Detecção de Ataques DDoS com Base em Métricas de Tráfego usando Redes Convolucionais

Antonio Alcir de Freitas Junior
Programa de Pós-Grad. em Eng. Elétrica e de Computação
Universidade Federal do Rio grande do Norte (UFRN)
Natal, Brasil
alcir_jr13@hotmail.com

Francisco S. de Lima Filho
Diretoria Acadêmica de Gestão e Tecnologia da Informação
nstituto Federal do Rio Grande do Norte (IFRN)
Natal, Brasil
sales.filho@ifrn.edu.br

Agostinho de M. Brito Junior
Departamento de Eng. de Computação e Automação (DCA)
Universidade Federal do Rio Grande do Norte (UFRN)
Natal, Brasil
ambj@dca.ufrn.br

Luiz Felipe Silveira
Departamento de Eng. de Computação e Automação (DCA)
Universidade Federal do Rio Grande do Norte (UFRN)
Natal, Brasil
lfelipe@dca.ufrn.br

***Resumo*—Este trabalho propõe o uso de uma Rede Neural Convolucional para detectar ataques de negação de serviço distribuído usando métricas de tráfego amostrado em ativos de rede. O treinamento da rede proposta é baseado em imagens formadas a partir das métricas consideradas. Após o treinamento, a rede usa imagens formadas a partir do fluxo de rede amostrado para diferir o tráfego normal de tráfego anômalo em um curto período de tempo. Dado que o tamanho das imagens criadas pela abordagem proposta é pequena, a rede convolucional é capaz de operar com eficiência, provendo respostas rápidas e bem sucedidas para os testes realizados, sendo competitiva quando comparada com trabalhos semelhantes na literatura.**

***Index Terms*—Redes Convolucionais, Segurança da Informação, Ataques de Negação de Serviço, Métricas de Tráfego.**

I. Introdução

Ataques de DDoS (Distributed Denial of Service) podem causar danos significativos em redes e sistemas, tais como interrupção de serviços, perda de dados, comprometimento da privacidade de usuários e prejuízos financeiros [1]. Os alvos de ataques DDoS aumentaram cerca de 22% no setor de serviços financeiros no ano de 2021 em comparação ao ano anterior [2], perdendo apenas para o setor de jogos, o que reforça a importância de técnicas eficazes de detecção e mitigação destes ataques.

A frequência e a sofisticação de ataques DDoS estão aumentando, tornando a detecção precoce uma tarefa cada vez mais desafiadora. As técnicas recentes de detecção de ataques DDoS concentram-se na coleta e análise de dados de tráfego em tempo real. No entanto, a grande quantidade de dados envolvidos e a complexidade dos padrões de tráfego dificultam a detecção de ataques DDoS por meio de técnicas convencionais. Neste contexto, técnicas de aprendizagem de máquina têm sido amplamente utilizadas para a detecção de DDoS, como as que fazem uso dos algoritmos Adaboost e o Random Forest [3–5]. No entanto, a eficácia desses algoritmos depende da seleção adequada de características relevantes para a detecção de ataques DDoS, o que pode aumentar o custo computacional.

Recentemente, a utilização de redes neurais para a detecção de ataques DDoS tem chamado a atenção da comunidade científica, uma vez que essa abordagem pode lidar com grandes volumes de dados de tráfego de rede e extrair características relevantes para a detecção de ataques [6–8], eventualmente também a um custo computacional elevado, especialmente na etapa de treinamento.

As redes neurais convolucionais (CNNs) são uma classe de redes neurais profundas que são especialmente adequadas para a análise de imagens e outras formas de dados multidimensionais. Essas redes são capazes de extrair características e gerar descritores relevantes a partir de dados brutos, e têm atraído a atenção de trabalhos voltados para a análise de tráfego de rede.

Em uma das propostas baseadas em CNN [9], os autores apresentaram um classificador para detectar ataques DDoS em redes de computadores treinado por meio de imagens construídas diretamente de dados de cabeçalho do tráfego de rede amostrado. São consideradas sete variáveis do cabeçalho, como IPs e portas de origem e destino dos quadros, cujos valores são representados em base 2 e organizados em imagens binárias, utilizadas como descritores das características do tráfego analisado.

Apesar de as CNNs conseguirem aprender a partir de dados brutos, a quantidade de dados processados e o número de camadas da rede podem resultar em um custo computacional proibitivo tanto para aplicações em tempo real em redes com taxas de dados elevadas, quanto para sistemas embarcados, de poder computacional reduzido.

Neste trabalho, é apresentada uma nova abordagem baseada em CNNs para a detecção de ataques DDoS, usando como entrada para o modelo imagens criadas a partir de métricas de tráfego, calculadas a partir de variáveis do cabeçalho dos quadros de rede amostrados, reduzindo assim consideravelmente

o número de camadas internas e de neurônios de entrada da CNN, logo diminuindo a complexidade da CNN necessária para a tarefa de classificação. A ideia é que essas imagens de entrada da CNN representem o estado atual do tráfego de rede, e que a partir dessas imagens possam ser identificados padrões que indiquem a ocorrência de ataques DDoS. A utilização de imagens para a detecção de DDoS pode fornecer uma nova visão sobre a relação existente entre os diversos parâmetros do tráfego amostrado, explorando evidências que facilitem a ação das ferramentas de inteligência artificial.

Serão mostrados o método desenvolvido, juntamente com os resultados obtidos na classificação de amostras de tráfegos considerando um banco de dados produzido por Lima-Filho [10]. Os resultados serão comparados com estudos anteriores que também utilizam métodos de aprendizagem de máquina aplicados ao mesmo banco de dados [11].

A proposta apresentada contribui para aprimorar as técnicas de detecção de DDoS ao apresentar um método eficaz e de baixo custo computacional para detecção.

O trabalho está organizado como segue. A próxima seção apresenta o estado atual da pesquisa na área. As Seções III e IV mostram a proposta do trabalho e os resultados alcançados. Finalmente, a Seção V apresenta as conclusões do trabalho e propostas para a sua continuação.

II. Trabalhos Relacionados

Sistemas de detecção de ataques DDoS geralmente operam analisando amostras de tráfego coletados em pontos estratégicos na rede. Os sistemas existentes normalmente são classificados como baseados em assinatura, baseados em anomalia e sistemas híbridos [10].

O uso de técnicas de aprendizado de máquina para a detecção de DDoS têm recebido considerável atenção, pois têm se mostrado capaz de identificar padrões nos ataques, geralmente de difícil gestão para técnicas convencionais. Além disto, estas técnicas podem ser usadas para identificar novas ameaças à medida que surgem, tornando os sistemas de segurança mais eficazes a longo prazo.

Alkasassbeh et al. [12] apresentaram uma revisão bibliográfica sobre algumas técnicas de intrusão, fazendo ainda um estudo comparativo entre três técnicas de detecção de intrusão conhecidas: Perceptron Multicamadas (MLP), Naïve Bayes e Random Forest. Os autores concluíram que o MLP alcançou a maior taxa de precisão, igual a 98,63%.

Hosseini and Azizi [13] propuseram um sistema híbrido com múltiplos algoritmos classificadores, utilizando a base de dados NSL-KDD [14] para avaliar o método apresentado. Foi utilizada uma técnica para dividir a carga computacional entre os lados do cliente e do *proxy*. Segundo os autores, os diferentes ataques existentes possuem comportamentos específicos e por isso, cada algoritmo usa diferentes recursos para se obter um desempenho adequado. Eles ainda afirmaram que a Random Forest foi o algoritmo que obteve melhor resultado entre os algoritmos classificadores estudados. No mesmo ano, foi apresentada uma proposta baseada em uma técnica semi-supervisionada com três algoritmos classificadores que requerem ajustes de parâmetros [15]. Os algoritmos obtiveram uma taxa de precisão superior a 92%, chegando 96,66% no melhor caso, quando utilizada a Random Forest. Em ambas as pesquisas, a análise e avaliação do conjunto de dados foi realizada *offline*.

Em 2019, Lima-Filho [10] propôs um novo sistema de proteção contra ataques DDoS utilizando um sistema distribuído e não invasivo visando combater ataques o mais próximo de sua origem. O sistema é composto por dois subsistema: O primeiro é o sistema de detecção (Smart Detection) que utiliza técnicas de aprendizado de máquina para identificar ataques com base em assinaturas preexistentes. O segundo subsistema é o de proteção (Smart Protection), que tem como objetivo mitigar o tráfego malicioso, aplicando regras para controlar o tráfego indesejado e reduzir os efeitos do DDoS, a partir de conjuntos de dados de referência atualizados e testes semi-controlados em um ambiente laboratorial. Os resultados demonstraram que a solução proposta possui alta precisão na detecção de ataques DDoS, com baixa incidência de alarmes falsos, e é capaz de isolar a ameaça dentro do primeiro minuto de ataque.

A proposta apresentada por Lima-Filho [10] serviu de inspiração para outros dois trabalhos. Silveira [16] propôs o Smart-IoT, que herdou o núcleo da lógica de detecção e de proteção do trabalho original, porém adaptando-as para o cenário de redes IoT, conseguindo contemplar assim também as ameaças que abordam as vulnerabilidades dessas redes, obtendo altas taxas de acertos. No mesmo ano, Freitas-Junior [11] realizou uma análise de desempenho de alguns algoritmos de aprendizagem de máquina para compor o núcleo do módulo de detecção. O autor concluiu que o *Adaboost* foi o algoritmo que apresentou melhores resultados, porém com um tempo de processamento expressivamente superior aos outros algoritmos.

Percebe-se que treinamento de grandes volumes de dados normalmente requer um processamento computacional extenso. Em redes com taxas de dados elevadas, a capacidade de treinamento adaptativo e rápido é fundamental para sistemas de detecção de ataques, pois as restrições temporais impostas para a detecção de ataques exigem respostas rápidas e precisas. Além disso, a detecção de ataques DDoS requer uma análise em tempo real de grandes quantidades de dados de tráfego de rede, o que pode ser desafiador para os algoritmos de inteligência artificial tradicionais. Nesse sentido, as redes neurais podem se destacar como uma alternativa eficiente e promissora para a detecção de ataques DDoS em tempo real, desde que sejam concebidas como arquiteturas de baixa ou média complexidade, capazes de aprender e se adaptar dinamicamente aos padrões de tráfego, bem como processar grandes volumes de dados em tempo hábil.

Neste trabalho, foi utilizada uma rede neural convolucional para detectar padrões em imagens geradas a partir das características do fluxo de tráfego de rede, como aquelas extraídas de campos dos cabeçalhos dos pacotes das camadas de rede e transporte das arquiteturas TCP, UDP e IP. As imagens são montadas a partir de uma organização dos parâmetros

supra-citados em formato matricial. Redes neurais convolucionais têm recebido bastante notoriedade na capacidade de reconhecimento de padrões em imagens. Sua aplicação na área de segurança da informação sob esta óptica ainda é incipiente, de sorte que carece de atenção com relação às suas potencialidades.

A motivação para essa abordagem é a capacidade de treinamento adaptativo e rápido oferecido pelo método. A ideia é criar um modelo de detecção de futuros ataques a partir do treinamento de uma CNN utilizando um banco de imagens geradas a partir do banco de assinaturas de ataques e tráfego normal compilado em [10]. Dessa forma, podemos prever ataques com base nos padrões de imagens geradas a partir do banco de assinaturas. Espera-se que essa abordagem permita uma arquitetura de detecção de ataques DDoS com um melhor equilíbrio entre precisão e tempo de processamento. A abordagem de utilizar imagens geradas a partir das variáveis de tráfego de rede pode trazer novas possibilidades e melhorias para a detecção de ataques DDoS.

Da literatura revisada, o trabalho de Cheng et al. [9] é o que apresenta a solução mais semelhante com a proposta, por utilizar imagens construídas a partir de dados de cabeçalho de quadros de dados como descritores do fluxo da rede. E por utilizar uma rede CNN para classificar ataques de DDoS.

Porém, algumas diferenças podem ser apontadas na comparação entre os dois trabalhos. O trabalho de Cheng et al. [9] utiliza omo descritores matrizes binárias construídas diretamente a partir dos valores de campos do cabeçalho de quadros do fluxo de rede. As imagens possuem dimensões elevadas, variando de de 172×300 pixels a 172×1500 pixels. Além de uma arquitetura de rede CNN complexa, com um número elevado de camadas (da ordem de 10 camadas) e de conexões.

Neste trabalho, foi utilizada a base de dados proposta por Lima-Filho [10], que contém além de variáveis dos campos dos cabeçalhos dos pacotes das camadas de rede e transporte TCP, UDP e IP, medidas estatísticas calculadas a partir dessas variáveis, representando de forma mais abrangente as características dos fluxos das redes analisadas.

As imagens geradas possuem dimensão de apenas 6×6 pixels, bem menor do que no proposto por Cheng et al. [9], o que implica em uma menor demanda de processamento computacional para o treinamento e operação, além de se mostrarem eficientes na classificação de tráfego malicioso de DDoS, permitindo assim o uso de uma arquitetura CNN de baixa complexidade como classificador.

III. Sistema proposto

A descrição do sistema proposto está organizada em cinco tópicos principais: pré-processamento dos dados, criação de imagens, divisão do conjunto de dados, treinamento do modelo e teste do modelo.

A. Banco de dados

O banco de dados usado neste trabalho [10] possui 23.088 instâncias de amostra de tráfego normal, 14.988 instâncias de ataques de inundação TCP, 6.894 instâncias de inundação UDP, 347 instâncias de inundação HTTP e 183 instâncias de ataques HTTP lento, totalizando 45.500 instâncias. O tráfego de rede com um comportamento de atividade legítima foi extraído do conjunto de dados ISCXIDS2012[1] [17].

Cada instância do banco possui um conjunto de 3 rótulos e 73 variáveis que foram derivadas de outras variáveis dos campos dos cabeçalho dos pacotes das camadas de rede e transporte da arquitetura TCP, UDP e IP.

De acordo com Lima-Filho [10], os protocolos amplamente utilizados para a amostragem de tráfego de rede frequentemente utilizam sete variáveis no processo de amostragem, sendo elas: IP de origem, IP de destino, portas de origem, portas de destino, protocolo da camada de transporte, tamanho do pacote IP e sinalizadores TCP.

Embora os endereços IP de origem e destino sejam úteis para identificar os dispositivos, eles não são suficientes para identificar o comportamento do tráfego de rede no ambiente da Internet. Isso reduz o número de variáveis disponíveis para cinco nos casos mais comuns. Com base nessas cinco variáveis e em medidas estatísticas que expressam a variabilidade dos dados no fluxo de rede, Lima-Filho [10] gerou as 29 variáveis listadas na Tabela I, as quais também serão utilizadas neste trabalho.

No processo de cálculo das variáveis do banco de dados, é importante ressaltar que as referências à média, mediana e desvio padrão devem ser interpretadas como medidas amostrais. Além disso, outras medidas estatísticas associadas à entropia, coeficiente quantil e taxa de mudança de valores assumidos pelas variáveis de pacotes amostrados na rede foram calculadas.

B. Pré-processamento e criação das imagens

A etapa de pré-processamento consiste em transformar os dados do conjunto de variáveis do banco de assinaturas em imagens para serem utilizadas como entrada da rede neural, sendo associada uma imagem para cada instância do banco.

Os valores das 29 variáveis, selecionadas anteriormente a partir de cada instância do banco, são normalizados na faixa $[0, 255]$ e organizados em imagens 6×6[2]. As imagens das Figuras 1a e 1b ilustram exemplos de imagens geradas pelo processo, representando, respectivamente, uma instância de ataque e uma instância de tráfego normal.

C. Divisão do conjunto de dados, treinamento e teste do modelo

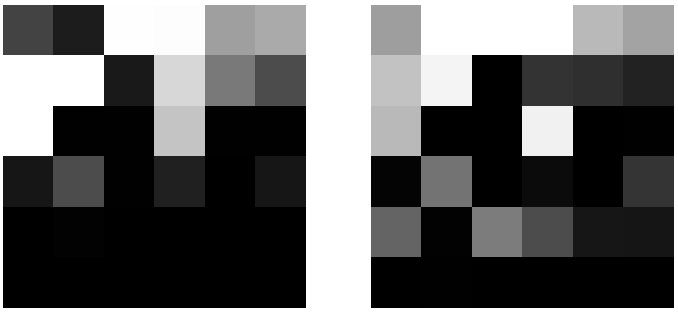
Para garantir que o modelo tenha uma boa capacidade de generalização ele deve capaz de realizar previsões precisas em dados que não foram usados durante o treinamento. Para isso, é comum dividir o conjunto de dados em conjuntos distintos de treinamento, validação e teste. Neste trabalho foi adotada a proporção de 80% das amostras para treinamento, 10% para

[1]ISCXIDS2012 está acessível em: https://www.unb.ca/cic/datasets/ids.html

[2]Com o objetivo de manter imagens de entrada quadradas, optou-se por fazer um preenchimento nulo dos últimos sete pixels de cada imagem 6×6.

Tabela I: Variáveis do banco de assinaturas utilizado para os experimentos.

#	Variável	Detalhes
01	ip_proto	Campo *IP proto* normalizado
02	ip_len_mean	Média do campo *IP length*
03	ip_len_median	Mediana do campo *IP length*
04	ip_len_std	Desvio padrão do campo *IP length*
05	ip_len_entropy	Entropia do campo *IP length*
06	ip_len_cv	Coef. de variação do campo *IP length*
07	ip_len_cvq	Coef. quantil do campo *IP length*
08	ip_len_rte	Taxa de mudança do campo *IP length*
09	sport_mean	Média do campo *Source port*
10	sport_median	Mediana do campo *Source port*
11	sport_std	Desvio padrão do campo *Source port*
12	sport_entropy	Entropia do campo *Source port*
13	sport_cv	Coef. de variação do campo *Source port*
14	sport_cvq	Coef. quantil do campo *Source port*
15	sport_rte	Taxa de mudança do campo *Source port*
16	dport_mean	Média do campo *Destination port*
17	dport_median	Mediana do campo *Destination port*
18	dport_std	Desvio padrão do campo *Destination port*
19	dport_entropy	Entropia do campo *Destination port*
20	dport_cv	Coef. de variação do campo *Destination port*
21	dport_cvq	Coef. quantil do campo *Destination port*
22	dport_rte	Taxa de mudança do campo *Destination port*
23	tcp_flags_mean	Média do campo *TCP flags*
24	tcp_flags_median	Mediana do campo *TCP flags*
25	tcp_flags_std	Desvio padrão do campo *TCP flags*
26	tcp_flags_entropy	Entropia do campo *TCP flags*
27	tcp_flags_cv	Coef. de variação do campo *TCP flags*
28	tcp_flags_cvq	Coef. quantil do campo *TCP flags*
29	tcp_flags_rte	Taxa de mudança do campo *TCP flags*



(a) Ataque. (b) Normal.

Figura 1: Exemplo de imagens representando uma instâncias de tráfego 1a de ataque e tráfego 1b.

testes e 10% para validação, aplicada de forma aleatória sobre o conjunto total de dados.

O conjunto de treinamento é utilizado para ajustar os pesos do modelo durante o treinamento a fim de minimizar o erro na previsão. O conjunto de validação, por sua vez, é utilizado para monitorar o desempenho do modelo durante o treinamento e ajustar os parâmetros do modelo, como taxa de aprendizado e número de épocas, a fim de melhorar a precisão das previsões. Por fim, o conjunto de teste é usado para avaliar o desempenho do modelo final, uma vez que ele foi treinado e ajustado com base nos conjuntos de treinamento e validação.

A rede neural criada para a detecção dos ataques DDoS foi desenvolvida utilizando a biblioteca Keras. A metodologia consistiu em carregar as imagens e seus respectivos rótulos associados às classes: 'ataque' ou 'normal'. O modelo de rede utilizado neste trabalho é composto por duas camadas principais: uma camada de convolução com 32 filtros de tamanho 3x3, ativação ReLU e uma entrada no formato (6,6,1), que significa uma imagem com dimensão 6×6 em escala de cinza (1 canal). Posteriormente, temos uma camada *flatten* que transforma a saída da camada de convolução em um vetor unidimensional para ser processado por uma camada densa com ativação *sigmoid*, que produz uma saída binária. A escolha da função de ativação ReLU é comum em modelos de redes neurais convolucionais, pois permite a aprendizagem de representações mais esparsas e eficientes dos dados. A função de ativação *sigmoid* é utilizada para garantir que a saída esteja no intervalo [0,1], permitindo interpretá-la como a probabilidade de a imagem ser classificada como uma anomalia ou não. A Figura 2 mostra o diagrama do modelo da Rede Neural Convolucional proposto no trabalho. Cada retângulo na imagem representa uma camada. Dentro de cada retângulo, é indicado o tipo de operação realizada por ela, e a descrição da entrada e saída da camada.

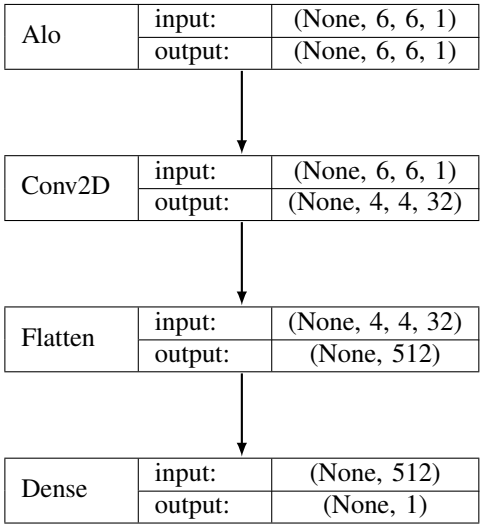


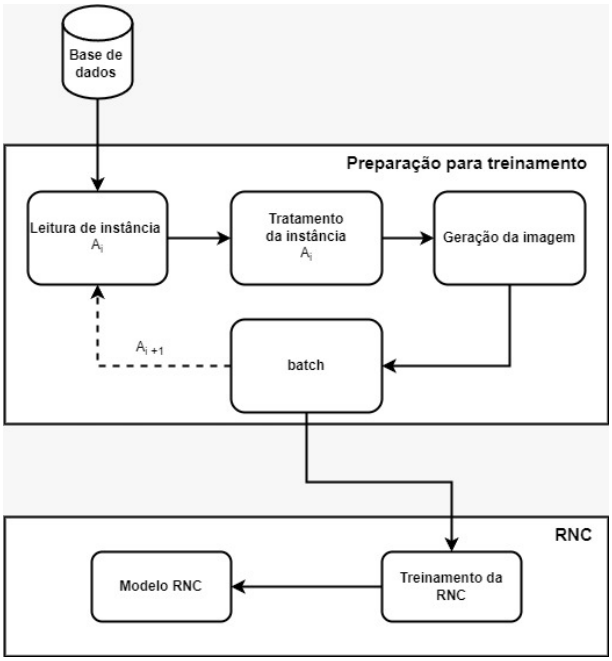
Figura 2: Diagrama do modelo da Rede Neural Convolucional.

Após definir o modelo, é necessário especificar seu otimizador, a função de perda e as métricas a serem utilizadas para avaliar o desempenho do modelo. Neste trabalho, foi utilizado o otimizador Adam, que é uma versão melhorada do gradiente descendente estocástico, e que ajusta as taxas de aprendizado de cada parâmetro da rede neural de forma adaptativa. A função de perda utilizada foi a *binary_crossentropy*, que é adequada para problemas de classificação binária. A métrica de avaliação escolhida foi a acurácia, que é a proporção de imagens classificadas corretamente pelo modelo.

O modelo foi então treinado para minimizar a função de perda segundo o fluxo de processamento ilustrado na Figura 3. O processo inicia com a leitura de uma instância do banco de dados. Em seguida, os valores das métricas que definem essa instância são escalonados em 256 níveis de amplitude e organizados em uma imagem em escalas de cinza 6×6. As

imagens são agrupadas em lotes (*batches*) para o treinamento da rede neural e criação do modelo.

Figura 3: Treinamento da arquitetura proposta.



O treinamento é realizado ao longo de 10 épocas, com um tamanho de lote de 32 imagens. A cada época, foi utilizado o conjunto de validação para validar o desempenho do modelo.

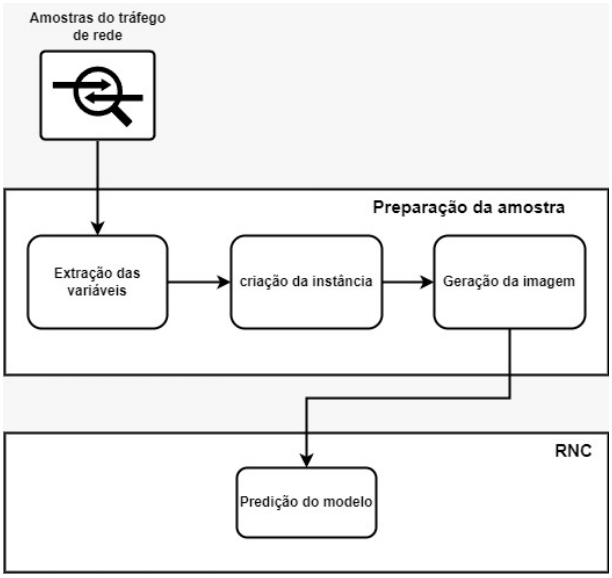
Após o treinamento, o modelo foi submetido ao conjunto de testes para avaliar seu desempenho em dados inéditos. Foram medidos a perda média e a acurácia do modelo. Adicionalmente, foram realizadas predições utilizando o conjunto de testes, permitindo a construção da matriz de confusão e a determinação das métricas de avaliação, incluindo precisão, recall e F1-score. Essas métricas são úteis para avaliar o desempenho do modelo em termos de erros e acertos, fornecendo uma visão detalhada sobre o desempenho do modelo em relação aos falsos positivos, falsos negativos e acertos.

A Figura 4 ilustra como a arquitetura proposta pode ser utilizada em um cenário de operação. O sistema opera coletando amostras do tráfego de dados de dispositivos de rede. Essas amostras contêm pacotes que são recebidos e processados para extrair as variáveis dos cabeçalhos. As métricas estatísticas definidas na Tabela I são calculadas a partir dessas variáveis, e assim gera-se uma nova imagem que funciona como um descritor associado aos dados de rede amostrados. A imagem gerada é então enviada para o modelo de rede neural obtido na etapa de treinamento que a analisa e a classifica como proveniente de tráfego normal ou malicioso.

IV. Resultados e Discussão

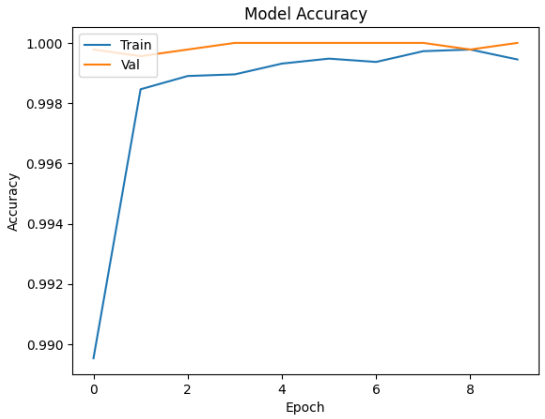
Os resultados obtidos na avaliação do modelo no conjunto de testes foram bastante satisfatórios, com perda média de **0,66%** e acurácia de **99,96%**, o que indica um desempenho excelente na classificação de novos dados.

Figura 4: Cenário de operação da arquitetura proposta.



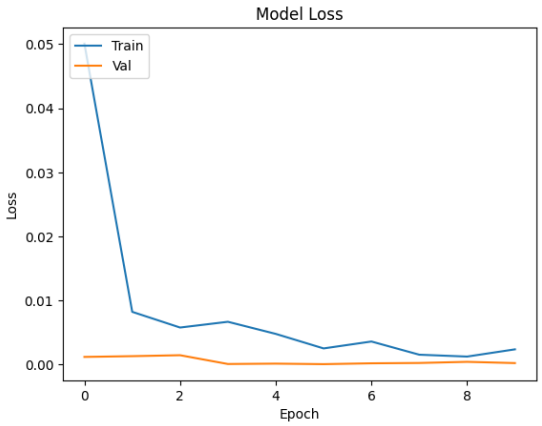
As Figuras 5 e 6 apresentam o gráfico de precisão e o gráfico de perda, respectivamente, que ilustram esses resultados. O gráfico de precisão mostra a taxa de acertos do modelo ao longo das épocas de treinamento, enquanto o gráfico de perda representa a evolução do erro do modelo durante o treinamento. Esses gráficos permitem acompanhar o desempenho do modelo durante o processo de treinamento. Embora ocorram oscilações nas curvas, é possível observar uma tendência geral de melhora na precisão e na perda ao longo das épocas de treinamento. Essa evolução é um indicativo de que o modelo está se ajustando aos dados e se tornando cada vez mais preciso em suas predições.

Figura 5: Gráfico de precisão (Acurácia) durante o treinamento e validação.



Das amostras testadas, a ferramenta desenvolvida acertou a classificação de 2309 amostras negativas e 2239 amostras positivas, cometendo apenas 2 falsos negativos e nenhum falso positivo, conforme visto na matriz de confusão ilustrada na Figura 7. Isso indica que o modelo apresentou um alto grau

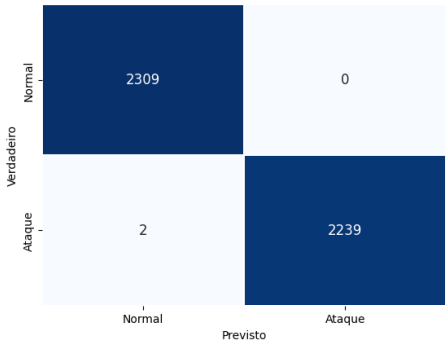
Figura 6: Gráfico de perda durante o treinamento e validação.



de precisão e recall, com valores iguais a 1. A métrica F1-score, também foi calculada e apresentou valor igual a 1, o que reforça a qualidade do modelo na tarefa de classificação.

Em resumo, os resultados obtidos indicam que o modelo de rede neural convolucional foi capaz de aprender as características distintas entre as classes de imagens e generalizar bem para novos dados, apresentando um desempenho excelente na classificação do tráfego da rede.

Figura 7: Matriz de confusão do classificador proposto.



Ao comparar os resultados obtidos com o trabalho anterior de [11], podemos observar um avanço significativo. No estudo anterior, o algoritmo *AdaBoost* foi o que atingiu o melhor resultado para os teste. No melhor dos cenários estudados, apresentou resultados iguais aos aqui apresentados, com as métricas de avaliação da acurácia, precisão, recall e F1-score também apresentando valores iguais a 1, e uma matriz de confusão sem nenhum erro de classificação. No entanto, o uso de recursos computacionais para o *AdaBoost* foi superior, levando cerca de 4 minutos e 20 segundos para selecionar as melhores características e quase 8 minutos para criar o melhor modelo com as características selecionadas anteriormente.

Já no trabalho atual, utilizamos uma rede neural convolucional que, em comparação com o *AdaBoost*, foi muito mais rápida na etapa de treinamento, em torno de 30 segundos. Além disso, a rede neural convolucional utiliza todo o conjunto de características disponíveis para a definição do modelo, tornando-se mais eficiente no processamento dos dados. Essa eficiência se deve à capacidade da rede neural em detectar automaticamente as características relevantes para a tarefa de classificação, enquanto o *AdaBoost* depende da seleção manual de características. Dessa forma, a arquitetura de classificação baseada em rede neural convolucional se mostra uma alternativa mais promissora, tanto em termos de desempenho quanto em custo computacional quando comparada com o *AdaBoost*.

V. Conclusão

Foi apresentado um modelo de rede neural convolucional para a detecção de ataques DDoS em redes de computadores. Os resultados obtidos mostraram que o modelo apresentou um desempenho excelente na classificação de novos dados, com alta precisão, recall e F1-score. Além disso, a comparação com o trabalho anterior destacou a superioridade da rede neural em relação ao algoritmo *AdaBoost*, em termos de tempo de treinamento e eficiência no processamento dos dados, visto que o modelo proposto apresentou um tempo de treinamento mais rápido e maior eficiência na seleção de características relevantes para a tarefa de classificação.

É importante ressaltar que a capacidade da rede neural em realizar um treinamento mais rápido e adaptativo é fundamental para sistemas de detecção de ataques, especialmente em redes com fluxos de dados elevados, onde a detecção deve ocorrer dentro de restrições temporais. Nesse sentido, a utilização da rede neural convolucional como técnica de detecção de ataques DDoS pode ser uma alternativa viável para melhorar a segurança e a eficiência do gerenciamento de redes. Nesse sentido, a utilização de modelos de aprendizado de máquina, como o proposto neste trabalho, pode contribuir para o desenvolvimento de sistemas de segurança mais eficientes e confiáveis.

Por fim, é importante destacar que, embora os resultados obtidos sejam bastante promissores, o trabalho ainda apresenta algumas limitações que devem ser consideradas em futuras pesquisas. Uma delas é a necessidade de verificar se o modelo apresentará o mesmo desempenho ao ser treinado com novos conjuntos de dados. Além disso, o fato de que o trabalho foi realizado de forma offline significa que a aplicação do modelo em um ambiente real e online pode apresentar desafios adicionais, tais como uma necessidade de se ajustar a taxa de amostragem do fluxo de dados em função da taxa de tráfego da rede, e atrasos na detecção de ataques.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Referências

[1] N. Vieira, “Ataques DDoS geram preocupação no setor financeiro,” 12 2019. [Online]. Available: https://canaltech.com.br/seguranca/ataques-ddos-geram-preocupacao-no-setor-financeiro-156861/

[2] Akamai, “A ameaça bate à porta: Análise de ataques a Serviços financeiros,” Akamai, Tech. Rep., 11 2022.

[3] B. Zhang, T. Zhang, and Z. Yu, “DDoS detection and prevention based on artificial intelligence techniques,” in *2017 3rd IEEE International*

Conference on Computer and Communications, ICCC 2017, vol. 2018-January, 2018.

[4] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet of Things (Netherlands)*, vol. 7, 2019.

[5] A. Seifousadati, S. Ghasemshirazi, and M. Fathian, "A Machine Learning Approach for DDoS Detection on IoT Devices," 2021, arXiv 2110.14911.

[6] A. E. Cil, K. Yildiz, and A. Buldu, "Detection of DDoS attacks with feed forward based deep neural network model," *Expert Systems with Applications*, vol. 169, 2021.

[7] A. Saied, R. E. Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using Artificial Neural Networks," *Neurocomputing*, vol. 172, 2016.

[8] A. Portela, W. Costa, and R. Gomes, "Detecção de Ataques DDoS em redes IoT usando Redes Neurais e Seleção de Características," in *Anais Estendidos do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, 2021, pp. 225–232.

[9] J. Cheng, Y. Liu, X. Tang, V. S. Sheng, M. Li, and J. Li, "DDoS attack detection via multi-scale convolutional neural network," *Computers, Materials and Continua*, vol. 62, no. 3, pp. 1317–1333, 2020.

[10] F. S. d. Lima-Filho, "Smart Defender: um sistema de detecção e mitigação de ataques DoS/DDoS usando aprendizagem de máquina," Ph.D. dissertation, UFRN, 2019.

[11] A. A. d. Freitas-Junior, "Um estudo de algoritmos de aprendizagem de máquinas para o Smart Defender," UFRN, Natal, Tech. Rep., 2020.

[12] M. Alkasassbeh, G. Al-Naymat, A. B.A, and M. Almseidin, "Detecting Distributed Denial of Service Attacks Using Data Mining Techniques," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 1, 2016.

[13] S. Hosseini and M. Azizi, "The hybrid technique for DDoS detection with supervised learning algorithms," *Computer Networks*, vol. 158, 2019.

[14] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*, 2009.

[15] M. Aamir and S. M. Ali Zaidi, "Clustering based semi-supervised machine learning for DDoS attack classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 4, 2021.

[16] F. A. F. Silveira, "Smart-IoT: um sistema de proteção contra DDoS para rede de Internet das Coisas," Master's thesis, Universidade Federal do Rio Grande do Norte, Natal, 2020. [Online]. Available: https://repositorio.ufrn.br/handle/123456789/30831

[17] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers and Security*, vol. 31, no. 3, 2012.