Detecção de Ataques *SlowLoris* usando *Perceptron* de Múltiplas Camadas

Dalbert Matos Mascarenhas e Rafael Saraiva Campos

***Resumo*— Os ataques de negação de serviço (DoS - *Denial of Service*) têm criado desafios relacionados a detecção e contenção dos mesmos. A dificuldade de detecção deve-se ao uso de vulnerabilidades específicas por parte destes ataques. O presente trabalho apresenta uma ferramenta baseada em aprendizado de máquina, utilizando uma rede neural artificial *feed-forward* - o *perceptron* de múltiplas camadas (MLP - *Multi-Layer Perceptron*) - para detectar um tipo específico de ataque DoS, o *SlowLoris*. Variáveis extraídas de *traces* de tráfego TCP e HTTP são empregadas para treinar o MLP, que atua como um classificador não-linear binário, atribuindo a cada pacote TCP o flag 0 (tráfego normal) ou 1 (ataque). De um total de 34694 pacotes, igualmente divididos entre as duas classes de tráfego, 75% foram empregados para o treinamento e validação do MLP, sendo o restante usado para teste. Na fase de teste validação cruzada foi empregada para aumentar a confiabilidade dos resultados, e a solução proposta atingiu acurácias de classificação acima de 99.6%.**

***Palavras-Chave*— TCP, SlowLoris, rede neural artificial *feed-forward*, perceptron de múltiplas camadas.**

***Abstract*— Denial of service attacks (DoS) have created challenges regarding its detection and containment. This detection difficulty stems from the use of specific vulnerabilities by these attacks. This work presents a machine learning based tool that uses a feed-forward artificial neural network - the multilayer perceptron (MLP) - to detect a specific type of denial attack, the *Slowloris* attack. Variables extracted from TCP and HTTP traffic traces are applied to train the MLP, which acts as a non-linear binary classifier, associating a flag 0 (normal traffic) or 1 (attack) to each TCP packet. From a total of 34694 packets, equally split between these two classes, 75% were used for training and validation of the MLP, while the remainder were used to test the MLP. In the on-line test phase, cross-validation was used to improve the confidence of the yielded results, and the proposed solution achieved classification accuracies between 99.6% and 99.9%.**

***Keywords*— TCP, SlowLoris, feed-forward artificial neural network, multilayer perceptron.**

I. Introdução

Atualmente ataques de negação de serviço (DoS - *Denial of Service*) têm representado uma grande ameaça para os recursos disponibilizados na Internet e nas redes internas. Este tipo de ataque tem como objetivo impossibilitar o acesso de usuários legítimos a recursos da rede [1]. O ataque é executado através da exploração de vulnerabilidades do alvo ou da exaustão de seus recursos. Estes recursos podem ser divididos em recursos da rede ou a performance de serviços de aplicação [2], [3]. Parte dos ataques de DoS são direcionados a servidores de aplicação que mantêm serviços como WEB e e-mail. Grande parte dos ataques DoS fundamentam-se em estabelecer um grande número de conexões TCP abertas ou semiabertas no hospedeiro alvo [4], [5]. Por consequência, a máquina provedora do serviço perde a habilidade de aceitar requisições legítimas devido ao grande número de conexões em espera. Além disso, recursos como memória e processamento também são consumidos durante um ataque. Desta forma, o desafio criado por ataques de DoS em relação ao consumo de serviços e conteúdos tem fomentado estudos que promovem a detecção e a redução dos danos causados por esse tipo de ataque.

Dalbert Matos Mascarenhas e Rafael Saraiva Campos, Engenharia de Computação, CEFET/RJ, Petrópolis, Emails: dalbert.mascarenhas@cefet-rj.br e rafael.campos@cefet-rj.br

Parte dos ataques de DoS podem ser segmentados em ataques que exploram protocolos da camada de Aplicação e são classificados como ADoS (*Application Layer DoS*) [6]. Dois tipos de ataques ADoS podem ser destacados: *Flooding* e *LowRate*. A utilização do primeiro tipo de ataque produz um excessivo fluxo de tráfego com o objetivo de consumir os recursos da aplicação, até que a mesma fique incapaz de atender novas requisições. O segundo tipo consiste na geração de tráfego semelhante ao de requisições legítimas, utilizando vulnerabilidades encontradas em protocolos como o HTTP e HTTPS. Este tipo de ataque tenta manter as requisições em espera por tempo indeterminado sem a utilização de um alto volume de dados [7]. Os ataques do tipo *LowRate*, como o *Slowloris*, consomem recursos apenas de serviços alvos sem afetar outros recursos disponíveis no hospedeiro.

O ataque *Slowloris* explora o protocolo HTTP e é capaz de tornar um servidor WEB indisponível com baixa sobrecarga de tráfego [8], [9]. Este tipo de ataque apresenta uma característica semelhante a de tráfego legítimo. O funcionamento consiste em enviar requisições HTTP GET HEADER incompletas no final do pacote, sempre em um intervalo de tempo, de modo a forçar suas renovações. Dessa forma, o servidor não finaliza as requisições, mantendo-as no *pool* de atendimento até o valor de *timeout* pré-configurado no servidor [10]. Os recursos são consumidos pelas requisições maliciosas, a medida em que o ataque prossegue realizando as solicitações e mantendo-as abertas.

Parte das soluções providas na literatura para o ataque *Slowloris* utilizam estratégias como limitar o número de conexões para cada usuário ou determinar *timeouts* para cada conexão [3], [11]. Estas limitações podem impedir que usuários legítimos mantenham conexões que superam um limite pré-estabelecido. Este limite pode ser ultrapassado quando usuários acessam páginas com um alto número de objetos em modo de conexão não-persistente do HTTP. Além disso, a limitação do *timeout* provoca a desconexão de usuários que estão com conexões ociosas.

Este trabalho apresenta uma solução baseada em aprendi-

zado de máquina para a detecção de pacotes TCP oriundos de ataques *SlowLoris*. Uma rede neural artificial (RNA) do tipo *feed-forward* - o *perceptron* de múltiplas camadas (MLP - *Multilayer Perceptron*) é treinada usando um subconjunto dos parâmetros disponíveis em *logs* do tráfego de rede que podem ser coletados no servidor ou em equipamentos intermediários, como roteadores com porta em modo promíscuo ou IPS (*Intrusion Prevention System*) em modo *stealth*. O MLP atua como um classificador não-linear binário, classificando cada pacote TCP como normal ou malicioso, i.e., oriundo de um ataque *SlowLoris*. Um endereço IP de origem com tráfego consistentemente classificado como malicioso pode ser então bloqueado no *firewall*.

A organização do trabalho é descrita a seguir. A Seção II apresenta os trabalhos relacionados referentes a estratégias de defesa contra ataques *SlowLoris*. A Seção III descreve o cenário de análise. A Seção IV apresenta a solução proposta. Os resultados dos experimentos são descritos na Seção V. A Seção VI traz uma breve conclusão.

II. Trabalhos Relacionados

Sousa *et al.* analisam duas ferramentas de IDS (Intrusion Detection System) para realizar a detecção do ataque *Slowloris* [12]. A primeira ferramenta, denominada Suricata, não gera um número adequado de alertas para que um ataque *Slowloris* seja detectado. Enquanto a segunda ferramenta, denominada SNORT [13], [14] realiza a detecção do ataque gerando uma avaliação de memória e processamento. Pascoal *et al.* apresentam um módulo de defesa embasado em uma estratégia seletiva denominada SeVen [6], [8]. Este módulo utiliza funções de probabilidade para definir, dentre as novas requisições, quais serão aceitas ou rejeitadas quando o servidor WEB encontra-se saturado. Quando uma nova solicitação chega à aplicação, SeVen precisa verificar se há disponibilidade no *pool* de conexões do servidor. Caso o mesmo tenha sido esgotado, a estratégia determinará probabilisticamente qual das conexões estabelecidas deverá ser encerrada para que a nova possa ser atendida. Porém, neste caso, conexões de usuários legítimos também podem ser encerradas.

Durcekova *et al.* abordam técnicas de detecção de ataque DoS na camada de aplicação baseadas em assinatura e anomalias [15]. No caso de técnicas que se baseiam em assinatura realiza-se um monitoramento de mudanças estatísticas, selecionando primeiramente um parâmetro do tráfego de entrada. A detecção do ataque ocorre se o tráfego inspecionado apresentar características familiares de uma atividade maliciosa. No entanto esta abordagem apresenta limitações, dada a facilidade existente de variação de tráfego de dados. No caso do *SlowLoris*, como seu tráfego de rede é similar ao de conexões legítimas, uma conexão pertencente a um usuário comum poderia ser confundida com uma originada por um atacante. Já a técnica baseada em anomalia identifica um ataque se seu tráfego de dados não apresentar um comportamento similar ao de um tráfego especificado como normal a partir de dados de treinamento. Um grande desafio para esta abordagem encontra-se na determinação de todos os tipos de comportamentos normais segundo os dados de treinamento.

Correa *et al.* propõem uma solução para o ataque *Slowloris* usando o *software (D)DoS Deflate* [3]. O *(D)DoS Deflate* é um *script* desenvolvido em *Shell Bash* e utilizado para verificar as conexões que ultrapassam o limite definido para os usuários e consequentemente bloquear o IP referente a essas conexões. O bloqueio pode ser feito utilizando regras de IPTABLES, ou pelo *software Advanced Policy Firewall* (APF) [16]. Porém, esta limitação pode restringir o acesso de usuários legítimos a páginas com múltiplos objetos em conexão persistente ou restringir clientes que estivessem usando o servidor através de NAT (*Network Address Translation*).

O trabalho de Faria *et al.* propõe um IPS para a detecção do ataque *Slowloris* em redes internas [17]. Os autores analisam os *traces* buscando por semelhanças de ataques e em caso positivo o atacante pode ser bloqueado. A ferramenta atua por meio de três módulos distintos, distribuídos nos roteadores, servidores WEB e em um concentrador. Os módulos atuam em conjunto para detectar e conter o ataque através da análise do padrão de comportamento do ataque. Para a detecção do ataque os autores utilizaram uma análise estatística do comportamento do ataque, sem empregar técnicas de inteligência artificial.

Diferentemente do que é apresentado nos trabalhos relacionados, a ferramenta proposta neste trabalho não limita o número de conexões e também não estabelece *timeout* de conexões. Estes comportamentos podem elevar consideravelmente o número de falsos positivos. Além disso, o presente trabalho utiliza aprendizado de máquina, através de redes neurais artificiais *feed-forward*, também denominadas MLPs, para detectar o ataque com maior grau de acurácia. Após a fase de treinamento, a ferramenta é capaz de responder com eficiência a um tráfego anômalo oriundo de um ataque *Slowloris*.

A literatura apresenta diversos trabalhos que empregam técnicas de aprendizado de máquina para a detecção de ataques DDos, como maquinas de vetores de suporte [18] e redes neurais artificiais [19], [20], [21]. Em [22], os autores usam técnicas de aprendizado de máquina para gerar uma árvore de decisão para a detecção efetiva de ataques DDos, reportando uma acurácia superior a 98%. O trabalho de Barati *et al.* utiliza redes neurais e algoritmos genéticos para detectar ataques de DDoS [23], alcançando uma taxa de falsos positivos de 0,002% na identificação de ataques high-rate gerados por pacotes direcionados às camadas de rede e transporte. Saiedi *et al.* apesentam uma proposta baseada em redes neurais para detectar ataques e filtrá-los utilizando nós sensores [24]. Apesar de atingir uma acurácia de 98%, a ferramenta utilizada apenas detecta ataques de DDoS nos protocolos TCP, UDP e ICMP. No entanto, vale ressaltar que o escopo do presente trabalho é o ataque low-rate direcionado a camada de aplicação e especificamente ao protocolo HTTP.

III. Cenário de Análise

O cenário de análise assume um ataque direcionado a um servidor WEB. A Fig. 1 apresenta um exemplo de *logs* de tráfego colhidos durante um ataque *SlowLoris*. Os *logs* de tráfego foram coletados na máquina servidora. Também foram coletados pacotes assumindo que apenas conexões legítimas

foram estabelecidas com servidor WEB, conforme ilustra a Fig. 2.

```
3.616749795,192.168.70.51,TCP,74,GET / HTTP/1.1  [TCP segment of a reassembled PDU]
3.820743235,192.168.70.51,TCP,296,[TCP Retransmission] 51092  >  80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=230
3.828757712,192.168.70.51,TCP,296,[TCP Retransmission] 51098  >  80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=230
3.828768136,192.168.70.51,TCP,296,[TCP Retransmission] 51096  >  80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=230
3.828770591,192.168.70.51,TCP,304,[TCP Retransmission] 51094  >  80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=238
3.828773012,192.168.70.51,TCP,296,[TCP Retransmission] 51100  >  80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=230
4.268790220,192.168.70.51,TCP,304,[TCP Retransmission] 51094  >  80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=238
4.268808896,192.168.70.51,TCP,296,[TCP Retransmission] 51100  >  80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=230
4.268811373,192.168.70.51,TCP,296,[TCP Retransmission] 51098  >  80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=230
4.268813476,192.168.70.51,TCP,296,[TCP Retransmission] 51096  >  80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=230
4.268815712,192.168.70.51,TCP,296,[TCP Retransmission] 51092  >  80 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=230
4.428496988,192.168.70.51,TCP,74,[TCP Retransmission] 51102  >  80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
```

Fig. 1. Exemplo de um *trace* colhido de um ataque *SlowLoris*.

```
44.085477645,192.168.70.24,HTTP,259,GET /index.html HTTP/1.1 ,33818,80
44.085491071,192.168.70.22,TCP,66,53818  >  80 [ACK] Seq=1029 Ack=2289463 Win=184832 Len=0
44.085711617,192.168.70.22,TCP,66,53818  >  80 [ACK] Seq=1029 Ack=2292359 Win=184832 Len=0
44.086207918,192.168.70.22,TCP,66,53818  >  80 [ACK] Seq=1029 Ack=2296703 Win=184832 Len=0
44.086457796,192.168.70.22,TCP,66,53818  >  80 [ACK] Seq=1029 Ack=2299599 Win=184832 Len=0
44.086708451,192.168.70.22,TCP,66,53818  >  80 [ACK] Seq=1029 Ack=2302495 Win=184832 Len=0
44.086958389,192.168.70.39,TCP,66,54858  >  80 [ACK] Seq=1029 Ack=37868271 Win=997760 Len=0
44.087227804,192.168.70.39,TCP,66,54858  >  80 [ACK] Seq=1029 Ack=37871167 Win=997760 Len=0
```

Fig. 2. Exemplo de um *trace* colhido com navegações legítimas

O ataque *SlowLoris* tem como característica a abertura de uma série de conexões incompletas em um servidor WEB. A ideia do ataque consiste em manter as conexões abertas durante um longo período de tempo e consequentemente esgotar os recursos relacionados a conexão do servidor WEB. Um exemplo do poder do ataque pode ser visto na Fig. 3, que apresenta o trafego destinado a um servidor WEB e que após o início dos ataques, por volta de 20 s, fica restrito a um baixo trafego de pacotes devido a ocupação dos recursos do servidor pelo ataque. Para manter as conexões estabelecidas, diversos pacotes do tipo *Reassembled PDU (Protocol Data Unit)* são enviados.

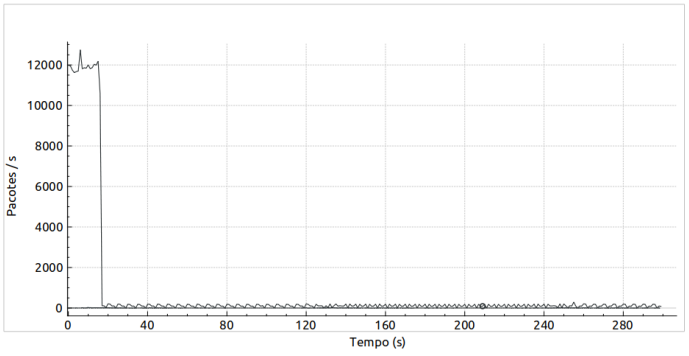

Fig. 3. Exemplo do comportamento do tráfego de pacotes durante um ataque *SlowLoris*.

Observando o comportamento apresentado no cenário descrito, os *logs* de tráfego foram filtrados escolhendo as informações mais relevantes para a identificação do ataque. Os filtros foram usados para selecionar os campos:

- Tamanho do pacote em bytes (*Length*);
- Porta TCP de origem (*SrcPort*);
- Flags de Conexão (ACK,SYN,PSH);
- Reassembled PDU (*rePDU*);
- GET;
- Retransmissão TCP (*reTCP*);
- Tamanho de Janela TCP (*WinSize*);
- Número de Sequência do Segmento TCP (*Seq*);

O campo Flags de Conexão pode apresentar diferentes valores a depender da fase de comunicação. O valor SYN é utilizado no início de uma conexão TCP e também na primeira resposta de uma conexão de três vias em conjunto com o valor ACK. O valor ACK também é utilizado para garantir a confiabilidade na transmissão de segmentos TCP. O valor PSH é utilizado para dar prioridade no processamento do segmento TCP. O campo Reassembled PDU indica que o pacote é parte de uma sequência contígua de segmentos TCP que compõem a reconstrução de uma informação completa dos dados da aplicação. Apenas a última parte desta sequência não será marcada como Reassembled PDU. O campo GET informa se o segmento TCP está carregando uma requisição WEB. O campo Tamanho de Janela indica o tamanho da janela do protocolo TCP. O campo Retransmissão TCP informa se houve uma retransmissão de segmento TCP.

IV. Solução Proposta

Este trabalho emprega uma rede neural artificial (RNA) do tipo *feed-forward*, também denominada *perceptron* de múltiplas camadas (MLP), para identificar pacotes TCP gerados durante um ataque *SlowLoris*.

RNAs são sistemas paralelos distribuídos compostos por unidades de processamento denominadas neurônios, que executam funções matemáticas, denominadas funções de ativação ou de transferência [25]. RNAs são usualmente empregadas para aproximar funções não-lineares e em problemas de classificação.

A Fig. 4 ilustra a topologia do MLP empregado no problema de classificação tratado neste trabalho. Há 10 nós de entrada, um para cada variável de entrada (descritas na Seção III). As variáveis de entrada são binárias, excetuando as duas primeiras (*Length* e *SrcPort*) e as duas últimas (*WinSize* e *Seq*). Estas variáveis são normalizadas para o intervalo $[0; 1]$ antes de serem apresentadas à rede. O MLP tem uma camada escondida com 10 neurônios. As conexões em vermelho representam os potenciais de ativação dos neurônios (*biases*). Os nós em vermelho (nós *bias*) tem entrada fixa igual a -1. Todos os neurônios (10 na camada escondida e 1 na camada de saída) têm função de ativação sigmóide, definida por $f(x) = \frac{1}{1+\exp(-x)}$.

O MLP é empregado como um classificador binário, com saída 0 para pacotes oriundos de tráfego normal e 1 para aqueles gerados por um ataque *SlowLoris*. Como a função sigmóide tem saída contínua no intervalo $(0; 1)$, faz-se necessário arredondar o *output* do neurônio da camada de saída para 0 ou 1.

O algoritmo de aprendizado supervisionado empregado é o *backpropagation* com gradiente descendente. A taxa de aprendizado é 0.45, e o número máximo de épocas é igual a 100.

V. Resultados Experimentais

Validação cruzada foi empregada para aumentar a confiabilidade dos resultados, tendo sido executadas 10 *runs*. Em cada *run*, de um total de 34694 pacotes, igualmente divididos entre as duas classes de tráfego (normal e ataque), 75% são

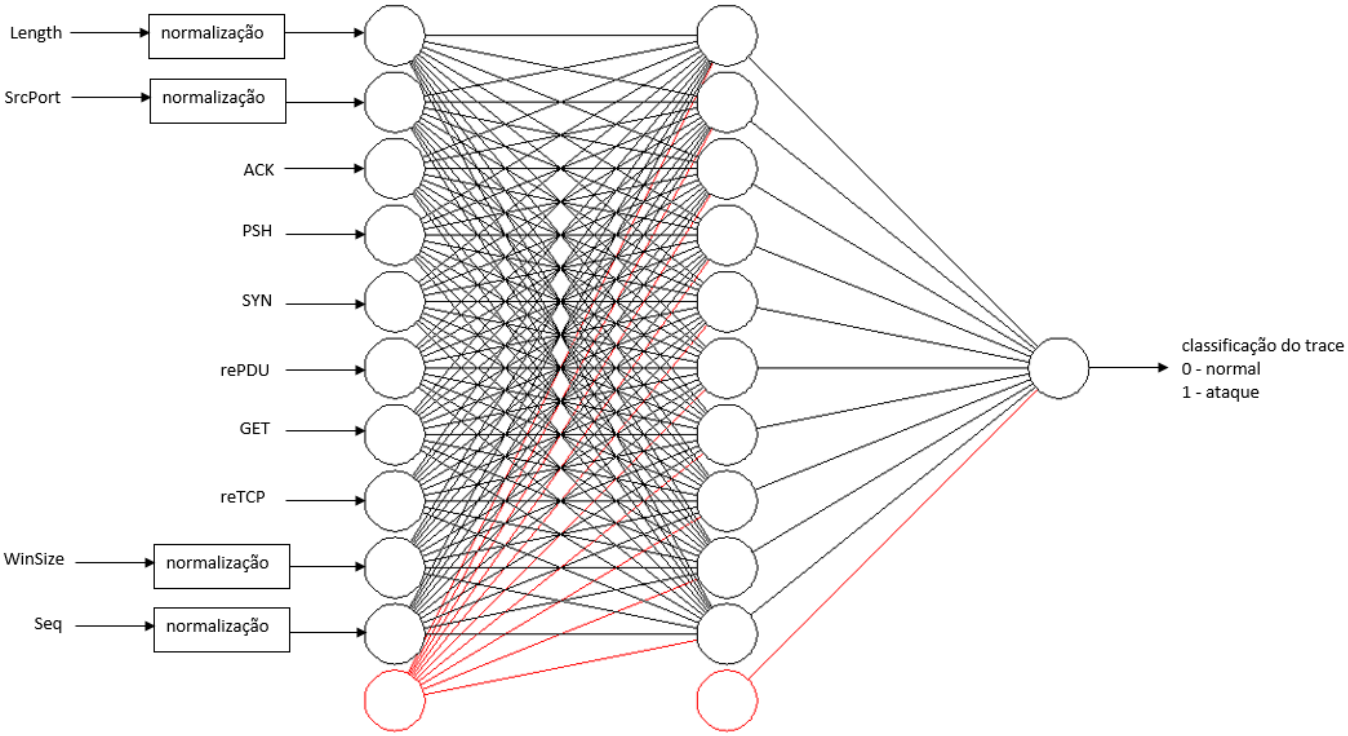

Fig. 4. Topologia do MLP.

randomicamente selecionados para o treinamento da rede, dos quais 30% são separados para a validação da rede durante o treinamento, de modo a prevenir o *overfitting* (i.e., a perda da capacidade de generalização). A Fig. 5 exibe a acurácia (i.e., percentual de classificações corretas) alcançada por *run*. Em todos os casos, a acurácia foi superior a 99.5%.

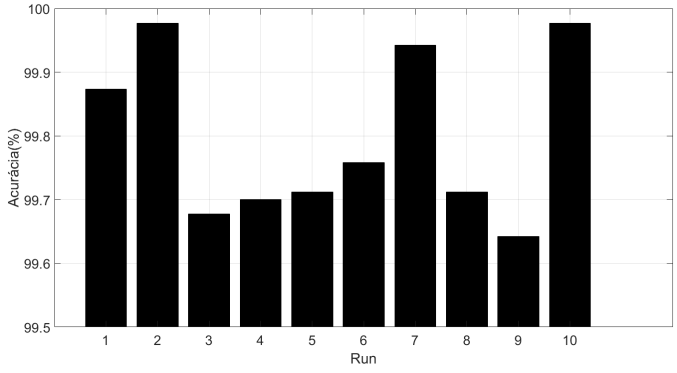

Fig. 5. Acurácia do classificador ao longo de 10 *runs* usando validação cruzada.

Matrizes de confusão são tipicamente usadas para avaliar a performance de classificadores supervisionados [26]. A Tabela I ilustra a matriz de confusão do classificador, considerando os resultados de todas as *runs* de validação cruzada em conjunto. Cada coluna representa as instâncias da classe predita (i.e., aquela fornecida na saída do MLP) e cada linha indica as instâncias da classe real (i.e., a classe alvo).

Seja a matriz quadrada $\mathbf{M} = [m_{l,c}]_{l,c=1,\ldots,N_c}$ a matriz de confusão do classificador, onde $N_c = 2$ é o número de classes. A partir desta matriz, os seguintes contadores podem ser definidos:

TABELA I
MATRIZ DE CONFUSÃO DO CLASSIFICADOR.

	Classe Predita	
Classe Alvo	**Normal**	**Ataque**
Normal	43168	150
Ataque	26	43396

- *Verdadeiros Positivos* (VP): o número de vetores de entrada pertencentes à k-ésima classe e que são corretamente classificados como tal, i.e., $\text{VP}_k = \sum m_{k,k}$, $k = 1, \ldots, N_c$;
- *Verdadeiros Negativos* (VN): o número de vetores de entrada que não pertencem à k-ésima classe e são corretamente classificados como tal, i.e., $\text{VN}_k = \sum m_{l,c}$, for $l, c = 1, \ldots, N_c$ e $l, c \neq k$;
- *Falsos Positivos* (FP): o número de vetores de entrada que não pertencem a k-ésima classe e que são incorretamente identificados como pertencentes a ela, i.e., $\text{FP}_k = \sum m_{l,k}$, $l = 1, \ldots, N_c$ and $l \neq k$;
- *Falsos Negativos* (FN): o número de vetores de entrada pertencentes à k-ésima classe e que são incorretamente classificados como não pertencentes a ela, i.e., $\text{FN}_k = \sum m_{k,c}$, $c = 1, \ldots, N_c$, $c \neq k$.

Três métricas, construídas a partir dos supracitados contadores, foram selecionadas para a avaliação de performance do classificador: *acurácia*, que é o percentual de classificação corretas; *precisão*, que é o percentual de classificações positivas corretas; *especificidade*, que é o percentual de classificações

negativas corretas. A acurácia, precisão e especificidade da k-ésima classe, $k = 1, 2$, são definidas pelas equações (1), (2) e (3), respectivamente.

$$\text{Acurácia}_k = \frac{\text{VP}_k + \text{VN}_k}{\text{VP}_k + \text{VN}_k + \text{FP}_k + \text{FN}_k} \quad (1)$$

$$\text{Precisão}_k = \frac{\text{VP}_k}{\text{VP}_k + \text{FP}_k} \quad (2)$$

$$\text{Especificidade}_k = \frac{\text{VN}_k}{\text{VN}_k + \text{FP}_k} \quad (3)$$

A Tabela II exibe os valores de VP, VN, FP, FN, acurácia, precisão e especificidade para o classificador proposto, considerando as 10 *runs* de validação cruzada.

TABELA II
PERFORMANCE DO CLASSIFICADOR.

	Classe Predita	
	Normal (0)	**Ataque (1)**
VP	43168	43396
FP	150	26
FN	26	150
VN	43396	43168
Acurácia	99.8%	99.8%
Precisão	99.6%	99.9%
Especificidade	99.6%	99.9%

A Tabela II mostra que o classificador proposto alcança elevada acurácia, precisão e especificidade para ambas as classes de tráfego.

VI. CONCLUSÃO

Este trabalho explorou o uso de aprendizado de máquina para a identificação de ataques do tipo *SlowLoris*. Uma rede neural artificial do tipo *feed-forward* foi treinada de forma supervisionada com variáveis extraída de dezenas de milhares de *logs* de tráfego de pacotes HTTP. A rede neural atuou como um classificador binário, atribuindo a cada pacote um identificador 0 (tráfego normal) ou 1 (ataque *SlowLoris*). Validação cruzada foi empregada para aumentar a confiabilidade dos resultados. O classificador neural treinado atingiu acurácia em torno de 99.8%.

REFERÊNCIAS

[1] C. A. M. da Silva, J. A. Gonçalves, V. da Silva Faria, G. de Britto Vieira, and D. M. Mascarenhas, "IREMAC: Um IPS para ataques internos," in *XXXIV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, 2016.

[2] Q. Gu and P. Liu, "Denial of service attacks," *Handbook of Computer Networks: Distributed Networks, Network Planning, Control, Management, and New Trends and Applications*, vol. 3, pp. 454–468, 2007.

[3] A. L. R. Corrêa and H. P. Martins, "Monitoramento de ataques de negação de serviço: Um caso prático utilizando slowloris," *Faculdade de Tecnologia de Bauru (FATEC)*, 2013.

[4] J. Goncalves, V. Faria, G. Vieira, C. Silva, and D. M. Mascarenhas, "WIDIP: Wireless distributed IPS for DDoS attacks," in *2017 1st Cyber Security in Networking Conference (CSNet)*. IEEE, 2017, pp. 1–3.

[5] H. Yuan, Y. Xia, H. Yang, and Y. Yuan, "Resilient control for wireless networked control systems under DoS attack via a hierarchical game," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 15, pp. 4604–4623, 2018.

[6] T. Pascoal, J. Correa, R. Brayner, V. Nigam, and I. Fonseca, "Módulo de proteção contra ataques de negação de serviço na camada de aplicação: uma análise de qualidade de serviço e experiência de usuário," in *XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. SBC, 2017.

[7] S. Toklu and M. Şimşek, "Two-layer approach for mixed high-rate and low-rate distributed denial of service (DDoS) attack detection and filtering," *Arabian Journal for Science and Engineering*, vol. 43, no. 12, pp. 7923–7931, 2018.

[8] Y. G. Dantas, V. Nigam, and I. E. Fonseca, "A selective defense for application layer DDos attacks," in *2014 IEEE Joint Intelligence and Security Informatics Conference*. IEEE, 2014, pp. 75–82.

[9] T. Shorey, D. Subbaiah, A. Goyal, A. Sakxena, and A. K. Mishra, "Performance comparison and analysis of Slowloris, GoldenEye and Xerxes DDoS attack tools," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2018, pp. 318–322.

[10] N. Tripathi and N. Hubballi, "Slow rate denial of service attacks against HTTP/2 and detection," *Computers & security*, vol. 72, pp. 255–272, 2018.

[11] R. Papadie and I. Apostol, "Analyzing websites protection mechanisms against DDoS attacks," in *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. IEEE, 2017, pp. 1–6.

[12] T. E. de Sousa Araújo, F. M. Matos, and J. A. Moreira, "Intrusion detection systems' performance for distributed denial-of-service attack," in *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. IEEE, 2017, pp. 1–6.

[13] D. Day and B. Burns, "A performance analysis of SNORT and Suricata network intrusion detection and prevention engines," in *Fifth International Conference on Digital Society, Gosier, Guadeloupe*, 2011, pp. 187–192.

[14] E. Albin, "A comparative analysis of the SNORT and Suricata intrusion-detection systems," NAVAL POSTGRADUATE SCHOOL MONTEREY CA, Tech. Rep., 2011.

[15] V. Durcekova, L. Schwartz, and N. Shahmehri, "Sophisticated denial of service attacks aimed at application layer," in *2012 ELEKTRO*. IEEE, 2012, pp. 55–60.

[16] S. B. Sangeetha, "DDoS deflate and APF (advanced policy firewall): A report," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 27, no. 2, September 2015.

[17] V. Faria, J. A. G. C. A. Silva, G. B. Vieira, and D. M. Mascarenhas, "Ferramenta para detecção e contenção de ataques Slowloris," in *XIX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2019)*, 2019.

[18] S. K. Q. Liao, H. Li and C. Liu, "Application layer DDos attack detection using cluster with label based on sparse vector decomposition and rhythm matching," *Security and Communication Networks*, vol. 8, no. 17, pp. 3111–3120, 2015.

[19] Y. L. J. Li and L. Gu, "DDos attack detection based on neural network," in *2nd International Symposium on Aware Computing (ISAC)*, 2010.

[20] R. Karimazad and A. Faraahi, "An anomaly-based method for DDos attacks detection using RBF neural networks," in *Proceedings of the International Conference on Network and Electronics Engineering*, 2011.

[21] L. M. Ibrahim, "Anomaly network intrusion detection system based on distributed time-delay neural network (DTDNN)," *Journal of Engineering Science and Technology*, vol. 5, no. 4, pp. 457–471, 2010.

[22] M. Zekri, S. E. Kafhali, N. Aboutabit, and Y. Saadi, "DDoS attack detection using machine learning techniques in cloud computing environments," in *2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, 2017.

[23] M. Barati, A. Abdullah, N. I. Udzir, R. Mahmod, and N. Mustapha, "Distributed denial of service detection using hybrid machine learning technique," in *2014 International Symposium on Biometrics and Security Technologies (ISBAST)*. IEEE, 2014, pp. 268–273.

[24] A. Saied, R. E. Overill, and T. Radzik, "Detection of known and unknown ddos attacks using artificial neural networks," *Neurocomputing*, vol. 172, pp. 385–393, 2016.

[25] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, 1994.

[26] F. J. P. et. al., "The case against accuracy estimation for comparing induction algorithms." in *Proceedings of 15th International Conference on Machine Learning (ICML-98)*, vol. 98, 1998, pp. 445–453.