Machine Translated by Google

Um sistema de detecção de intrusão para várias classes Classificação baseada em Redes Neurais Profundas

Petros Toupas, Dimitra Chamou, Konstantinos M. Giannoutakis, Anastasios Drosou, Dimitrios Tzovaras Instituto de Tecnologias da Informação
Centro de Pesquisa e Tecnologia-Hellas
6º km Charilaou-Thermi, 57001, Thessaloniki, Grécia
{ptoupas,dimicham,kgiannou,drosou,Dimitrios.Tzovaras} @iti.gr

Resumo—Os Sistemas de Detecção de Intrusão (IDSs) são considerados um dos elementos fundamentais na segurança da rede de uma organização, pois formam a primeira linha de defesa contra ameaças cibernéticas, sendo responsáveis por efetivamente uma potencial invasão na rede. Muitas implementações de IDS usam análise de tráfego de rede baseada em fluxo para detectar possíveis ameaças. A pesquisa de segurança de rede é um campo em constante evolução e os IDSs em particular têm sido o foco dos últimos anos com muitos métodos inovadores propostos e desenvolvidos. Neste artigo, propomos um modelo de aprendizado profundo, mais especificamente uma rede neural composta por várias camadas totalmente conectadas empilhadas, a fim de implementar um IDS de detecção de anomalias baseado em fluxo para classificação multiclasse.
Usamos o conjunto de dados CICIDS2017 atualizado para fins de treinamento e avaliação. O resultado experimental usando MLP para sistema de detecção de intrusão mostrou que o modelo proposto pode alcançar resultados promissores na classificação multiclasse com relação à precisão, revocação (taxa de detecção) e taxa de falso positivo (taxa de alarme falso) neste conjunto de dados específico.

Termos do Índice — Cibersegurança, Sistema de Detecção de Intrusão, Redes Neurais Profundas, CICIDS2017, Baseado em Recursos de Fluxo, Classificação Multiclasse

I. INTRODUÇÃO

Durante os últimos anos, a crescente exposição de muitas organizações a ataques cibernéticos sofisticados levou a um rápido desenvolvimento de IDSs inovadores. O desenvolvimento de IDSs preocupa tanto a comunidade acadêmica quanto a industrial em todo o mundo, devido ao impacto que cada ataque cibernético tem, como custo econômico, danos reputacionais e consequências legais.
Portanto, é de grande importância proteger as redes contra acessos não autorizados e proteger a comunicação do usuário e seus dados, [1], bem como revelar novos problemas de segurança que surgem.

A. Sistema de Detecção de Intrusão

O Intrusion Detection System (IDS) é uma eficiente ferramenta de reforço de segurança para a detecção e proteção de ataques cibernéticos em qualquer rede ou host. A responsabilidade dos IDSs é detectar comportamentos suspeitos e agir adequadamente para proteger a rede do início de ataques e reduzir perdas funcionais e financeiras, [2].

Na literatura, os IDSs podem ser categorizados como, [3], baseados em assinaturas, [4], baseados em anomalias, [5] ou uma combinação híbrida de ambos.

Sistemas de detecção de intrusão baseados em assinatura (SIDS), também conhecidos como IDS baseados em regras ou uso indevido, conduzem o monitoramento contínuo do tráfego de rede e buscam sequências ou padrões de tráfego de rede de entrada que correspondam a uma assinatura de ataque. Uma assinatura de ataque pode ser identificada com base em cabeçalhos de pacotes de rede, destinos ou endereços de rede de origem; sequências de dados que correspondem a malware conhecido ou outros padrões, sequências de dados ou séries de pacotes que se sabe estarem associados a um determinado ataque. Eles trabalham com altas taxas de precisão na identificação de possíveis invasões conhecidas, mantendo baixas as taxas de erro. No entanto, o banco de dados do sistema deve ser atualizado manualmente pelo administrador e o SIDS pode detectar apenas invasões que existam no banco de dados do sistema, excluindo a detecção de novos ataques (zero-day-attack), pois não há padrão de assinatura de ataque relevante no sistema banco de dados.

Os sistemas de detecção de intrusão baseados em anomalias (AIDS), ou detecção baseada em comportamento, analisam o comportamento normal das redes, monitorando o tráfego da rede para detectar atividades anormais. Os AIDS têm a capacidade de serem treinados com algoritmos de detecção de anomalias ou de serem autotreinados com algoritmos de autoaprendizagem, para que possam detectar novos tipos de invasões. Comparado com o baseado em assinatura, o baseado em anomalia mostra uma diferença significativa na identificação de novos ataques. Além disso, o perfil de configuração de cada sistema pode ser personalizado, tornando difícil para os invasores descobrir quais atividades de intrusão não serão detectadas [6].

O Sistema de Detecção de Intrusão Híbrido (HIDS) pode combinar as vantagens do sistema baseado em assinatura e no sistema baseado em anomalia e aumentar a detecção de ataques de intrusão conhecidos, enquanto elimina as taxas de erro de ataques desconhecidos. A maioria dos IDSs híbridos mais recentes é baseada em métodos de aprendizado profundo e de máquina.

Devido às vantagens dos sistemas de detecção de intrusão baseados em anomalias no campo de ataques de dia zero, o modelo proposto desenvolve um Sistema de Detecção de Intrusão baseado em anomalia que é baseado em aprendizado profundo.

B. Classificação baseada em recursos de fluxo

Um dos principais métodos de detecção de intrusão é a análise do tráfego de rede e a extração dos recursos estatísticos desejados para detectar o tráfego de rede anormal, quase em tempo real. Assim, a classificação do tráfego é um componente central na

Machine Translated by Google

um Sistema de Detecção de Intrusão, que ao analisar pacotes de rede, pode determinar se o comportamento da rede viola a segurança do sistema, monitorando continuamente a rede.
IDSs, a fim de funcionar corretamente e detectar atividades anormais de forma eficaz, usam pacotes de tráfego divididos em fluxos de rede, de acordo com IP de origem/destino, porta de origem/destino, protocolo e timestamp, [7], [8]. Uma definição de fluxo útil é mencionada abaixo. Um fluxo é um grupo de pacotes IP com algumas propriedades comuns passando por um ponto de monitoramento em um intervalo de tempo especificado [9].

Cisco, [10], referiu que Um fluxo completo é uma troca unidirecional de pacotes consecutivos na rede entre uma porta em um endereço IP e outra porta em um endereço IP diferente, usando um determinado protocolo de aplicação.

Portanto, a classificação do tráfego é necessária para o gerenciamento eficiente do fluxo, processamento e exploração do aprendizado de máquina, [11]. Em geral, as categorias de classificação de tráfego mais difundidas e amplas estão usando diferentes recursos de fluxo e são divididas em métodos baseados em porta, métodos baseados em carga útil, métodos baseados em host e métodos baseados em recursos de fluxo.

Os sistemas de detecção de intrusão aplicam diferentes métodos de detecção de anomalias, dependendo de cada estudo de caso, dos recursos disponíveis e das tecnologias acessíveis. O trabalho atual se concentra na técnica baseada em recursos de fluxo, uma vez que pode superar inúmeras limitações de outras técnicas, como números de porta não registrados, payload de pacotes criptografados etc.
O método baseado em fluxo usa recursos de fluxo como discriminadores para explorar a diversidade dos pacotes de tráfego e mapear fluxos para classes, [11]. Além disso, em relação às questões de privacidade, o método baseado em fluxo é preferível ao método de carga útil, devido à ausência de carga útil.

A classificação de tráfego baseada em fluxo é realizada com alto grau de precisão, usando técnicas de aprendizado de máquina, e ocupa uma grande área de pesquisa. Boutaba et al., [11] referiram que o classificador MLP-NN discriminativo pode atingir mais de 99% de precisão para classificação de tráfego com fluxos. Além disso, Sperotto et al. [12] fornecem uma pesquisa abrangente de detecção de intrusão baseada em fluxo.

C. Técnicas de aprendizado de máquina e aprendizado profundo para IDS As

técnicas de aprendizado de máquina e aprendizado profundo têm sido usadas para desenvolver IDSs no campo da segurança cibernética. A fim de aumentar a eficácia dos IDSs, a pesquisa tem se concentrado em novas tecnologias de aprendizagem e algoritmos de Redes Neurais Artificiais (ANN), Máquinas de Vetores de Suporte (SVM), Naive-Bayesianas (NB), Florestas Aleatórias (RF), auto-organização mapa (SOM) etc. Na verdade, o aprendizado de máquina consiste em um conjunto de algoritmos para tirar conclusões usando métodos matemáticos e estatísticos. O uso generalizado de aprendizado de máquina se estende aos campos de previsão, classificação e estimativa, especialmente no campo da segurança de rede.

A necessidade de um conjunto de dados completo, rico, atualizado e bem formado com vários critérios e recursos é uma preocupação fundamental dos pesquisadores para a condução de experimentos, testes e avaliação dos modelos, [13], [14] em redes modernas. Um conjunto de dados é apropriado quando:

- é atualizado a tempo devido à alta mutação de malware e evolução
- representa o tráfego de rede do mundo real • tem diversidade e volume de tráfego • é desejável que esteja publicamente disponível Na literatura,

existem vários conjuntos de dados disponíveis para experimentação, mas apenas alguns atendem a todos os recursos desejados. Convém referir alguns dos mais conhecidos, nominalmente, com poucos detalhes.

KDD-99, [15], CAIDA, [16], ISCX2012, [17], Kyoto, [18] são conjuntos de dados que representam o tráfego de rede do mundo real, mas atualmente são considerados desatualizados devido à contínua evolução dos ataques e ameaças à rede . Por outro lado, um bastante popular no campo de pesquisa é o conjunto de dados CICIDS2017, que foi lançado recentemente e contém muitos tipos de tráfego, atende aos critérios do tráfego de rede do mundo real e foi criado para superar alguns problemas dos conjuntos de dados existentes.

Com base em Gharib et al., [19], o CICIDS2017 atende a todos os critérios de um conjunto de dados preciso e completo. No entanto, uma questão que precisa ser abordada neste conjunto de dados é que há um desequilíbrio de alta classe que pode enganar o classificador, [13]. Atualmente, surgiu um novo conjunto de dados denominado CIC AWS 2018, [20], similar a sua versão anterior, CICIDS2017, porém ainda pouco divulgado.

Com base no conjunto de dados CICIDS2017, nosso objetivo foi implementar um sistema de intrusão de rede de anomalias usando dados estatísticos baseados em fluxo que detectam e classificam com alta precisão cada tipo de ataque em multiclassificação. Classificação de fluxo obtida com o uso de métodos de aprendizado profundo no campo de aprendizado de máquina. Usamos aprendizado supervisionado e um Multi-Layer Perceptron (MLP). Projetamos um modelo de rede neural profunda aprimorado para classificar o tráfego de rede.

Portanto, este artigo propõe um sistema de detecção de anomalias baseado em fluxo usando a abordagem de aprendizado profundo.

A estrutura deste artigo é a seguinte. A Seção II descreve alguns dos trabalhos relacionados à detecção de invasões de rede usando técnicas de aprendizado de máquina e aprendizado profundo, principalmente no conjunto de dados CICIDS2017. A Seção III explica o conjunto de dados usado para a análise e descreve o procedimento de pré-processamento. Na Seção IV, fornecemos uma breve descrição da arquitetura da rede neural profunda, os algoritmos e nosso sistema de detecção de anomalias baseado em fluxo MLP proposto. Na Seção V, analisamos os resultados experimentais do MLP e discutimos as soluções propostas. Por fim, a Seção VI apresenta a conclusão deste trabalho e apresenta trabalhos futuros.

II. TRABALHO RELACIONADO

Na literatura recente, a maioria dos estudos em Sistemas de Detecção de Intrusão baseados em fluxo baseados em tecnologias de aprendizado de máquina está usando o conjunto de dados CICIDS2017, para treinamento e avaliação. No entanto, devido ao novo conjunto de dados no campo da segurança cibernética, os estudos publicados ainda são limitados.

Ullah e Mahmou, [21], propuseram um modelo híbrido de detecção de anomalias, usando tecnologias de detecção de anomalias baseadas em fluxo para a classificação nos conjuntos de dados CICIDS2017 e UNSW 15. Eles usaram a eliminação recursiva de recursos (RFE)

Machine Translated by Google

para a seleção de características significativas, Synthetic Minority Over-Sampling Technique (SMOTE) para a sobreamostragem e Edited Nearest Neighbors (ENN) para a limpeza dos conjuntos de dados CICIDS2017 e UNSW-15, a fim de serem balanceados.
No nível 1 os fluxos da rede foram classificados com o classificador de árvore de decisão, como normal ou anormal (classificação binária) e, em seguida, encaminhados para o nível 2 para determinar o tipo de ataque (multiclassificação). Os resultados de especificidade, precisão, recall e F-score para nível 1 foram medidos 100% para o conjunto de dados CICIDS2017 e 99% para o conjunto de dados UNSW-15, enquanto a precisão, recall e F-score do modelo de nível 2 foram medido em 100% para o conjunto de dados CICIDS2017 e 97% para o conjunto de dados UNSW-15, respectivamente.

Vijayanand, Devaraj e Kannapiran, [22], propuseram um novo IDS com seleção de recursos baseada em algoritmo genético e vários classificadores de máquinas de vetores de suporte para redes mesh sem fio. Para obter melhor precisão, eles selecionam recursos específicos explorando a seleção de recursos baseada em Algoritmo Genético e o classificador SVM. A avaliação do sistema é feita usando um conjunto de dados de intrusão, gerado a partir de uma WMN, e simulado na ferramenta Network Simulator 3 (NS3) usando o conjunto de dados de intrusão padrão. Além disso, eles validam o sistema usando conjuntos de dados de intrusão ADFA-LD e CICIDS2017.

Uma análise comparativa é realizada, entre o sistema proposto e a seleção de recursos baseada em MI, sugerindo que a seleção de recursos baseada em AG com classificador SVN apresenta melhores métricas de desempenho, com maior precisão, cerca de 99%, e menor complexidade computacional.

Zhang, et al., [2], apresentou um modelo de detecção de anomalias baseado em rede neural. Eles projetaram um IDS usando a rede neural convolucional LeNet 5 e a rede LSTM para extração de recursos. Os experimentos foram conduzidos usando os conjuntos de dados CICIDS2017 e CTU para classificação binária e múltipla. Eles realizaram CNN, LSTM e a combinação híbrida de ambos, que obtiveram bons resultados de classificação em experimentos de classificação binária e multiclassificação. A precisão conseguida foi de cerca de 99%. Eles também analisaram os fluxos que foram importantes para a classificação e para a detecção anormal eficiente.

Ferrag e Maglaras, [23], apresentaram o DeepCoin, que é um novo aprendizado profundo e uma estrutura de energia baseada em blockchain para proteger a rede inteligente contra ataques cibernéticos. O algoritmo de rede neural recorrente do algoritmo prático de tolerância a falhas bizantino foi usado para a rede baseada em blocos usando aprendizado profundo. Eles trabalharam em três conjuntos de dados diferentes para fins de avaliação e teste de desempenho, incluindo CICIDS2017, um conjunto de dados de sistema de energia e um conjunto de dados de robô da web (Bot)-Internet of Things. A taxa de acerto usando redes neurais recorrentes, com retropropagação ao longo do tempo foi de 98,23%.

Binbusayyis e Vaiyapuri, [24], focaram principalmente na criação de um ensemble para seleção de recursos usando diferentes medidas de avaliação, que pode implementar um sistema de detecção de intrusão.
Em particular, eles propuseram um conjunto de seleção e extração de recursos e desenvolveram um modelo IDS usando o algoritmo de aprendizado, Random Forest. A avaliação foi feita em vários conjuntos de dados de avaliação, ou seja, KDDCup'99, NSL KDD, UNSW-NB15 e CICIDS2017, a fim de demonstrar a eficácia do modelo proposto. Os resultados revelaram que o subconjunto específico de recursos é promissor devido às métricas finais de alto desempenho, atingindo 99,88% de precisão, em comparação com outras abordagens.

Radford, Richardson e Davis, [25], apresentaram um modelo de sequência de detecção de anomalias baseado na rede neural recorrente (RNN) Long Short-term Memory (LSTM). Eles usaram sequências embutidas passadas por dois modelos LSTM bidirecionais para implementar o sistema proposto. Os experimentos de teste foram conduzidos com o uso do CICIDS2017 e os resultados do modelo visavam a multiclassificação.

Ahmim et al., [26], propuseram um novo IDS usando três classificadores diferentes de aprendizado de máquina. Eles usaram o algoritmo REP Tree e JRip para classificação binária e a saída deles usada como entrada para o Forest PA para classificação múltipla de ameaças cibernéticas. Os experimentos conduzidos no conjunto de dados CICIDS2017 foram comparados com alguns classificadores conhecidos (J48, Jrip, Naive Bayes, MLP, REP Tree, Random Forest, FURIA, LIBSVM, J48 Consolidated, Forest PA, WISARD).

As métricas de desempenho mostraram que eles alcançaram taxa de precisão de 96,66%, taxa de detecção de 94,47% e taxa de alarme falso de 1,14%, onde descobriu-se que seu modelo tinha um desempenho melhor e aprimorado do que o restante dos classificadores.

Idhammad, Afdel e Belouch, [27], implementaram um sistema distribuído de detecção de intrusão para ambientes Cloud. Primeiramente, eles usaram o modelo Naive Bayes para detecção de anomalias e pré-processamento de dados e depois, para a multiclassificação, eles usaram um classificador baseado em Random Forest, que detecta o tipo de cada ataque. Os experimentos foram conduzidos no conjunto de dados CICDDS-001 com métricas de alto desempenho, como taxa de precisão geral de 97,05% e taxa de falsos positivos de 0,21%.

III. PRÉ-PROCESSAMENTO E ANÁLISE DE DADOS

A. Descrição do Conjunto de Dados CICIDS2017

Este trabalho está contando com um conjunto de dados de detecção de intrusão público chamado CICIDS2017, [14], criado pela University of New Brunswick (UNB) em cooperação com o Canadian Institute for Cybersecurity (CIC). O conjunto de dados CICIDS2017 não apenas contém os cenários de ataque de rede mais atualizados, mas também atende a todos os critérios de ataques cibernéticos do mundo real.

O conjunto de dados contém tráfego de rede benigno (normal) e anormal (diferentes tipos de ataques) de cinco dias consecutivos de captura e é dividido em 8 arquivos diferentes. Para cada dia, um tipo diferente de ataque foi implantado, conforme mostrado na Tabela I. Unimos os 8 arquivos em um único arquivo contendo todo o conjunto de dados e nosso trabalho foi baseado nesse arquivo. O número de instâncias/exemplos no arquivo mesclado é igual a 2830743, e a distribuição de todas as 15 classes é mostrada na Tabela II.

Cada linha no conjunto de dados contém 83 recursos que têm foram extraídos do tráfego de rede usando a ferramenta CICFlowMe ter, [28], [29]. O CICFlowMeter gera Fluxos Bidirecionais (Biflow), onde o primeiro pacote determina o encaminhamento (da origem ao destino) e o retorno (do destino à origem)

Machine Translated by Google

TABELA I

TRÁFEGO DIÁRIO DO CONJUNTO DE DADOS CICIDS2017

Dia	Tipo de Tráfego
Segunda-feira	Benigno (normal)
Terça-feira	Benigno, FTP-Patator, SSH-Patator
Quarta-feira	Benigno, Dos GoldenEye, Dos Hulk, DoS Slowhttptest, Dos slowloris, coração sangrando
Quinta-feira	Benigno, Ataque na Web - Força Bruta, Ataque na Web - Injeção de SQL, Ataque na Web - XSS, Infiltração
Sexta-feira	Benigno, Bot, PortScan, DDoS

direções, portanto, os 83 recursos estatísticos incluem dados que derivam da direção direta e reversa. Usamos um subconjunto dos 83 recursos originais, omitindo alguns recursos como IPs de origem e destino, o ID do Biflow, timestamp etc., e acabamos com um conjunto de dados de 79 recursos onde o 79º é o rótulo, denotando que tipo de o tráfego é descrito no Biflow atual.

TABELA II

DISTRIBUIÇÃO DE CLASSES DO CONJUNTO DE DADOS CICIDS2017

(ANTES DO PRÉ-PROCESSAMENTO)

EU IA	Rótulo	Número de Instâncias	% wrt o número total de instâncias
1	BENIGNO	2273097	80,3
2	DoS Hulk	231073	8,16
	PortScan	158930	5,61
4	DDoS	128027	4,52
5	Dos GoldenEye	6 10293	0,36
	FTP-Patator	7 7938	0,28
	SSH-Patator	5897	0,21
8	DoS lentoris	5796	0,20
9	DoS teste http lento	5499	0,19
10	robôs	1966	0,07
11	Ataque na Web - força bruta	1507	0,05
12	Ataque na Web - XSSName	652	0,02
13	Infiltração	36	0,0012
14	Ataque na Web - Injeção SQL	21	0,0007
15	coração sangrando	11	0,0004

B. Limpeza de dados

Os algoritmos de aprendizado de máquina estão diretamente relacionados aos dados e, para serem o mais precisos possível, esses dados precisam ser refinados. Em primeiro lugar, identificamos linhas/bifluxos no conjunto de dados com valores ausentes, valores infinitos e valores que não faziam sentido (ou seja, duração de tempo negativa de uma comunicação, etc.). Existem muitas maneiras de lidar com valores ausentes/errados em um conjunto de dados, como substituir por média/mediana/moda da coluna (recurso), usar um modelo de regressão para prever e substituir esses valores, omitir toda a linha/exemplo que contém os valores ausentes e ainda mais. Esta etapa é crucial para manter a confiabilidade do conjunto de dados e não adicionar ruído, portanto a escolha do método deve ser feita com cautela. No nosso caso, a maioria das linhas com valores ausentes/errados estavam em classes com muitos exemplos (BENIGN, DoS Hulk, PortScan, DDoS), então decidimos removê-los, pois já tínhamos exemplos suficientes para trabalhar.

Além disso, analisamos e extraímos algumas estatísticas (desvio padrão, variância, média, etc.) para cada uma das características de forma independente. Descartamos as feições com variância zero (ou seja, feições com valor constante para todos os exemplos), pois não poderiam fornecer informações estatísticas adicionais, [30], para que nosso algoritmo de ML pudesse “aprender” com essas feições.

Também realizamos o teste de correlação de Pearson, [31]–[33] nas demais características para avaliar as associações entre elas. Se dois ou mais recursos são altamente correlacionados, isso implica que eles estão medindo a mesma informação subjacente, portanto, remover um não deve comprometer o desempenho do modelo e pode até levar a melhores resultados.
O coeficiente de correlação de Pearson ou **r** de Pearson é a métrica que mede a correlação linear entre duas variáveis e seu valor está em [ÿ1, 1]. Removemos os recursos em que essa métrica estava acima ou abaixo do limite de 0,95 ou -0,95, respectivamente.

Por fim, verificamos e excluímos todas as linhas/bifluxos duplicados. Como resultado dos métodos de limpeza e extração de recursos acima, acabamos com um conjunto de dados de 2515416 exemplos e 45 recursos em que a 45ª coluna é o rótulo.

C. Transformação de dados

Decidimos mesclar três das classes de conjunto de dados em uma classe comum maior. Essas classes são Web Attack-Brute Force, Web Attack-XSS e Web attack Sql Injection. Nós os mesclamos porque seu comportamento no nível de tráfego de rede é quase idêntico (algo que também foi confirmado pelos resultados de vários modelos de ML diferentes durante a fase de avaliação). A distribuição final das classes no conjunto de dados limpo é mostrada na Tabela III.

Antes de começar a alimentar os dados limpos em nosso algoritmo de ML, fizemos mais algumas análises estatísticas plotando e inspecionando visualmente a distribuição de cada recurso. Conseguimos fazer isso porque tínhamos um número relativamente pequeno de recursos e pudemos inspecioná-los um por um. Durante esse processo, notamos que muitos dos recursos estavam altamente distorcidos (principalmente à esquerda). A assimetria é a assimetria em uma distribuição estatística, na qual a curva aparece inclinada para a esquerda (assimétrica negativamente) ou para a direita (assimétrica positivamente). Há muitas maneiras de lidar com a assimetria de dados, como raiz cúbica, raiz quadrada, transformação de logaritmo ou quadrado, cubo ou uma transformação de maior potência. Embora esses métodos funcionem em muitos casos, em nosso caso usamos a transformação Yeo-Johnson, [34] porque funcionou melhor para nosso algoritmo de ML.

Yeo-Johnson é uma família de transformações apropriada para reduzir a assimetria e aproximar a normalidade.
Estende de alguma forma a funcionalidade da Caixa original

Machine Translated by Google

TABELA III
DISTRIBUIÇÃO DE CLASSES DO CONJUNTO DE DADOS CICIDS2017
(APÓS O PRÉ-PROCESSAMENTO)

EU IA	Rótulo	Número de Instâncias	% wrt o número total de instâncias
1	BENIGNO	2089692	83,07
2	DoS Hulk 3	172838	6,87
	PortScan	128008	5,08
4	DDoS	90694	3,6
5	Dos GoldenEye 6	10283	0,41
	FTP-Patator 7	5931	0,23
	SSH-Patator	5385	0,21
8	DoS lentoris	5228	0,20
9	DoS teste http lento	3219	0,12
10	Ataques na Web	2143	0,085
11	bot	1948	0,08
12	Infiltração	36	0,0014
13	coração sangrando	11	0,0004

& Cox, [35] transformações, que é válida apenas para valores positivos, para poder usar tanto valores negativos quanto positivos.
As transformações de Yeo-Johnson são definidas da seguinte forma:

$$\ddot{y}(\ddot{y}, x) = \begin{cases} \ddot{y}\ ((y + 1)\ddot{y}\ \ddot{y}\ 1)/\ddot{y} & \text{se } \ddot{y} = 0,\ y\ \ddot{y} \\ 0\ \log(y + 1) \text{ se } \ddot{y} = 0,\ y\ \ddot{y}\ 0\ \ddot{y}[(\ddot{y}y + 1)2\ddot{y}\ \ddot{y} \\ 1)]/(2\ \ddot{y}\ \ddot{y}) \text{ se } \ddot{y} = 2,\ y < 0 \text{ se } \ddot{y} = 0,\ y < 0 \\ \ddot{y}\ \log(\ddot{y}y + 1) \end{cases} \quad (1)$$

Na Figura 1 apresentamos o efeito da transformação de Yeo-Johnson nas distribuições de alguns dos recursos
no conjunto de dados CICIDS2017. A transformação de Yeo-Johnson também normaliza os dados, então não precisamos fazer essa etapa explicitamente.

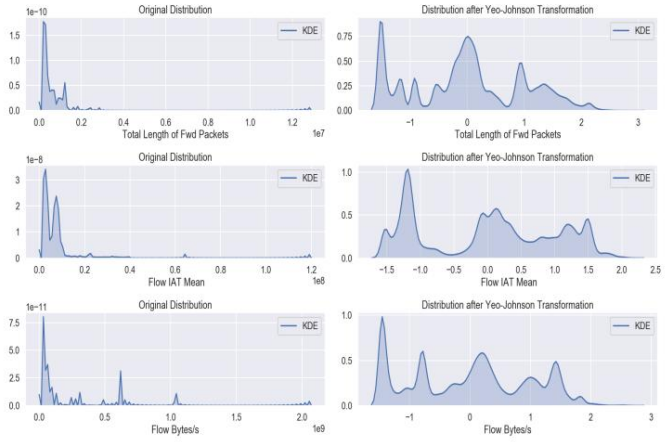


Fig. 1. Transformação de Yeo-Johnson

Olhando para a Tabela III, alguém pode notar o alto desequilíbrio de classe no conjunto de dados. Existem algumas classes (ou seja, Heartbleed, Infiltration e Bot) com poucos exemplos.
Pode ser muito difícil para um algoritmo de ML ser capaz de "aprender" como mapear os recursos de entrada de tais classes para seus rótulos correspondentes, pois não há muitos dados para aprender de. Uma das maneiras mais diretas de lidar com o desequilíbrio de classe é alterar as distribuições de classe para uma distribuição mais equilibrada. distribuição. Existem dois métodos básicos para balancear distribuições de classes. Under-sampling, ou seja, eliminando exemplos da classe majoritária, e Over-sampling, ou seja, replicando exemplos da classe minoritária.

Existem muitas maneiras de subamostrar e sobreamostrar um conjunto de dados desbalanceado, onde algumas das mais comuns são descritas por Gustavo Batista et al. em suas obras, [36] e, [37]. No nosso caso, usamos o método SMOTEENN, que é uma combinação do conhecido Synthetic Minority Over-Sampling Technique (SMOTE), [38] método para sobreamostrar a classe minoritária (no nosso caso havia mais de uma classe minoritária) , seguido pelo método Edited Nearest Neighbor (EEN), [39] para sub-amostragem não apenas da classe majoritária, mas de todas as classes como um método de limpeza de dados.

Ao usar métodos de sobreamostragem, alguém deve ter muito cuidado com o processo de avaliação. Se a sobreamostragem for realizada antes da divisão do conjunto de dados em conjuntos de treinamento, desenvolvimento e teste, a avaliação não será confiável, pois o conjunto de teste foi mesclado com novos dados criados artificialmente e sua distribuição será diferente da original . A maneira correta de proceder é dividir o conjunto de dados e, em seguida, usar o método de sobreamostragem apenas no conjunto de treinamento. Dessa forma, o algoritmo de ML obtém mais dados para treinamento, mas os conjuntos de desenvolvimento e teste permanecem intocados e confiáveis para ajuste fino e avaliação do modelo.

4. ARQUITETURA DE REDE NEURAL PROFUNDA

Como etapa anterior à implementação da rede neural, dividimos os dados em 3 partes. Os 80% dos dados foram usados para o treinamento (conhecido como conjunto de treinamento), 10% foram usados no processo de desenvolvimento e ajuste de hiperparâmetros (conhecido como dev ou conjunto de validação) e a última parte que também é 10% foi usado para fins de teste (conjunto de teste).

Depois de muita experimentação e várias arquiteturas diferentes, chegamos à arquitetura mostrada na Figura 2. Na arquitetura proposta para a rede neural profunda, o modelo consiste em uma camada de entrada que possui 44 recursos passados como entrada para a rede neural rede como as que surgiram da engenharia de recursos descrita no capítulo III.
A camada de entrada é seguida por 8 camadas ocultas com 140, 120, 100, 80, 60, 40, 20 e 120 nós, respectivamente. A camada final é a camada de saída ou camada softmax, que produz as probabilidades para as 13 classes onde ocorre a previsão.

Para a inicialização dos pesos em todas as camadas Dense (Fully Connected) usamos a inicialização lecun-uniforme, [40], enquanto para a camada de saída/softmax usamos a inicialização uniforme glorot, [41]. Surgimos usando o ReLU, [42] como função de ativação para todas as camadas densas, depois de testar diferentes funções de ativação como tanh, sigmoid, selu etc., uma vez que produziu os melhores resultados em comparação com as funções de ativação observadas anteriormente. Para treinar a rede neural, testamos muitos otimizadores como descida de gradiente estocástico, RMSProp, Adagrad, mas finalmente usamos o otimizador Adam, [43]

Machine Translated by Google

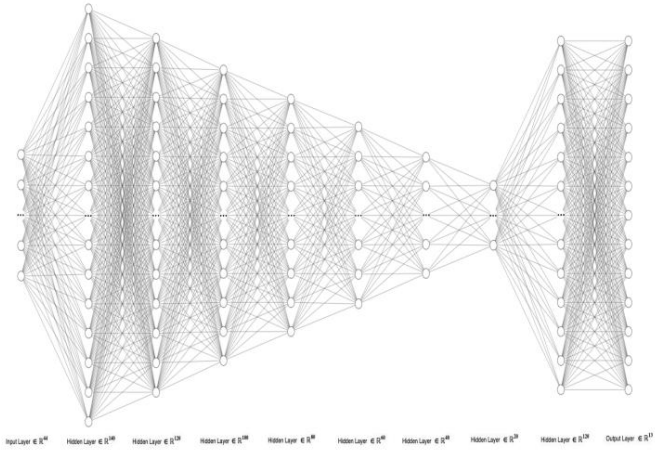


Fig. 2. Arquitetura de rede neural

uma vez que produziu um modelo mais robusto e com melhores resultados durante a avaliação. Embora as técnicas de regularização como regularização L1, L2 e dropout sejam usadas com bastante frequência para lidar com o overfitting em redes neurais, em nosso caso não houve diferença nos resultados finais, então decidimos não adicionar nenhum tipo de regularização ao nosso modelo.

V. RESULTADOS NUMÉRICOS

Neste capítulo apresentamos os resultados da arquitetura proposta em termos de recall, precisão e pontuação F1 para cada uma das 13 classes diferentes que nosso modelo é capaz de detectar. A avaliação do modelo foi baseada em uma validação cruzada de 10 vezes. Em cada uma das 10 divisões, escolhemos 90% dos dados como conjunto de treinamento e 10% como conjunto de teste, enquanto o conjunto de teste em cada uma das 10 divisões é único e nunca se sobrepõe a nenhum outro conjunto de teste em qualquer outra divisão . Para produzir um único valor final para cada métrica específica em cada classe, calculamos a média dos resultados de cada uma das 10 divisões dessa métrica, para finalmente obter os resultados da avaliação para o nosso modelo.

As métricas que usamos para avaliar nosso modelo em cada uma das 10 divisões de validação cruzada são todas baseadas na matriz de confusão que cada uma das divisões produziu. As métricas são as seguintes:

$$\text{Precisão} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precisão} = \frac{PT}{TP + PF}, \quad \text{Recuperar} = \frac{PT}{PT + FN}$$

$$\text{Pontuação F1} = \frac{2 \ddot{y} TP}{2 \ddot{y} TP + FP + FN}$$

$$\text{Taxa P ositiva F alsa} = \frac{PF}{FP + TN}$$

Para cada uma das métricas, calculamos a média (macro) das 10 divisões da validação cruzada para ser o mais robusto e preciso possível na avaliação do nosso modelo.

Os resultados (média de 10 splits) de Precision, Recall, F1 são apresentados na Figura 3. Com base nos resultados da Figura 3 também calculamos as médias sobre todas as classes do modelo. A precisão geral do nosso modelo é de 99,95%, a precisão é igual a 94,31%, a taxa de recuperação ou detecção é de 95,62% e o F1 Score é de 94,1%. Além disso, também calculamos a Taxa de Falso Positivo ou Taxa de Falso Alarme, que é igual a 0,0005, como uma média do FPR de todas as classes e todos os splits.

Finalmente, calculamos as curvas ROC para cada classe que nosso modelo detecta, desde que as médias micro e macro dessas curvas. Isso pode ser mostrado na Figura 4 juntamente com a métrica de Área sob a Curva (AUC) para cada classe e a média para todas as classes. O valor AUC médio (macro) de todas as classes é igual a 0,99.

A comparação com a literatura relevante baseada no conjunto de dados CICIDS2017 não pode ser realizada diretamente, especialmente no caso de classificação multiclasse. Embora, na maioria dos casos, os detalhes da avaliação não sejam definidos explicitamente, uma comparação qualitativa pode ser realizada. Por exemplo, Vijayanand et al., [22], usou um classificador SVM resultando em uma precisão de classificação multiclasse igual a 99,85% e FPR igual a 0,0009, mas não é declarado se isso foi resultado de uma avaliação de validação cruzada ou se fosse uma divisão aleatória do conjunto de dados.
O mesmo ocorre em mais alguns casos, como em [23], onde Ferrag et. al. usando um classificador RNN chegou aos seguintes resultados: precisão de 99,81%, FPR de 0,009 e taxa de detecção de 94,09%. Da mesma forma, em [24] e [25] os autores usando floresta aleatória e LSTM, respectivamente, avaliaram seus modelos usando a métrica AUC relatando valores de 0,96 e 0,87 como uma média de todas as classes usadas. Em, [2], Zhang et al., utilizando os modelos CNN e LSTM relatam acurácia igual a 99,77%, precisão igual a 99,94%, recordação igual a 99,95% e F1 Score igual a 99,94%, classificando apenas 10 classes (eliminando a aqueles com menos instâncias no conjunto de dados) e sem relatar o uso de validação cruzada ou não.

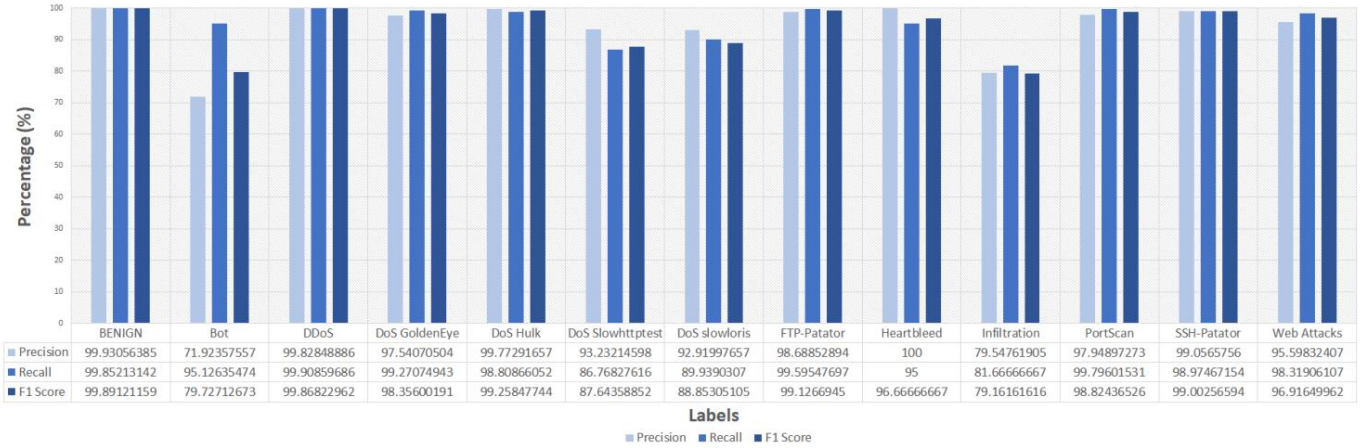
Mesmo que a comparação com trabalhos relacionados não seja um processo direto, o trabalho proposto funciona de forma eficiente mesmo com o modelo relativamente pequeno usado para a classificação.

VI. CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho implementamos um modelo de rede neural profunda que é capaz de detectar comportamento anormal ou malicioso (um potencial ataque cibernético) no tráfego de rede de uma empresa, e classificar o tipo de tráfego entre 13 casos diferentes.
Durante a análise de dados e o pré-processamento do conjunto de dados, conseguimos reduzir significativamente o número de recursos de entrada do modelo sem reduzir seu desempenho. Usando técnicas de over-sampling e down-smap, conseguimos detectar até mesmo classes minoritárias com poucos exemplos no conjunto de dados original com altos valores de revocação e precisão. O modelo proposto alcança resultados muito promissores em um problema de classificação multiclasse, sendo ao mesmo tempo um modelo muito simples e relativamente pequeno.

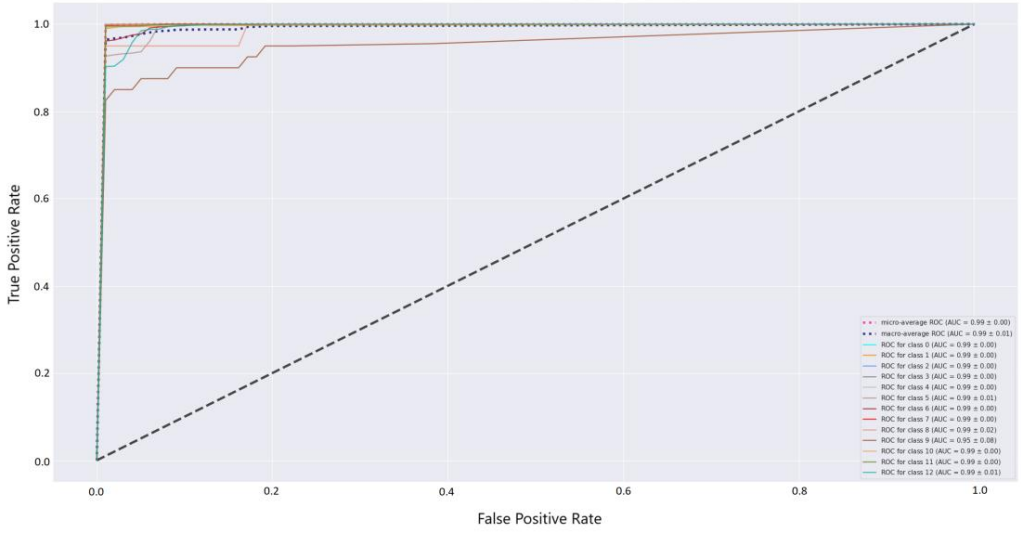
Machine Translated by Google

Fig. 3. Avaliação do modelo na validação cruzada de 10 vezes



	BENIGN	Bot	DDoS	DoS GoldenEye	DoS Hulk	DoS Slowhttptest	DoS slowloris	FTP-Patator	Heartbleed	Infiltration	PortScan	SSH-Patator	Web Attacks
Precision	99.93056385	71.92357557	99.82848886	97.54070504	99.77291657	93.23214598	92.91997657	98.68852894	100	79.54761905	97.94897273	99.0565756	95.59832407
Recall	99.85213142	95.12635474	99.90859686	99.27074943	98.80866052	86.76827616	89.9390307	99.59547697	95	81.66666667	99.79601531	98.97467154	98.31906107
F1 Score	99.89121159	79.72712673	99.86822962	98.35600191	99.25847744	87.64358852	88.85305105	99.1266945	96.66666667	79.16161616	98.82436526	99.00256594	96.91649962

Fig. 4. Curvas ROC na validação cruzada de 10 vezes



Como trabalho futuro planejamos fazer uma análise para reduzir ainda mais os recursos de entrada, testando várias técnicas como Análise de Componente Principal (PCA), Autoencoders, Análise de Componente Independente (ICA), etc. para não reduzir o desempenho do modelo. Mais um pensamento é melhorar e estender o conjunto de dados atual com ainda mais tipos de ataques baseados em rede e treinar novamente o modelo para poder detectá-los sem reduzir seu desempenho nas 13 classes originais. Por fim, planejamos tentar diferentes tipos de arquiteturas para abordar esse problema. Mais precisamente, consideramos a implementação de uma arquitetura RNN (LSTM, GRU, etc.) para o modelo, uma vez que o conjunto de dados contém dados sequenciais.

RECONHECIMENTO

Este trabalho recebeu financiamento da União Europeia Programa-Quadro Horizon 2020 para Pesquisa e Inovação, com o Título H2020-FORTIKA “Cyber-security Acceler ator para o ecossistema de TI de PMEs confiáveis” sob o contrato de concessão nº 740690.

REFERÊNCIAS

[1] N. Sultana, N. Chilamkurti, W. Peng e R. Alhadad, “Pesquisa sobre o sistema de detecção de intrusão de rede baseado em sdn usando abordagens de aprendizado de máquina,” Peer-to-Peer Networking and Applications, vol. 12, não. 2, pp. 493–501, março de 2019. [Online]. Disponível: https://doi.org/10.1007/s12083-017-0630-0

[2] Y. Zhang, X. Chen, L. Jin, X. Wang e D. Guo, “Detecção de intrusão de rede: com base em rede hierárquica profunda e dados de fluxo originais,” Acesso IEEE, vol. 7, pp. 37 004–37 016, 2019.

[3] Q. Zhou e D. Pezaros, "Avaliação de classificadores de aprendizado de máquina para detecção de intrusão de dia zero - uma análise do conjunto de dados CIC-AWS-2018", CoRR, vol. abs/1905.03685, 2019. [Online]. Disponível: http://arxiv.org/abs/1905.03685 [4] H.

Holm, “Detecção de intrusão baseada em assinatura para ataques de dia zero: (não) um capítulo fechado?” em 2014 47ª Conferência Internacional do Havaí sobre Ciências do Sistema, janeiro de 2014, pp. 4895–4904.

[5] V. Jyothsna, VV Rama Prasad e K. Munivara Prasad, “Uma revisão de sistemas de detecção de intrusão baseados em anomalias,” International Journal of Computer Applications, vol. 28, pp. 26–35, 08 2011.

Machine Translated by Google

[6] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou e C. Wang, “Aprendizado de máquina e métodos de aprendizado profundo para segurança cibernética ,” Acesso IEEE, vol. 6, pp. 35 365–35 381, 2018.

[7] J. Zhang, C. Chen, Y. Xiang, W. Zhou e Y. Xiang, "Classificação de tráfego da Internet agregando previsões de baías ingênuas correlacionadas", IEEE Transactions on Information Forensics and Security, vol. 8, não. 1, pp. 5–15, janeiro de 2013.

[8] Y. Zhang, X. Chen, L. Jin, X. Wang e D. Guo, “Detecção de intrusão de rede: com base em rede hierárquica profunda e dados de fluxo originais,” Acesso IEEE, vol. 7, pp. 37 004–37 016, 2019.

[9] J. Quittek, T. Zseby, B. Claise e S. Zander, "Requisitos para exportação de informações de fluxo IP (ipfix)", RFC, vol. 3917, pp. 1–33, 2004.

[10] C. Systems, “Cisco ios netflow”, 2012.

[11] R. Boutaba, M. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano e O. Caicedo Rendon, “Uma pesquisa abrangente sobre aprendizado de máquina para redes: evolução, aplicações e pesquisa oportunidades,” Journal of Internet Services and Applications, vol. 9, 05 2018.

[12] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras e B. Stiller, “Uma visão geral da detecção de intrusão baseada em fluxo IP,” IEEE Communications Surveys and Tutorials, vol. 12, pp. 343–356, 09 2010.

[13] R. Panigrahi e S. Borah, “Uma análise detalhada do conjunto de dados cicids2017 para projetar sistemas de detecção de intrusão”, vol. 7, pp. 479–482, 01 2018.

[14] I. Sharafaldin, A. Habibi Lashkari e A. Ghorbani, “Toward generator a new intrusion detection dataset and intrusion traffic caracterization,” 01 2018, pp. 108–116.

[15] C. Thomas, V. Sharma e N. Balakrishnan, "Utilidade do conjunto de dados darpa para avaliação do sistema de detecção de intrusão", 03 de 2008.

[16] J. Udhayan e H. Thiag, “Método de segregação estatística para minimizar as falsas detecções durante ataques ddos,” vol. 13 de 11 de 2011.

[17] A. Shiravi, H. Shiravi, M. Tavallaee e A. Ghorbani, “Para desenvolver uma abordagem sistemática para gerar conjuntos de dados de referência para detecção de intrusão,” Computers Security, vol. 31, pág. 357374, 05 2012.

[18] D. Protic, “Revisão dos conjuntos de dados kdd cup '99, nsl-kdd e kyoto 2006+,” Vojnotehnicki glasnik, vol. 66, pp. 580–596, 07 2018.

[19] A. Gharib, I. Sharafaldin, AH Lashkari e AA Ghorbani, "Uma estrutura de avaliação para conjunto de dados de detecção de intrusão", em 2016 Conferência Internacional sobre Ciência da Informação e Segurança (ICISS), dezembro de 2016, pp. 1–6 .

[20] O. Yavanoglu e M. Aydos, “Uma revisão sobre conjuntos de dados de segurança cibernética para algoritmos de aprendizado de máquina”, 12 2017.

[21] I. Ullah e QH Mahmoud, "Um modelo híbrido de dois níveis para detecção de atividade anômala em redes iot", em 2019 16ª Conferência Anual de Redes de Comunicação do Consumidor IEEE (CCNC), janeiro de 2019, pp. 1–6.

[22] V. Anand, D. Devaraj e B. Kannapiran, “Sistema de detecção de intrusão para rede mesh sem fio usando vários classificadores de máquinas de vetores de suporte com seleção de recursos baseada em algoritmo genético,” Computers Security, vol. 77, 04 de 2018.

[23] MA Ferrag e L. Maglaras, “Deepcoin: A novel deep learning and blockchain-based energy exchange framework for smart grids,” IEEE Transactions on Engineering Management, pp. 1–13, 2019.

[24] A. Binbusayyis e T. Vaiyapuri, “Identificando e comparando os principais recursos para detecção de intrusão cibernética: uma abordagem de conjunto”, IEEE Access, vol. 7, pp. 106 495–106 513, 2019.

[25] BJ Radford, BD Richardson e SE Davis, “Regras de agregação de sequência para detecção de anomalias no tráfego de rede de computadores,” CoRR, vol. abs/1805.03735, 2018. [Online]. Disponível: http://arxiv.org/abs/1805.03735 [26] A. Ahmim, L. Maglaras, MA Ferrag, M. Derdour e H. Janicke, “Um novo sistema hierárquico de detecção de intrusão baseado em árvore de decisão e regras-modelos baseados”, 12 2018.

[27] M. Idhammad, A. Karim e M. Belouch, “Sistema de detecção de intrusão distribuído para ambientes de nuvem com base em técnicas de mineração de dados”, vol. 127, 03 de 2018.

[28] A. Habibi Lashkari, G. Draper Gil, M. Mamun e A. Ghorbani, “Characterization of tor traffic using time based features,” 01 2017, pp.

[29] ——, “Caracterização de tráfego criptografado e vpn usando recursos relacionados ao tempo,” 02 2016.

[30] M. Kuhn e K. Johnson. (2013) Modelagem preditiva aplicada. Nova York, NY. [On-line]. Disponível: http://www.amazon.com/Applied Predictive-Modeling-Max-Kuhn/dp/1461468485/ [31] JC Kenna, “a contribuição de Sir Francis Galton para a antropologia,” O Jornal do Instituto Antropológico Real da Grã-Bretanha e Irlanda, vol. 94, nº. 2, pp. 80–93, 1964. [Online]. Disponível: http://www.jstor.org/stable/2844375

[32] F. Galton, “Regressão à mediocridade em estatura hereditária.” The Journal of the Anthropological Institute of Great Britain and Ireland, vol. 15, pp. 246–263, 1886. [Online]. Disponível: http://www.jstor.org/stable/2841583

[33] K. Pearson, “Nota sobre regressão e herança no caso de dois pais,” Proceedings of the Royal Society of London, vol. 58, pp. 240–242, 1895. [On-line]. Disponível: http://www.jstor.org/stable/115794 [34] I.-K. Yeo e RA Johnson, “Uma nova família de transformações de energia para melhorar a normalidade ou a simetria”, Biometrika, vol. 87, 12 2000.

[35] G. Box e D. Cox, “Uma análise das transformações (com discussão),” Jornal da Royal Statistical Society, Série B, vol. 26, pp. 211–252, 01 1964.

[36] G. Batista, R. Prati e M.-C. Monard, “Um estudo do comportamento de vários métodos para balancear dados de treinamento de aprendizado de máquina”, SIGKDD Explorations, vol. 6, pp. 20–29, 06 2004.

[37] G. Batista, A. Bazzan e M.-C. Monard, “Equilibrando dados de treinamento para anotação automatizada de palavras-chave: um estudo de caso.” 01 2003, pp. 10–18.

[38] N. Chawla, K. Bowyer, LO Hall e W. Philip Kegelmeyer, "Smote: técnica de sobreamostragem minoritária sintética", J. Artif. Intel. Res. (JAIR), vol. 16, pp. 321–357, 01 2002.

[39] DL Wilson, “Propriedades assintóticas das regras do vizinho mais próximo usando dados editados,” IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-2, não. 3, pp. 408–421, julho de 1972.

[40] YA LeCun, L. Bottou, GB Orr, e. G. Muller, Klaus-Robert”, GB Orr e K.-R. Muller, ¨ BackProp eficiente. Berlim, Heidelberg: SpringerBerlin Heidelberg, 2012, pp. 9–48. [On-line]. Disponível: https://doi.org/10.1007/978-3-642-35289-8 3 [41] X. Glorot e Y. Bengio, “Entendendo a dificuldade de treinar redes neurais feedforward profundas”, Journal of Machine Learning Research - Pista de Procedimentos, vol. 9, pp. 249–256, 01 2010.

[42] X. Glorot, A. Bordes e Y. Bengio, "Redes neurais retificadoras esparsas profundas", em Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson e M. Dudk, Eds., vol. 15. Fort Lauderdale, FL, EUA: PMLR, 11–13 de abril de 2011, pp. 315–323. [On-line]. Disponível: http://proceedings.mlr.press/v15/glorot11a.html [43] D. Kingma e J. Ba, “Adam: Um método para otimização estocástica,” Conferência Internacional sobre Representações de Aprendizagem, 12 2014.