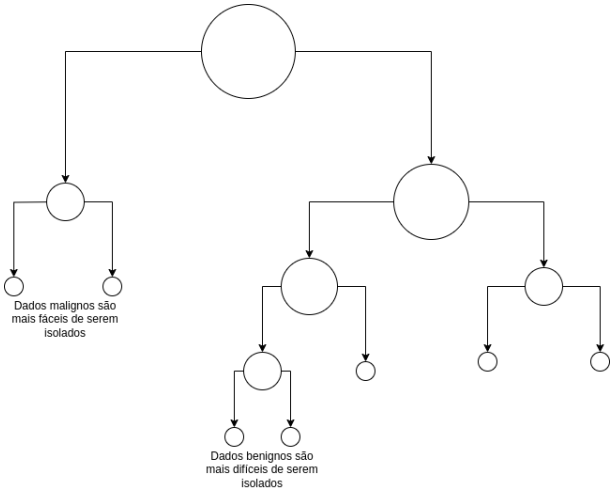

Figura 2.9: Isolation tree.

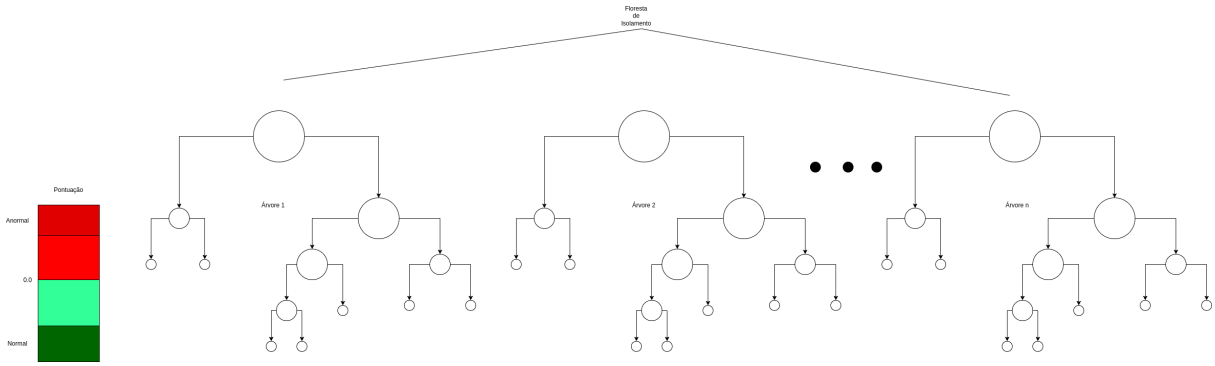

Figura 2.10: Isolation forest.

Comparando OCSVM e isolation forest, observa-se que o segundo modelo pode apresentar um tempo de inferência menor em relação ao primeiro. Isso depende da quantidade de vetores de suporte na OCSVM, do número de isolation trees na isolation forest e nós em cada isolation tree.

Capítulo 3

Trabalhos Relacionados

A abordagem de sistemas de detecção de intrusão (IDSs), com um único estágio, emprega um modelo de detecção usando o algoritmo supervisionado random forest. Este modelo é capaz de classificar entre fluxos de rede benignos e maliciosos.

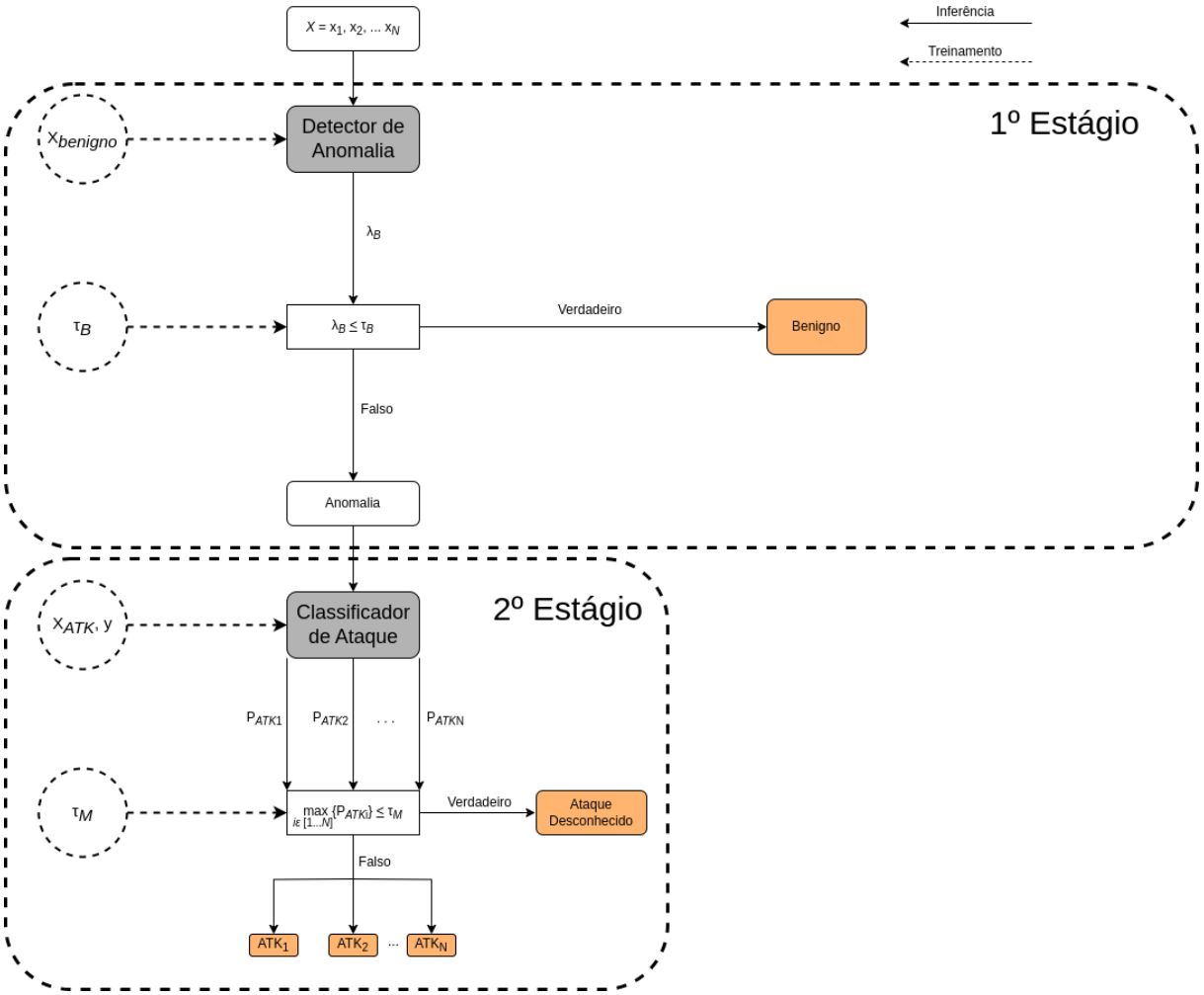

Figura 3.1: Arquitetura da abordagem multiestágio de dois estágios.

A abordagem de dois estágios proposta em [3] (ver figura 3.1) usa o algoritmo não supervisionado M2-DAE no primeiro estágio e utiliza o algoritmo supervisionado random forest no segundo. No primeiro estágio, acontece a detecção de fluxos benignos, na qual um fluxo de rede recebe uma pontuação de anomalia em que, no caso de pontuação abaixo de um limiar, o fluxo é identificado como benigno e não é passado para o próximo estágio. No caso de um fluxo receber uma pontuação acima do limiar, ele é passado para o segundo estágio, que irá classificá-lo de acordo com as classes de ataques que recebeu durante o treinamento.

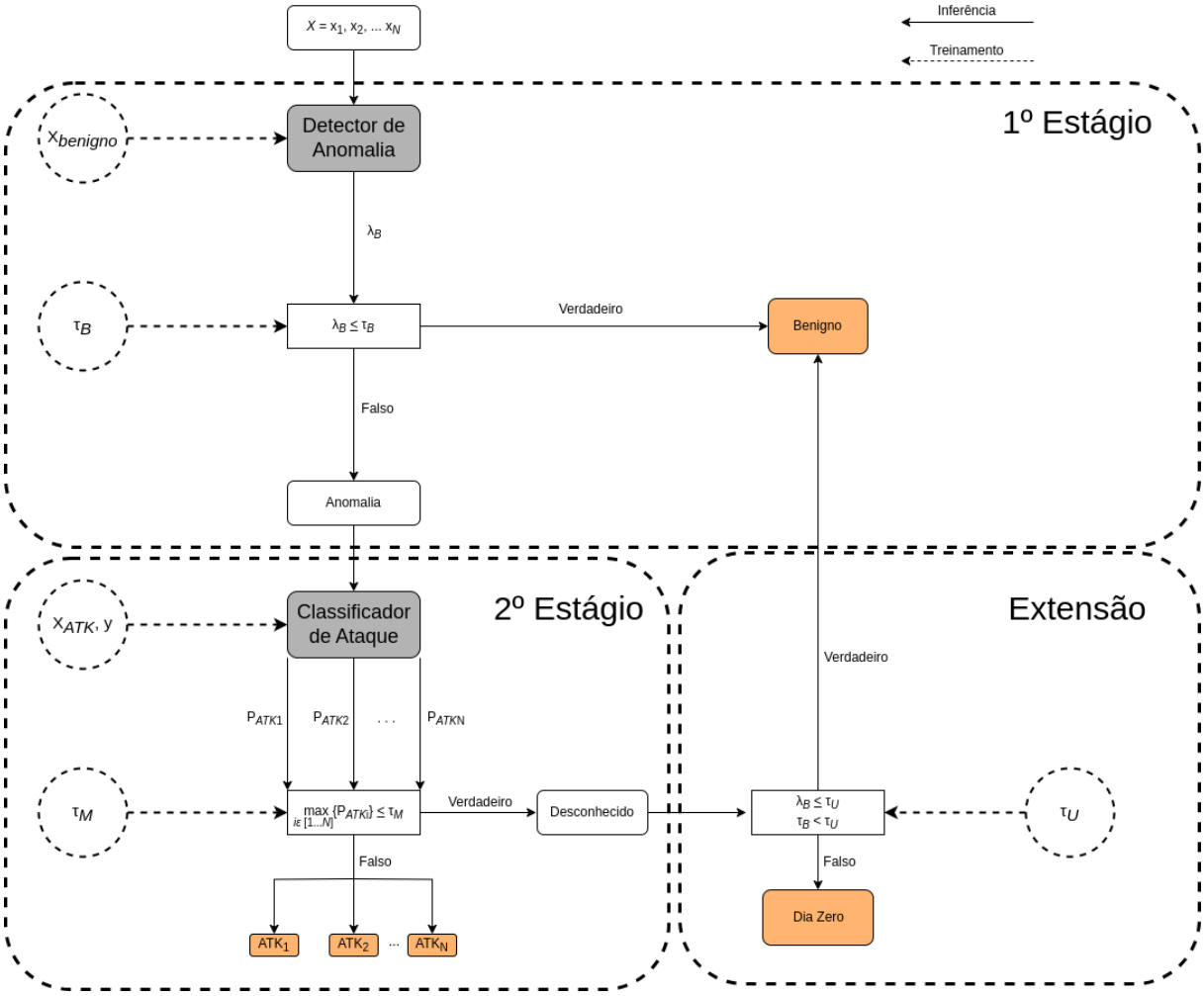

Figura 3.2: Arquitetura da abordagem multiestágio de três estágios.

A abordagem multi-estágio de três estágios (ver figura 3.2), proposta por [8], funciona de forma semelhante à abordagem proposta em [3]. Os dois primeiros estágios funcionam exatamente da mesma maneira que os dois estágios propostos em [3], enquanto que existe a adição de um terceiro estágio que obtém os fluxos de rede classificados como desconhecidos (ataques de dia-zero) e repete o procedimento adotado no primeiro estágio com uma exceção: o limiar usado é maior do que aquele usado no primeiro estágio. Sendo assim, em caso da

pontuação do fluxo ser maior do que o valor do novo limiar, o fluxo, de fato, será classificado como desconhecido e, caso contrário, o mesmo deverá ter sua identificação no primeiro estágio corrigida, isto é, o fluxo será considerado benigno [8]. O primeiro estágio usa um modelo OCSVM enquanto que o segundo utiliza um modelo random forest. No terceiro estágio, há somente uma comparação, na qual, se a pontuação de anomalia recebida no primeiro estágio for menor do que o valor do limiar do terceiro estágio, o fluxo é identificado como benigno. Caso contrário, o fluxo é classificado como desconhecido. A principal desvantagem dessa abordagem, em comparação com as outras duas, é o maior tempo de inferência. Por outro lado, é uma abordagem mais acurada devido ao terceiro estágio, que pode corrigir decisões do primeiro estágio.

A tabela 3.1 resume as características dos trabalhos relacionados.

Trabalho	Detecção	Classificação	Estágios	Dia-Zero
RF Baseline. [8]	Não	Sim	1	Sim
Bovenzi et al. [3]	Sim	Sim	2	Sim
Verkerken et al. [8]	Sim	Sim	3	Sim
Este	Sim	Sim	3	Sim

Tabela 3.1: Trabalhos relacionados

Capítulo 4

Solução Proposta

A solução proposta adota uma abordagem multiestágio com três fases distintas, como a proposta em [8] (ver figura 3.2).

No primeiro estágio, realiza-se a detecção de anomalias usando o algoritmo isolation forest. Neste estágio, é atribuída uma pontuação de anomalia a cada fluxo de rede. Se a pontuação estiver abaixo de um limiar definido, o fluxo é considerado benigno e não avança para o próximo estágio. Caso a pontuação ultrapasse o limiar, o fluxo é encaminhado ao segundo estágio, com modelo random forest, onde é classificado em uma das classes de ataques presentes no conjunto de treinamento. Esta classificação também é baseada em um limiar específico: cada tipo de ataque recebe uma probabilidade, e se a maior probabilidade exceder o limiar do segundo estágio, o fluxo é classificado com o tipo correspondente à maior probabilidade. Se a maior probabilidade for inferior ao limiar, o fluxo é considerado desconhecido e enviado ao terceiro estágio. No terceiro estágio, que consiste somente de uma comparação sem uso de modelos de machine learning (ML), utiliza-se um limiar mais alto do que o do primeiro estágio para comparar com a pontuação obtida anteriormente. Se a pontuação for menor que este limiar, a classificação do fluxo é ajustada e ele é rotulado como benigno. Caso contrário, o fluxo é classificado como um ataque desconhecido.

A principal vantagem desta abordagem é o tempo de inferência reduzido em comparação com os modelos apresentados em [3] e [8], pois o tempo de inferência do modelo isolation forest é menor que os tempos de inferência dos modelos M2-DAE e OCSVM. Assim, a solução proposta se diferencia da abordagem de [8] principalmente pelo modelo utilizado no primeiro estágio. Em vez do One Class Support Vector Machine (OCSVM) empregado na referência, a solução proposta utiliza a isolation forest para a detecção de anomalias.

Capítulo 5

Experimentos e Resultados

Para a realização dos experimentos e obtenção dos resultados, foi utilizado um ambiente de execução no Google Colab, com 12,7 GB de memória RAM, 107,7 GB de armazenamento e uma CPU.

No treinamento da isolation forest, foram utilizados os mesmos dez mil dados benignos do treinamento da OCSVM descritos em [8], que fazem parte do conjunto de dados CIC-IDS-2017. Cada fluxo de rede neste conjunto de dados possui 67 atributos. Para validação e teste, foram empregados os mesmos dados utilizados com a OCSVM, 129485 dados benignos e 6820 dados maliciosos. Dados da classe heartbleed e infiltration, compondo a classe desconhecido, só foram usados no conjunto de teste. Assim como em [8], foi criado um pipeline (ver figura 5.1) que inclui um escalonador com distribuição normal para garantir que os valores de cada atributo no conjunto de dados de entrada fiquem na mesma escala, seguido por um redutor de dimensionalidade sensível à escala, considerando 56 das 67 dimensões a fim de otimizar os tempos de treinamento e de inferência. O último componente do pipeline é o isolation forest, que utiliza 98 estimadores. A tabela 5.1 mostra, em detalhes, a quantidade de dados usada no treinamento, validação e teste.

	Benign	(D)DOS	Botnet	Brute Force	Heartbleed	Infiltration	Port Scan	Web Attack	Attack (Sum)
Training (Stages 1 and 3)	10000	0	0	0	-	-	0	0	0
Validation (Stages 1 and 3)	129485	1364	1364	1364	-	-	1364	1364	6820
Test (Full-Model)	56468	584	584	584	11	36	584	584	2967

Tabela 5.1: Quantidade de dados usada no treinamento, validação e teste.

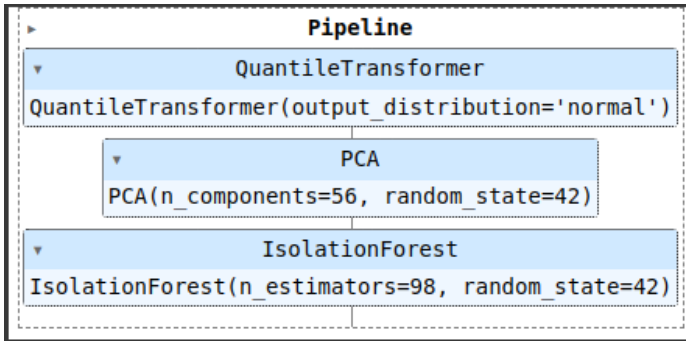

Figura 5.1: Pipeline.

A escolha dos limiares foi realizada durante o treinamento do modelo isolation forest, de forma semelhante a [8], utilizando f-scores, que se trata de nove valores possíveis de limiares para o primeiro estágio com isolation forest. Estas nove métricas são nomeadas de F1 até F9 (ver tabela 5.2) e representam limiares possíveis de serem usados no conjunto de validação. Seus respectivos limiares encontram-se, também, na tabela 5.2, acompanhados de um false positive rate (FPR) e de um true positive rate (TPR). Os valores de FPR e TPR são obtidos ao usar o limiar correspondente nos fluxos de rede do conjunto de validação. Esses limiares são obtidos a partir dos pontos de anomalia atribuídos aos fluxos de rede pela isolation forest. Uma tabela de quantis também foi usada no treinamento da isolation forest. Nessa tabela, há quatro valores de limiares para o terceiro estágio, obtidos a partir dos quantis das pontuações de anomalia, fornecidas pela isolation forest, dos fluxos benignos no conjunto de validação (ver tabela 5.3).

Métrica	Limiar	TPR	FPR
F1	-0,023618	0,3257	0,079252
F2	-0,096991	0,7452	0,305896
F3	-0,098074	0,7567	0,314484
F4	-0,122349	0,9139	0,489138
F5	-0,138989	0,9812	0,620674
F6	-0,138989	0,9812	0,620674
F7	-0,138989	0,9812	0,620674
F8	-0,139984	0,9833	0,628876
F9	-0,140556	0,9843	0,633108

Tabela 5.2: F-scores.

Quantil	Limiar
0,95	0,005137046051861486
0,975	0,03358335958452108
0,99	0,06766017781741557
0,995	0,09091109171053309

Tabela 5.3: Quantis.

Com base na tabela 5.2, a métrica F4 foi escolhida, pois garante um TPR acima de 90% e um FPR abaixo de 60% no primeiro estágio. E, com base na tabela 5.3, o quantil 0,99 teve o seu limiar escolhido para o terceiro estágio, já que o quantil 0,99 também foi utilizado em [8]. Como o isolation forest tende a apresentar uma taxa de falsos positivos mais alta em comparação com a OCSVM, foi permitida uma maior tolerância para a FPR a fim de alcançar um TPR mais próximo dos 96% obtidos em [8] no estágio de detecção de anomalias. Consequentemente, não houve uma perda significativa no desempenho do TPR no primeiro estágio (ver figuras 5.2 e 5.3).

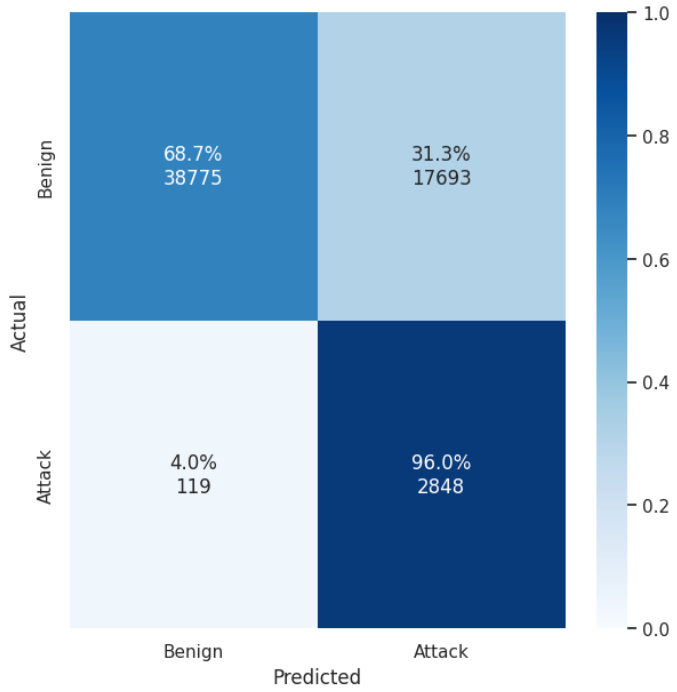

Figura 5.2: Matriz de confusão do estágio 1 com OCSVM.

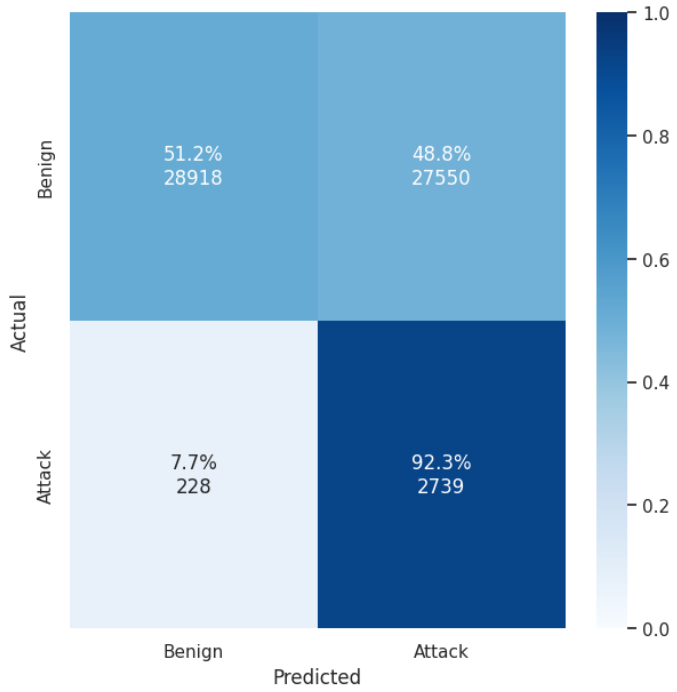

Figura 5.3: Matriz de confusão do estágio 1 com isolation forest.

No segundo estágio, foi utilizado o modelo random forest treinado em [8]. Por isso, o valor de limiar utilizado neste estágio foi o mesmo empregado em [8].

O terceiro estágio envolve uma comparação entre a pontuação de anomalia da amostra, obtida no primeiro estágio, e o limiar definido para o terceiro estágio.

De acordo com as matrizes de confusão nas figuras 5.4 e 5.5, a solução proposta mostrou um desempenho superior na detecção de dados benignos e ataques (D)DoS, enquanto a abordagem descrita em [8] apresentou um desempenho melhor na identificação de ataques desconhecidos e botnets. Esse equilíbrio indica que, apesar da alta FPR no primeiro estágio, o segundo estágio compensa com classificações precisas. Além disso, o estágio de extensão demonstrou uma boa capacidade de correção, destacando-se pela maior TPR na identificação de dados benignos apresentada pela solução proposta.

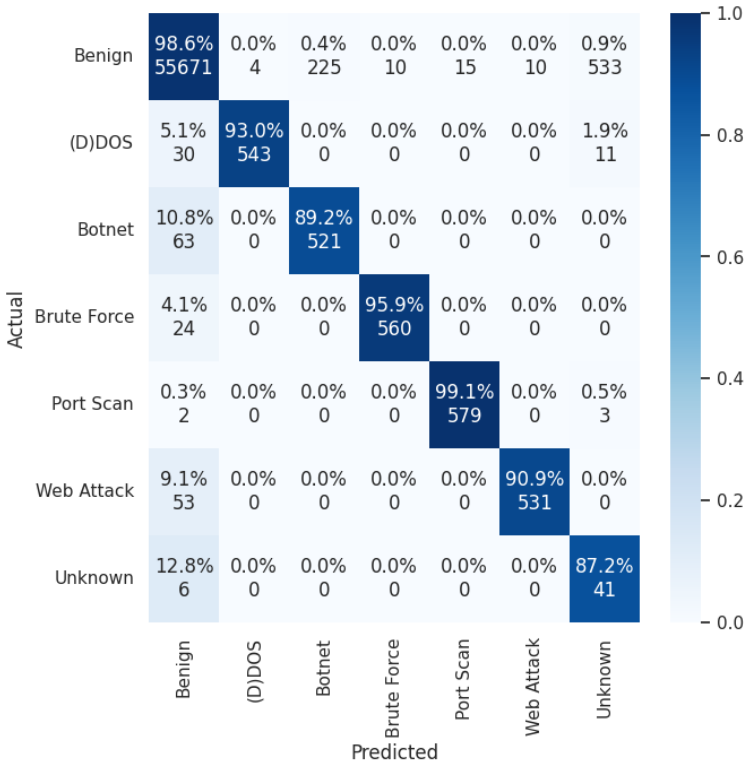

Figura 5.4: Matriz dos três estágios combinados com OCSVM no primeiro estágio.

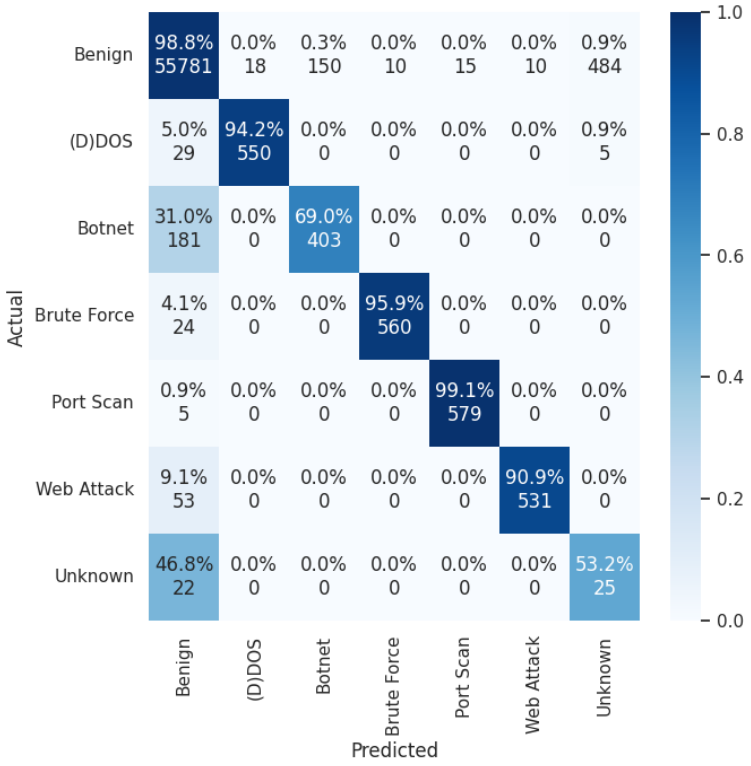

Figura 5.5: Matriz dos três estágios combinados com isolation forest no primeiro estágio.

A tabela 5.4 demonstra que a solução proposta apresentou o menor tempo de inferência em comparação com as soluções de [3] e [8]. Embora o segundo estágio exija mais recursos devido

à maior FPR no primeiro estágio, o tempo adicional necessário para os estágios dois e três é inferior ao tempo economizado no primeiro estágio, quando comparado às soluções de [8] e [3]. Todas as abordagens listadas na tabela 5.4 foram executadas no mesmo ambiente de execução no Google Colab a fim de garantir consistência na medição do tempo de inferência.

	τ_B	τ_M	τ_U	F1 weighted	F1 macro	Accuracy	Bal. Accuracy	Bandwidth Reduction	zero-day recall	Inference (s)
Max F-score Verkerken et al. [8]	F5-8	F1	0,995	0,9897	0,8276	0,9877	0,8954	68,75%	0,5957	6,766 ± 0,407
Max bACC Verkerken et al. [8]	F9	F1	0,95	0,9580	0,7496	0,9341	0,9608	57,91%	0,9574	6,692 ± 0,294
Balanced Verkerken et al. [8]	F5-8	F1	0,99	0,9875	0,8231	0,9834	0,9342	65,44%	0,8723	6,609 ± 0,230
This	**F4**	**F1**	**0,99**	**0,9865**	**0,8059**	**0,9831**	**0,8587**	**49,04%**	**0,5319**	**4,466 ± 0,024**
RF Baseline. [8]	-	-	-	0,9849	0,7981	0,9832	0,8877	-	0,8936	1,275 ± 0,055
Bovenzi et al. [3]	F3	F1	-	0,9383	0,7549	0,8957	0,8550	86,22%	0,9574	6,459 ± 0,376

Tabela 5.4: Resultados.

Capítulo 6

Conclusão e Trabalhos Futuros

Este trabalho desenvolveu uma abordagem multiestágio com três estágios, na qual utilizou-se o algoritmo de isolation forest no primeiro estágio. O treinamento do modelo ocorreu por meio do mesmo conjunto de dados usados no treinamento do modelo OCSVM, em [8]. Além disso, os limiares do primeiro e do terceiro estágios foram obtidos a partir das tabelas 5.2 e 5.3. Assim, a abordagem desenvolvida neste trabalho obteve uma acurácia superior à acurácia do modelo proposto em [3] e próxima da acurácia do modelo proposto em [8] enquanto apresentou um menor tempo de inferência. Contudo, a solução proposta mostrou uma queda de desempenho nos recalls das classes de ataques zero-day e botnets.

Portanto, a abordagem proposta se mostrou uma alternativa eficiente em termos de tempo de inferência, superando os modelos apresentados em [8] e [3] ao oferecer uma execução mais rápida sem uma perda significativa na acurácia. No entanto, é crucial realizar um estudo focado em aprimorar o modelo isolation forest, visando melhorar o recall para zero-day e botnets, por exemplo. Além disso, seria interessante investigar a possibilidade de reduzir o false positive rate no primeiro estágio, em detrimento do tempo de inferência, de modo que o tempo ainda seja inferior ao dos modelos propostos em [8] e [3].

Referências Bibliográficas

[1] ABDALLAH, Emad E. et al. Intrusion detection systems using supervised machine learning techniques: a survey. **Procedia Computer Science**, v. 201, p. 205-212, 2022.

[2] BOUZIDA, Yacine; CUPPENS, Frederic. Neural networks vs. decision trees for intrusion detection. In: **IEEE/IST workshop on monitoring, attack detection and mitigation (MonAM)**. 2006.

[3] BOVENZI, Giampaolo et al. A hierarchical hybrid intrusion detection approach in IoT scenarios. In: **GLOBECOM 2020-2020 IEEE global communications conference**. IEEE, 2020. p. 1-7.

[4] LIU, Fei Tony; TING, Kai Ming; ZHOU, Zhi-Hua. Isolation forest. In: **2008 eighth ieee international conference on data mining**. IEEE, 2008. p. 413-422.

[5] LIU, Hongyu; LANG, Bo. Machine learning and deep learning methods for intrusion detection systems: A survey. **applied sciences**, v. 9, n. 20, p. 4396, 2019.

[6] NI, Man. A review on machine learning methods for intrusion detection system. **Applied and Computational Engineering**, v. 27, p. 57-64, 2023.

[7] UPPAL, Hussain Ahmad Madni; JAVED, Memoona; ARSHAD, M. An overview of intrusion detection system (IDS) along with its commonly used techniques and classifications. **International Journal of Computer Science and Telecommunications**, v. 5, n. 2, p. 20-24, 2014.

[8] VERKERKEN, Miel et al. A novel multi-stage approach for hierarchical intrusion detection. **IEEE Transactions on Network and Service Management**, v. 20, n. 3, p. 3915-3929, 2023.