UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de São José do Rio Preto

JOÃO OTAVIO GONÇALVES CALIS

MODELO DE PREDIÇÃO DE ATAQUES DE REDE UTILIZANDO NETFLOW

São José do Rio Preto
2021

JOÃO OTAVIO GONÇALVES CALIS

MODELO DE PREDIÇÃO DE ATAQUES DE REDE UTILIZANDO NETFLOW

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.
Financiadora: CAPES

Orientador:
Prof. Dr. Adriano Mauro Cansian

São José do Rio Preto
2021

C154m

Calis, João Otavio Gonçalves
Modelo de predição de ataques de rede utilizando Netflow / João Otavio Gonçalves Calis. -- São José do Rio Preto, 2021
70 f.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto
Orientador: Adriano Mauro Cansian

1. Ciência da computação. 2. Redes neurais (Computação). 3. Ciberterrorismo. 4. Análise de séries temporais. 5. Inteligência artificial. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

JOÃO OTAVIO GONÇALVES CALIS

MODELO DE PREDIÇÃO DE ATAQUES DE REDE UTILIZANDO NETFLOW

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.
Financiadora: CAPES

Comissão Examinadora:

Prof. Dr. Adriano Mauro Cansian
UNESP – Câmpus de São José do Rio Preto
Orientador

Prof. Dr. Geraldo Francisco Donegá Zafalon
UNESP – Câmpus de São José do Rio Preto

Prof. Dr. Siang Wun Song
USP – Universidade de São Paulo

São José do Rio Preto
09 de Setembro de 2021

Dedico esse trabalho a Deus, porque dele, por ele e para ele são todas as coisas. Dedico também a todos aqueles que me deram auxílio na confecção desse texto e também me apoiaram durante os anos de estudo.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus por ter me ajudado, guardado e sustentado até aqui.

Agradeço aos meus colegas do laboratório ACME!, Amanda Barbosa Sobrinho e Bruno Ferreira Leal, pela valiosa amizade ao longo de todos esses anos e ajuda fornecida durante todo o desenvolvimento desta pesquisa.

Agradeço ao meu orientador Adriano Mauro Cansian, pelo apoio, pelos ensinamentos fornecidos e pela contribuição para minha formação acadêmica e pessoal.

Agradeço aos atuais membros do laboratório ACME!, pelo apoio ao desenvolvimento deste projeto e aos membros antigos pelo conhecimento transferido até aqui.

Agradeço, também, à Universidade Estadual Paulista “Júlio de Mesquita Filho”, pelo curso de pós-graduação ali ministrado por professores de excelência.

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

RESUMO

A Internet proporcionou inúmeros avanços para a sociedade desde de sua criação. Os computadores têm se mostrado úteis para todas as áreas do conhecimento. Esses dispositivos cada vez mais heterogêneos estão interligados à Internet de modo a prover diversas facilidades a indivíduos e organizações. Como em todos setores da sociedade, pessoas mal-intencionadas prejudicam o bom funcionamento da rede. Por esse motivo, faz-se necessária a aplicação de abordagens de defesa contra ataques cibernéticos. Por longo tempo tem-se elaborado técnicas de proteção contra ataques reativas, que são baseadas em sua detecção em tempo real. Embora tais técnicas tenham sua importância no cenário de defesa atual, a abordagem proativa de prevenção de ataques tem ganhado força recentemente. O potencial dos modelos preditivos aplicados em situações de cibersegurança ainda continua sendo explorado pelos pesquisadores da área de cibersegurança, sendo o uso das redes neurais a abordagem que se destaca entre as possibilidades. Dessa forma, o presente trabalho propõe um modelo expansível de previsão de ataques a redes de computadores baseado em redes neurais artificiais capaz de prever o volume de novos ataques de uma determinada categoria bem como realizar a classificação de ataques em uma rede com base no IP do atacante e da vítima.

Palavras-Chave: Segurança de Redes; Previsão de ataques; Aprendizado de máquina; Redes Neurais Artificiais; Séries Temporais.

ABSTRACT

The Internet has provided countless advances to society since its creation. Computers have proven to be useful for all areas of knowledge. These increasingly heterogeneous devices are interconnected to the Internet in order to provide different facilities to individuals and organizations. As in all sectors of society, malicious people hinder the proper functioning of the network. For this reason, it is necessary to apply defense approaches against cyber attacks. For a long time, techniques to protect against reactive attacks, which are based on real-time detection, have been developed. While such techniques are important in today's defense landscape, the proactive approach to attack prevention has gained traction recently. The potential of predictive models applied in cybersecurity situations is still being explored by cybersecurity researchers, with neural networks being the approach that stands out among the possibilities. Thus, this work proposes an expandable model for predicting attacks on computer networks based on artificial neural networks capable of predicting the volume of new attacks in a given category as well as classifying attacks in a network based on the IP of the attacker and victim.

Keywords: Network Security; Attack Prediction; Machine Learning; Artificial Neural Networks; Time Series.

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Representação gráfica da estrutura de um neurônio artificial 28
Figura 2.2 – Representação de uma rede neural com duas camadas ocultas 30
Figura 4.1 – Arquitetura do modelo de CSA 47
Figura 4.2 – Exemplo de visualização de dados do *Kibana* 48
Figura 4.3 – Volume de ataque de força bruta num período de 90 dias 53
Figura 5.1 – Recorte da série temporal gerada a partir dos dados de varredura 60
Figura 5.2 – Valor predito e valor esperado do conjunto de varredura..... 61
Figura 5.3 – Recorte de resultado da previsão para os ataques de força bruta 62

LISTA DE TABELAS

Tabela 3.1 – Resumo comparativo dos trabalhos relacionados 44
Tabela 5.1 – Resultados do treinamento das redes neurais consideradas 57
Tabela 5.2 – Resultados dos testes das redes neurais consideradas..... 58
Tabela 5.3 – Resultados do modelo de predição de volume de ataque..... 59

LISTA DE ALGORITMOS

Algoritmo 4.1 – Preparação dos dados do índice exportado 49

Algoritmo 4.2 – Preparação dos dados para a classificação 51

Algoritmo 4.3 – Treinamento das redes neurais do classificador 52

Algoritmo 4.4 – Preparação do conjunto de treinamento da rede recorrente 54

LISTA DE ABREVIATURAS E SIGLAS

API: *Application Programming Interface*

APT: *Advanced Persistent Threat*

CSA: *Cyber Situational Awareness*

CNN: *Convolutional Neural Network*

DDoS: *Distributed Denial of Service*

DNS*: Domain Name System*

ELK: *Elasticsearch, Logstash and Kibana*

GBDT: *Gradient Boosted Decision Trees*

IDS: *Intrusion Detection System*

IP: *Internet Protocol*

LSTM: *Long Short-Term Memory*

NN: *Neural Network*

RAM: *Random Access Memory*

RNN: *Recurrent Neural Network*

SSH: *Secure Shell*

SVM: *Support Vector Machine*

TCP: *Transmission Control Protocol*

ToS: *Type of Service*

UDP: *User Datagram Protocol*

SUMÁRIO

CAPÍTULO 1 - Introdução 13
1.1 Introdução e Motivação 13
1.2 Justificativa 15
1.3 Objetivo 16
1.4 Organização do trabalho 16
CAPÍTULO 2 - Fundamentação Teórica 17
2.1 Considerações iniciais 17
2.2 Fluxo de Rede de Computadores 17
2.2.1 *Netflow* versão 9 18
2.3 Ataques às redes de computadores 19
2.3.1 Ataques de varredura (*Scanning*) 20
2.3.2 Ataques de força bruta 20
2.4 Comunicações maliciosas 21
2.5 Elastic Stack 21
2.5.1 Elasticsearch 22
2.5.2 *Kibana* 22
2.5.3 *Beats* 22
2.6 Previsão de ataques 23
2.6.1 Métodos baseados em modelos discretos 23
2.6.2 Métodos baseados em modelos contínuos 24
2.6.3 Métodos baseados em aprendizado de máquina e mineração de dados 25
2.6.4 Métodos baseados em outras abordagens 26
2.7 Consciência Situacional em Cibersegurança 27
2.8 Redes neurais artificiais 28
2.8.1 Perceptron multicamadas 30

2.8.2 Redes Convolucionais 31
2.8.3 Redes Recorrentes 32
2.9 Aprendizado profundo 33
2.10 Métricas de desempenho 34
CAPÍTULO 3 - Trabalhos Relacionados 37
3.1 Considerações iniciais 37
3.2 Principais trabalhos relacionados 37
3.3 Trabalhos relacionados secundários 39
3.4 Conclusões sobre os trabalhos relacionados 43
CAPÍTULO 4 - Metodologia 46
4.1 Considerações iniciais 46
4.2 Visão geral do sistema de CSA 46
4.3 Visão geral do modelo proposto 49
4.4 Preparação dos dados 49
4.5 Rede de classificação de ataques 50
4.6 Rede de previsão de volume de novos ataques 52
4.7 Considerações finais 55
CAPÍTULO 5 - Resultados 56
5.1 Considerações iniciais 56
5.2 Classificador de ataques 56
5.3 Previsão de volume de ataque 59
5.4 Considerações sobre os resultados obtidos 62
CAPÍTULO 6 - Conclusões 64
6.1 Conclusões 64
6.2 Dificuldades encontradas 65
6.3 Trabalhos futuros 65
REFERÊNCIAS 67

CAPÍTULO 1 - Introdução

1.1 Introdução e Motivação

Embora a ciência da computação seja relativamente nova, os expressivos avanços tecnológicos produzidos por ela têm revolucionado a forma como vivemos hoje. A percepção do potencial promissor da tecnologia da informação tornou-a o foco de grandes organizações. Elas perceberam que os dados, hoje, são mais valiosos que o ouro. Esse fato permitiu que os computadores evoluíssem de máquinas severamente limitadas para dispositivos cada vez mais eficientes e capazes de se comunicar. Essa comunicação é feita predominantemente por meio da *Internet*, que trouxe uma nova fase para a revolução da era da informação [1].

A *Internet* é uma das maiores obras de engenharia produzidas pela humanidade. Bilhões de pessoas fazem uso direto e indireto deste sistema diariamente, através de diversos tipos de dispositivos. Um sistema de tamanha complexidade, composto por diversos subsistemas e mantido por diversas organizações com interesses não necessariamente alinhados, requer princípios bem definidos para garantir seu correto funcionamento [2].

Existem indivíduos mal-intencionados por toda parte, e o meio digital não é uma exceção. Embora a *Internet* possua uma ampla gama de protocolos e diretrizes que visam nortear seu bom funcionamento, situações em que seus usuários causam danos à rede são comuns. Esses danos podem ser originados por imprudência ou falta de

conhecimento, isto é, pela má operação de algum elemento da rede. No entanto, existem situações em que o dano é proposital [3].

Tanto *hardware* quanto *software* possuem pontos fracos que podem ser explorados e atacados por pessoas que tenham conhecimento da existência dos mesmos. No contexto da *Internet*, a preocupação está em proteger negócios, instituições em geral e dados pessoais inseridos na rede de possíveis atacantes [3].

Os avanços tecnológicos permitem que os ataques aos sistemas computacionais sejam cada vez mais diversificados, sofisticados e complexos [3]. Desta tendência decorre a necessidade de procedimentos de defesa que acompanhem a evolução dos ataques, visando manter os níveis de segurança aceitáveis.

Esses mecanismos procuram deter ou ao menos mitigar um ataque. No contexto de redes de computadores, os mecanismos mais comuns são as técnicas de criptografia, configuração de filtros nos roteadores, filtragem de pacotes, ferramentas de detecção de intrusão[1] e o uso de *firewalls* [4].

Estas abordagens são direcionadas a detecção de ataques e, embora tenham seu valor por de fato serem efetivas para alguns casos, sua natureza reativa acarreta na dependência da correspondência com padrões específicos e anomalias já observados na rede para que uma contramedida seja tomada [5].

Em contrapartida, existem abordagens de natureza proativa que permitem inferir preventivamente atividades maliciosas. Essas abordagens são baseadas em previsão, predição[2] ou projeção de ataques, permitindo que ações possam ser tomadas antes que um possível ataque seja bem-sucedido [5].

Embora seja um campo menos explorado em comparação com a detecção, a previsão de ataques está ganhando atenção pelo seu potencial proativo. Pesquisas na área têm aplicado métodos de aprendizado de máquina para produzirem sistemas de previsão de ataques em tempo real, contribuindo com o avanço em segurança cibernética [5], [7].

O maior desafio nas abordagens baseadas em previsão é a elaboração de um modelo que seja dinâmico e capaz de apresentar projeções confiáveis com a

[1] Ou Sistema de Detecção de Intrusão (Intrusion Detection System – IDS).

[2] Embora sejam termos tratados praticamente como sinônimos e não haja consenso absoluto sobre a diferença entre eles, a predição se refere a dizer algo com antecedência, que ocorrerá no futuro. A previsão é um tipo especial de predição no qual utiliza-se dados de acontecimentos passados para prever acontecimentos futuros [6].

antecedência necessária para que os responsáveis pela segurança da rede possam planejar estratégias de defesa antes que a atividade maliciosa se inicie. O dinamismo é necessário para que o modelo não seja capaz de prever apenas um conjunto limitado de ataques pré-definidos por especialistas, o que o torna inútil conforme as ameaças evoluam ao longo do tempo. Esses dois aspectos devem ser considerados e incorporados no projeto do modelo de previsão. Caso contrário, os esforços de previsão nada valerão [5].

Diante do exposto, é perceptível a necessidade do desenvolvimento de técnicas eficientes de previsão de ataques como uma nova abordagem para a defesa cibernética. Essas técnicas serão úteis para que os sistemas sejam mantidos seguros e protegidos contra as constantes variações e complexidade crescente dos ataques modernos, contribuindo para a manutenção da integridade dos mesmos.

1.2 Justificativa

Os sistemas de detecção de intrusão foram por muitos anos a abordagem predominante na defesa contra ataques às redes de computadores. No entanto, a desvantagem dessa abordagem é sua característica reativa, isto é, uma contramedida só pode ser tomada no momento em que a ameaça já começou. Os métodos de previsão de ataques possuem a vantagem de serem proativos, isto é, permitem aos administradores de rede tomaram as devidas medidas antes que os ataques iniciem. Segundo [5], existem vários métodos para prever ataques em cibersegurança, sendo agrupados em modelos discretos, modelos contínuos, modelos baseados em aprendizado de máquina e mineração de dados e modelos que utilizam outros métodos não classificáveis nos grupos anteriores. As abordagens baseadas em aprendizado de máquina, mais especificamente em redes neurais artificiais, apresentam atualmente os melhores resultados, sendo entre os grupos o mais indicado.

Os trabalhos propostos atualmente possuem limitações ou questões importantes que não são consideradas em sua formulação. Além disso, eles apresentam, na maioria das vezes, a possibilidade de predição em um contexto ou granularidade específicos, o que limita sua contribuição efetiva. Assim, a proposta desenvolvida neste trabalho possibilita realizar a previsão de ataques de rede em dois contextos diferentes.

1.3 Objetivo

O objetivo deste trabalho é o desenvolvimento de um modelo de predição de ataques capaz de atuar na classificação e previsão do volume de ataques em uma rede de grande porte utilizando apenas fluxos de rede. O modelo deve ser útil para funcionar como o componente de previsão de um sistema de CSA também baseado em fluxos para uma rede de grande porte. Este objetivo principal será alcançado por meio de três objetivos secundários:

- Coletar, armazenar e classificar eventos de rede maliciosos indexados a partir dos dados de fluxo do Sistema Autônomo da UNESP visando utilizá-los como informação de entrada para o modelo;
- Limpar, organizar, preparar e rotular os dados a fim de gerar os conjuntos de entrada para as redes neurais que compõem o modelo;
- Escolher a melhor arquitetura de rede neural para o submodelo responsável pela classificação de ataques;
- Analisar a melhor forma de realizar a previsão de volumes de ataques para os tipos de ataques considerados.

1.4 Organização do trabalho

O restante deste trabalho está dividido em seis capítulos, incluindo o atual. No Capítulo 2 é apresentada a fundamentação teórica do trabalho, necessária para que o leitor compreenda os conceitos e tecnologias utilizadas. No capítulo 3, tem-se a descrição dos principais trabalhos relacionados e do estado da arte. No Capítulo 4 é descrita a metodologia aplicada para o desenvolvimento do modelo proposto. No Capítulo 5 descreve-se os resultados obtidos para o modelo. Por fim, no Capítulo 6 são apresentas as conclusões, dificuldades enfrentadas e possíveis trabalhos futuros com relação a esse projeto.

CAPÍTULO 2 - Fundamentação Teórica

2.1 Considerações iniciais

Este capítulo tem como objetivo apresentar os conceitos fundamentais para a compreensão do presente trabalho. Na seção 2.2, são abordados conceitos básicos de fluxo de redes computadores; na seção 2.3, é apresentada uma descrição dos principais tipos de ataques a redes de computadores considerados nesse trabalho; na seção 2.4, tem-se a descrição de comunicações maliciosas; na seção 2.5, é apresentada a pilha Elastic, responsável pela manipulação dos fluxos de rede *Netflow*; na seção 2.6, são discutidas as quatro categorias de abordagens de previsão de ataques; na seção 2.7 tem-se um breve panorama do conceito de consciência situacional em cibersegurança; na seção 2.8 são expostos alguns conceitos pertinentes às principais arquiteturas de redes neurais artificiais; na seção 2.9, são feitas algumas considerações relacionadas ao aprendizado profundo e, por fim, na seção 2.10, são dadas breves definições das métricas de desempenho consideradas neste trabalho.

2.2 Fluxo de Rede de Computadores

O monitoramento do fluxo de rede facilita sistemas de detecção e prevenção de intrusões em larga escala, análise de dados, planejamento de capacidade, retenção de dados e outras operações importantes para o gerenciamento de rede [8], [9]. Existem diversas aplicações relevantes utilizando fluxos de redes [10]. Alguns exemplos são a

análise por meio de técnicas probabilísticas [11], identificação de fluxo em tempo real [12], detecção de anomalias utilizando redes neurais [13] e a coleta e análise de dados de segurança [14].

Várias definições de fluxo de rede podem ser encontradas na literatura. Os autores em [9] propuseram uma definição de fluxo de rede como sendo uma sequência de pacotes que passam por um ponto de observação na rede durante um determinado intervalo de tempo. Todos os pacotes que pertencem a um fluxo específico têm um conjunto de propriedades comuns derivadas dos dados contidos no pacote, pacotes anteriores do mesmo fluxo e do tratamento de pacotes no ponto de observação.

As propriedades obtidas a partir de um fluxo podem variar. No entanto, existem sete campos considerados como campos chave para um fluxo, sendo eles: endereço IP de origem e de destino, número da porta de origem e de destino, tipo de protocolo da camada de transporte, tipo de serviço (ToS) e interface lógica de entrada. O número de *bytes*, as *flags* TCP, número de pacotes do fluxo e o carimbo de tempo de início do fluxo são outros campos comumente considerados [8], [9], [10].

O processo de criação de um fluxo compreende inicialmente a captura e o processamento dos pacotes. A captura dos pacotes é geralmente realizada por um equipamento da camada de rede habilitado para esta tarefa. O objetivo desse processamento é extrair os valores das propriedades escolhidas de pacotes individuais e as informações de tratamento de pacotes correspondentes. De posse dos dados de fluxo, é possível realizar análise e classificação do tráfego, definição de assinaturas de ataques, além de previsão de eventos maliciosos [9], [10].

2.2.1 *Netflow* versão 9

Netflow é o nome dado pela Cisco, empresa que o criou, à tecnologia presente em alguns roteadores que os habilita a capturar dados de pacotes de entrada e saída. Os fluxos capturados ainda ativos são armazenados em cache. Ao término do fluxo, este é exportado para um dispositivo de coleta, comumente conhecido como coletor. Caso seja necessário o *Netflow* completo, deve-se ativá-lo explicitamente [8].

Com o tempo, os dados de fluxo podem custar grandes quantidades de armazenamento, especialmente para roteadores que gerenciam grandes redes. Para

minimizar este problema, os roteadores exportam os fluxos por amostragem; esta é a configuração padrão. Se a exportação completa for necessária, também deve-se ativá-la explicitamente [8].

Os dados são exportados pelo *Netflow* em datagramas UDP em um dos seguintes formatos: versão 1, versão 5, versão 7, versão 8 ou versão 9. A versão 9 é a versão mais atual e é também utilizada neste trabalho por ser mais flexível e extensível que as anteriores [8].

O conjunto de estatísticas capturado pelo *Netflow* pode ser usado para uma ampla variedade de propósitos. Entre eles, destaca-se a utilização da tecnologia para análise de segurança, o que permite identificar e classificar diferentes tipos de ataques de negação de serviço, vírus[3] e *worms*[4] em tempo real [8].

2.3 Ataques às redes de computadores

Segundo [15], no contexto de cibersegurança, ataques são técnicas que os atacantes usam para explorar as vulnerabilidades em aplicativos. O objetivo deste tipo de indivíduo ou organização é violar o sistema de outro indivíduo ou organização [16]. Por consequência, os ataques cibernéticos viabilizam crimes cibernéticos, como roubo de informações, fraudes e esquemas de *ransomware*[5] [17].

Existem diversos tipos de ciberataques. Os mais comuns são: ataques por *malware*[6], *phising*[7], força bruta, ataque de negação de serviço, *botnets*[8] e ataques de

[3] Vírus é um programa ou parte de um programa de computador, normalmente malicioso, que se propaga inserindo cópias de si mesmo e se tornando parte de outros programas e arquivos. (Disponível em cartilha.cert.br/malware)
[4] *Worm* é um programa capaz de se propagar automaticamente pelas redes, enviando cópias de si mesmo de computador para computador. (Disponível em cartilha.cert.br/malware)
[5] *Ransomware* é um tipo de código malicioso que torna inacessíveis os dados armazenados em um equipamento, geralmente usando criptografia, e que exige pagamento de resgate (*ransom*) para restabelecer o acesso ao usuário. (Disponível em cartilha.cert.br/ransomware)
[6] Códigos maliciosos (*malware*) são programas especificamente desenvolvidos para executar ações danosas e atividades maliciosas em um computador. (Disponível em cartilha.cert.br/malware)
[7] *Phishing*, *phishing-scam* ou *phishing/scam*, é o tipo de fraude por meio da qual um golpista tenta obter dados pessoais e financeiros de um usuário, pela utilização combinada de meios técnicos e engenharia social.
[8] *Botnet* é uma rede formada por centenas ou milhares de computadores zumbis e que permite potencializar as ações danosas executadas pelos *bots*.

varredura [16], [17], [18], [19]. Os ataques considerados no escopo deste trabalho são descritos a seguir.

2.3.1 Ataques de varredura (*Scanning*)

Ataques de varredura costumam ser o primeiro passo no ciclo de vida de um ataque cibernético. Em um cenário típico, o atacante executa uma atividade de verificação para procurar vulnerabilidades antes de iniciar um ataque à vítima vulnerável [19].

O objetivo deste ataque é levantar o máximo de informações possível sobre a rede ou hospedeiro para determinar possíveis tentativas de ataques futuros. Informações de interesse costumam ser: portas abertas, endereços IP e versões de *software* sendo executadas em um hospedeiro.

2.3.2 Ataques de força bruta

Um ataque de força bruta tem por objetivo obter acesso a uma máquina remota executando tentativas de autenticação. Esses ataques são realizados por meio da verificação sistemática de todas as senhas possíveis até que a correta seja encontrada. As senhas escolhidas por humanos são inerentemente fracas. Os usuários tendem a selecionar senhas simples porque são mais fáceis de lembrar. Às vezes, eles não alteram a senha padrão da máquina ou simplesmente usam o nome de usuário como senha [20], [21].

Uma máquina comprometida por um ataque de força bruta pode causar sérios danos ao se juntar a *botnets*, distribuir informações confidenciais, participar de ataques distribuídos entre outros. Os ataques de força bruta ainda são uma ameaça preocupante para as redes de computadores [20].

A pesquisa sobre a detecção de ataques de força bruta geralmente se concentra na detecção no nível do hospedeiro, em que os *logs* de acesso são inspecionados. Caso o número de tentativas de *login* com falha em um horário específico exceda um número limite predefinido, um alerta será acionado [21].

2.4 Comunicações maliciosas

Além de ataques bem definidos, existe troca de informações entre dispositivos que não necessariamente configuram um ataque, ao menos um que seja conhecido, mas que levantam algum tipo de suspeita do ponto de vista analítico. Em geral, essas comunicações maliciosas ocorrem com dispositivos que possuem seus endereços IP catalogados em uma lista de bloqueio [22].

A adição a essas listas ocorre quando uma instituição especializada considera que o IP em questão está envolvido em atividades de mineração de criptomoedas, comando e controle, disseminação de *malwares* e demais atividades similares que o façam ser considerado comprometido [23]. Neste trabalho são consideradas as listas de IPs comprometidos e IPs de mineradores.

2.5 Elastic Stack

Também conhecida como ELK Stack, é uma pilha de aplicações projetada para lidar com grande volume de dados. A sigla ELK é o acrônimo para três projetos *open source*: *Elasticsearch*, *Logstash* e *Kibana*. O *Elasticsearch* é um mecanismo de busca e análise. O *Logstash* é um *pipeline* de processamento de dados do lado do servidor que faz a ingestão de dados a partir de inúmeras fontes simultaneamente, transforma-os e envia-os para o *Elasticsearch*. O *Kibana* permite que os usuários visualizem dados por meio de diagramas e gráficos no *Elasticsearch* [24].

Atualmente, a pilha ELK conta com um agente de dados leve, o *Beats*. Por esse motivo, a pilha for rebatizada para Elastic Stack, levando apenas o nome da aplicação central.

2.5.1 Elasticsearch

O *Elasticsearch* é um mecanismo de busca e análise *RESTful*[9] distribuída capaz de atender a um número crescente de casos de uso. Sendo o núcleo do *Elastic Stack*, ele armazena dados de maneira centralizada para que se possa descobrir informações relevantes sobre eles por meio da realização e combinação de muitos tipos de buscas. Outra vantagem da utilização das agregações do *Elasticsearch* é que elas permitem considerar um panorama geral para analisar tendências e padrões nos dados [25].

2.5.2 *Kibana*

O *Kibana* é a aplicação que permite visualizar os dados armazenados no *Elasticsearch*. A ferramenta permite selecionar a maneira como os dados serão visualmente apresentados. É possível visualizá-los por meio de formas bem conhecidas, como: histogramas, gráficos de linha, setores, dispersão, entre outros. Também é possível definir visualizações personalizadas. Por fim, as visualizações do *Kibana* são interativas, de modo a fornecer os meios para o usuário obter altos níveis de percepção sobre os dados sob análise [26].

2.5.3 *Beats*

Aplicações denominadas *Beats* são as aplicações leves da pilha Elastic instaladas em hospedeiros dos quais se deseja enviar dados operacionais ao *Elasticsearch*. Cada tipo de dado possui um *Beat* específico, sendo eles [27]:

- *Auditbeat*: para dados de auditoria;
- *Filebeat*: para arquivos de *log*;
- *Functionbeat*: dados de nuvem;
- *Heartbeat*: informações de disponibilidade;
- *Journalbeat*: journals do Systemd do Linux;

[9] Mecanismo ou API útil no contexto de serviços *web* responsável por definir certas restrições arquiteturais a componentes de software.

- *Metricbeat*: métricas;
- *Packetbeat*: tráfego de rede;
- *Winlogbeat*: eventos de log do Windows.

2.6 Previsão de ataques

Segundo o levantamento dos autores em [5], as técnicas de previsão de ataques presentes na literatura podem ser agrupadas em quatro categorias: modelos discretos, modelos contínuos, métodos baseados em aprendizado de máquina e mineração de dados, e outras abordagens não classificáveis nas categorias anteriores.

As próximas subseções 2.6.1 a 2.6.4 são baseadas no levantamento feito em [5] e apresentam breves descrições de cada uma dessas categorias.

2.6.1 Métodos baseados em modelos discretos

Dentro desta categoria se enquadram os métodos baseados em grafos de ataque, redes bayesianas, modelos de Markov e teoria dos jogos.

Um grafo de ataque é uma representação gráfica de um cenário de ataque, e é considerado um método popular de representação formal de ataques. Os primeiros métodos de previsão de ataque foram baseados em gráficos de ataque. Os grafos de ataque também servem de base para outras abordagens de verificação de modelo, por exemplo, métodos que utilizam redes bayesianas, modelos de Markov e teoria dos jogos.

As redes bayesianas constituem outra abordagem para a previsão de ataques. Elas estão intimamente relacionadas às abordagens de verificação de modelo com base em grafos de ataque, uma vez que uma rede bayesiana é tipicamente construída a partir de um grafo de ataque. A principal característica das redes bayesianas são as variáveis e probabilidades condicionais que são refletidas no modelo. Em alguns casos, restrições adicionais são definidas nas redes bayesianas.

Os modelos de Markov são frequentemente representados também como um grafo, o que tornam os métodos baseados neles semelhantes aos métodos baseados em

grafos de ataque e redes bayesianas. Ao contrário das abordagens descritas anteriormente, os modelos de Markov operam bem na presença de estados e transições não observáveis, o que remove a dependência dos métodos de detecção de intrusões e previsão de ataques por informações completas. Isso permite uma detecção bem-sucedida de intrusões e previsão de ataques, mesmo que algumas etapas de ataque não tenham sido detectadas ou não possam ser completamente deduzidas.

As abordagens baseadas na teoria dos jogos para predição de ataques são semelhantes às abordagens gráficas discutidas anteriormente. O jogo é usado como um modelo de interação entre um atacante e um defensor. Ao contrário das abordagens gráficas apresentadas, os métodos teóricos do jogo visam encontrar a melhor estratégia para os jogadores, em vez da progressão de ataque mais frequente observada nos dados históricos. Assim, as abordagens de teoria de jogos parecem promissoras, especialmente para a previsão da atividade de atacantes avançados.

2.6.2 Métodos baseados em modelos contínuos

Dentro desta categoria se enquadram os métodos baseados em séries temporais e modelos cinza.

Séries temporais representam uma ferramenta muito interessante para análise preditiva, sendo utilizadas em vários campos, incluindo segurança cibernética. São também frequentemente empregadas na detecção de anomalias.

Uma série temporal representa padrões de tráfego de rede comuns. Posteriormente, os desvios que não coincidem com os valores esperados de tráfego de rede em um determinado momento são proclamados como uma anomalia. Embora a terminologia e os métodos de detecção de anomalias sejam semelhantes à previsão de ataque, os dois casos de uso são substancialmente diferentes.

Os modelos cinza são normalmente usados para prever situações de segurança cibernética e definem outro exemplo de metodologias que empregam um modelo matemático contínuo. Na terminologia da teoria de cinza, uma situação sem informação é definida como preta e uma situação com informações completas como branca. Como as duas opções são idealizadas, os problemas do mundo real estão em algum lugar no meio, em uma situação definida como cinza.

2.6.3 Métodos baseados em aprendizado de máquina e mineração de dados

Métodos baseados em aprendizado de máquina e mineração de dados formam uma nova categoria devido a sua eficiência de classificação. O aprendizado de máquina está intimamente ligado à mineração de dados, pois visa superar uma grande desvantagem dos modelos de previsão de ataques baseados em modelo, que é a dependência de modelos fornecidos por um especialista em segurança.

Existem etapas comuns na preparação de um modelo de aprendizado de máquina. A princípio, são consideradas duas fases, treinamento e teste. Durante a fase de treinamento, amostras apropriadas do conjunto de dados de aprendizagem são aprendidas. É importante ressaltar que essas amostras devem ser o mais diversificadas possível para que o modelo possa alcançar uma boa capacidade de generalização. Na fase de teste, dados diferentes daqueles usados na fase de treinamento são processados pelo modelo e o método de aprendizado de máquina produz resultados, como continuações previstas de sequências de ataque.

No processo de criação do modelo, entretanto, há também uma fase de validação entre o treinamento e os testes. Nesta fase, outro conjunto de dados, ou uma partição diferente do mesmo, é usado para avaliar se o modelo atingiu resultados satisfatórios na fase de treinamento ou, no caso da comparação entre vários modelos, qual deles deve ser usado para teste.

Outro aspecto importante do aprendizado de máquina é a supervisão. Um modelo pode ser treinado autonomamente, isto é, ele ajusta suas configurações internas com base nos dados de entrada sem a intervenção direta de um especialista. Os modelos que se enquadram nesta definição são chamados não supervisionados.

Outra possibilidade é entregar os dados de entrada do modelo total ou parcialmente rotulados por um especialista humano, e o algoritmo irá se basear nos rótulos para ajustar seus mecanismos de classificação a fim de melhorar a taxa de acerto. Modelos em que a aprendizagem depende total ou parcialmente de dados rotulados são chamados de supervisionados ou semi-supervisionados.

Os dados de entrada para o modelo também devem ser devidamente analisados e tratados. Nem todos os dados disponíveis no conjunto serão necessariamente relevantes para previsão do modelo. Dessa forma, o problema de identificação das classes e atributos de classe nos dados, ou seja, de determinar as entradas adequadas

dos métodos de aprendizado de máquina, é conhecido como extração de características.

Há várias abordagens e métodos de aprendizado de máquina que podem ser usados para prever eventos futuros, como ataques cibernéticos. As redes neurais artificiais são o método de aprendizado de máquina mais utilizado atualmente. Embora fossem proeminentes desde o início dos estudos sobre aprendizado de máquina, devido às limitações de *hardware* da época, elas foram substituídas por SVMs (*Support Vector Machines*), que ofereciam menor complexidade computacional e menor tempo de aprendizado. No entanto, com os avanços tecnológicos, especialmente em termos de processamento e memória RAM, as redes neurais ganharam destaque novamente.

A mineração de dados está intimamente relacionada ao aprendizado de máquina. Sua introdução no domínio da segurança cibernética criou um avanço para as previsões de ataques. Em contrapartida, ela desempenha apenas uma função de suporte na previsão de ataques. Quer seja usada em conjunto com técnicas de aprendizado de máquina ou com os demais modelos, a técnica não deve influenciar diretamente o modelo.

2.6.4 Métodos baseados em outras abordagens

Existem outras abordagens encontradas na literatura que não podem ser facilmente classificadas devido a sua natureza ou alta especificidade. Dentro desta categoria, enquadram-se os métodos baseados em similaridades, previsão de volume DDoS[10], computação evolutiva e fontes de dados não convencionais.

As abordagens baseadas em similaridade abordam principalmente o problema do reconhecimento da intenção do invasor, calculando uma métrica de similaridade com um ataque observado anteriormente. Esta abordagem pode ser aplicada em conjunto com alertas de segurança, rede de *links* semânticos, *logs* de *honeypots*[11] e redes definidas por *software*.

[10] Um ataque de rede realizado de maneira distribuída com o objetivo de negar algum serviço inundando-o com um volume enorme de requisições de modo que o servidor fique inoperante.
[11] Um *honeypot* é um recurso computacional de segurança dedicado a ser sondado, atacado ou comprometido. (Disponível em cert.br/docs/whitepapers/honeypots-honeynets)

As previsões de ataques DDoS concentram-se principalmente na identificação da fase inicial de um ataque, na qual o volume de tráfego de rede falso aumenta, e na previsão do volume do ataque. O volume de um ataque DDoS é a característica mais importante desses ataques. As métricas para o volume DDoS são a taxa de pacotes ou *bytes* por segundo e o número estimado de máquinas comprometidas envolvidas no ataque. O conhecimento prévio do volume do ataque nos informa se o sistema de destino ou a rede podem suportar o ataque ou se há capacidade suficiente de defesa, por exemplo, em centros de limpeza.

Algoritmos evolutivos são uma abordagem relativamente nova na previsão de ataques. Trabalhos neste segmento apontam melhorias na avaliação da situação de segurança da rede. Os conceitos de computação evolutiva podem ser empregados em conjunto com modelos de regras de crenças, que são construídos com base em conhecimentos especializados e em dados históricos. Os estudos também apontam que esta abordagem pode ser uma boa alternativa em relação aos modelos cinza.

Uma nova tendência nas previsões de segurança cibernética está usando fontes de dados não convencionais. Exemplos de dados não convencionais para previsão de ataques são os *logs* DNS e os sentimentos extraídos das postagens no *Twitter*.

2.7 Consciência Situacional em Cibersegurança

O termo consciência situacional em cibersegurança ou, do inglês, *Cyber Situational Awareness* (CSA) se refere à necessidade de empresas, governos e outras organizações manterem uma visão holística de seus sistemas de informação e controle em relação a ameaças cibernéticas [28].

A consciência situacional é um fenômeno plurifacetado, bem estudado e de nível estratégico superior que pode ser considerado de várias perspectivas diferentes. Do ponto de vista técnico, a consciência situacional se resume à compilação, processamento e fusão de dados. Considerando a perspectiva cognitiva, a consciência situacional diz respeito à capacidade humana de ser capaz de compreender as implicações técnicas e tirar conclusões para chegar a decisões informadas [28].

Cognitivamente, portanto, é interessante medir até onde um tomador de decisão humano está ciente da situação, ou seja, atingiu um certo nível de consciência

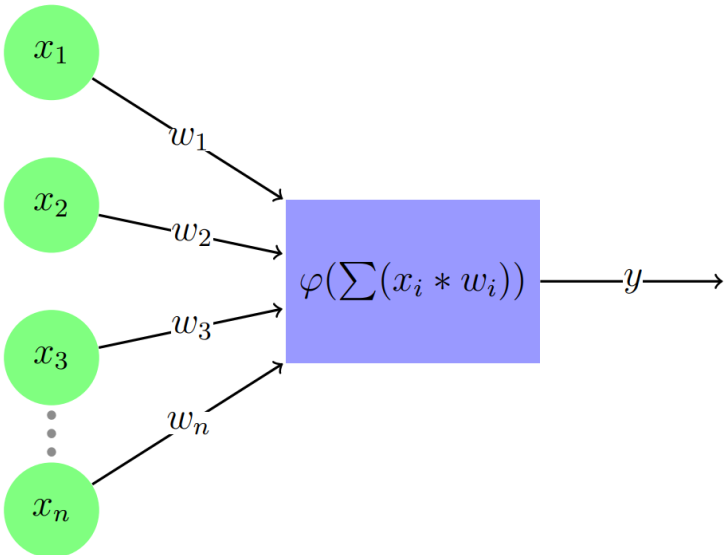
situacional, e quão bem ele consegue manter e desenvolver essa consciência com o passar do tempo [28].

Para avaliar em termos cognitivos o nível de CSA, pode-se considerá-la como "a percepção dos elementos do ambiente dentro de um volume de tempo e espaço, a compreensão de seu significado e a projeção de seu status em um futuro próximo". Destacam-se as palavras percepção, compreensão e projeção, que podem ser tomadas para denotar níveis de consciência progressivamente crescentes. A percepção sugere a tomada de conhecimento sobre dados importantes, a compreensão se refere interpretação e combinação de dados sumarizados, e projeção trata da capacidade de prever eventos futuros e suas implicações. Este último nível é o foco deste trabalho [28].

2.8 Redes neurais artificiais

A descoberta das redes neurais artificiais é atribuída a Walter Pitts e Warren McCulloch [29]. Em seu trabalho publicado em 1943, os autores buscavam uma forma de fazer com que os computadores pudessem executar o raciocínio lógico baseando-se nos neurônios biológicos [30]. Esse neurônio biológico é hoje representado matematicamente como ilustrado na Figura 2.1.

Figura 2.1 – Representação gráfica da estrutura de um neurônio artificial



Fonte: adaptado de [31].

Como pode ser visto nesta representação, existem três etapas de processamento em um neurônio. As n entradas, identificadas na figura pelos círculos verdes que compõem o vetor [x_1, x_2, …, x_n] são multiplicadas pelos seus respectivos pesos [w_1, w_2, …, w_n]. O resultado da soma acumulada dessas multiplicações serve como entrada para uma função de ativação φ escolhida para o neurônio. Por fim, temos a saída y, também conhecida como valor de ativação, como resultado do cálculo de φ.

Ao se basear no cérebro humano, as redes neurais artificiais são compostas por vários neurônios como o do modelo descrito acima interconectados formando uma rede. Cada um desses neurônios é ativado dependendo da intensidade dos estímulos que recebe, e o sinal pode ser transmitido deste para os outros neurônios da rede se o sinal de entrada tiver sido forte o suficiente para ativá-lo.

Dessa forma, os estímulos de entrada fornecidos aos neurônios iniciais da rede se propagam pelo mapeamento específico que cada neurônio particular realiza, mapeando um estimulo de entrada para um estimulo de saída, que servirá como entrada para outro neurônio. O processo se repete até a camada final de neurônios da rede, onde o resultado do processamento se torna conhecido [30],[31].

É importante observar que o poder das redes neurais não está no neurônio propriamente dito. Cada neurônio desempenha um papel relativamente simples, no entanto, é em conjunto que eles, formando uma rede, são capazes de executar as tarefas que tornam as redes neurais artificiais tão famosas. Expondo de outra forma, pode-se dizer que elas são baseadas na estratégia de divisão e conquista. Cada um dos muitos neurônios da rede resolve uma pequena parte do problema inicial que, por sua vez, só é de fato resolvido quando combinam-se as soluções de todos eles [31].

Com os avanços nas pesquisas envolvendo o tema, surgiram outras variantes de redes neurais além do modelo inicial proposto em [29]. Elas buscam resolver problemas de natureza mais complexa que não podem ser resolvidos por redes comuns, como por exemplo detecção de objetos em imagens, reconhecimento de voz, tradução textual, previsão de séries temporais, entre outros. As principais variações, também conhecidas como arquiteturas, aplicadas neste trabalho são descritas nas subseções 2.8.1 até 2.8.4.

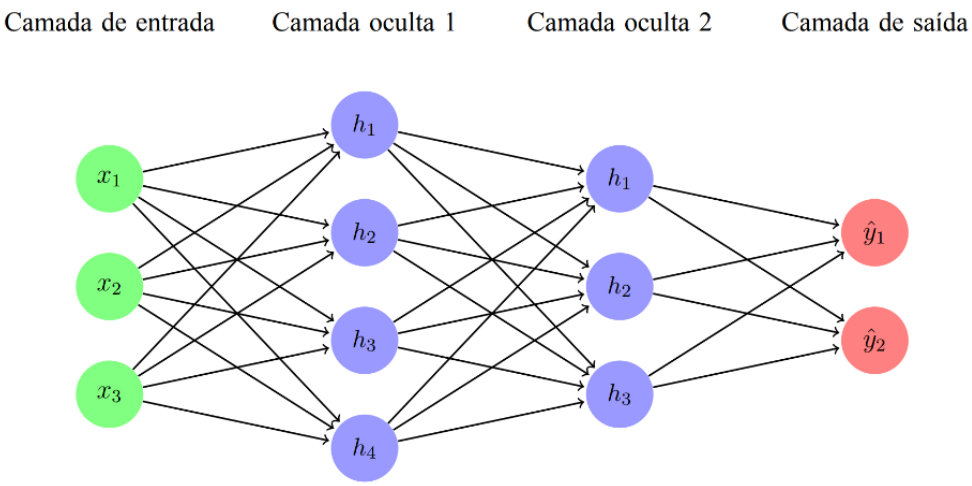
2.8.1 Perceptron multicamadas

Os neurônios artificiais inicialmente propostos em [29] eram divididos em dois grupos: os neurônios de entrada e os de saída. Inicialmente não haviam camadas ocultas, ou seja, o aprendizado profundo ainda não havia sido estabelecido [30].

Com o avanço do poder de processamento de *hardware*, foi possível que essas redes aumentassem em profundidade. Isso significa que, além do conjunto de neurônios de entrada e do conjunto de saída, tem-se a possibilidade de incluir uma quantidade arbitrária de camadas de neurônios intermediários, que compõem as chamadas camadas ocultas da rede.

Uma rede neural artificial totalmente conectada em que as conexões dos neurônios ocorrem somente no sentido da entrada para a saída dos dados é conhecida como *perceptron*. A Figura 2.2 ilustra um exemplo de uma rede de *perceptrons* com 2 camadas ocultas. Os componentes da camada de saída são geralmente representados por ŷ para diferenciá-los dos componentes de y, que representa o valor verdadeiro de previsão esperado para a saída da rede.

Figura 2.2 – Representação de uma rede neural com duas camadas ocultas



Fonte: elaborado pelo próprio autor.

Após a definição do número de camadas da rede, é necessário iniciar o processo conhecido como treinamento. Esse processo busca calibrar os pesos internos da rede com base em vários exemplos de dados de entrada e os seus respectivos valores-alvo obtidos a partir dos rótulos deles. O objetivo é que a atualização dos pesos torne os

valores de saída da rede cada vez mais próximos dos valores dos rótulos. Quando essa condição é atingida, a fase de treinamento termina e os pesos obtidos são então fixados para serem aplicados na fase de inferência [31].

Uma rede considerada bem treinada será aquela capaz de generalizar os padrões aprendidos na fase de treinamento quando for exposta a dados nunca vistos. Essa é a chamada fase de previsão ou inferência do modelo. O objetivo final de todo o processo é que a rede forneça bons resultados para esses dados novos, especialmente quando eles não forem obtidos do mesmo conjunto dos exemplos de treinamento.

2.8.2 Redes Convolucionais

Redes compostas por neurônios comuns apresentam desempenho insatisfatório quando a estrutura espacial ou temporal dos dados é importante para o processo de aprendizado. Essas informações estruturais importantes dos dados são simplesmente descartadas ao se achatar a entrada em um simples vetor unidimensional. Isso é particularmente ruim quando se objetiva um classificador de imagens, onde as informações de *pixels* vizinhos são importantes para o aprendizado da rede. As redes neurais convolucionais ou CNNs foram projetadas para resolver esse tipo de situação [31].

Embora projetadas inicialmente para tarefas de reconhecimento de imagens no campo de visão computacional, hoje as CNNs têm se mostrado eficientes em áreas antes dominadas por redes recorrentes como tarefas de reconhecimento de áudio, texto e análise de séries temporais.

O nome convolucional é devido à operação base realizada nelas. A convolução é um operador matemático empregado para a filtragem espacial numa matriz ou vetor de entrada. O processo consiste em mover uma máscara (ou *kernel*), que também é uma matriz ou vetor, rotacionada em 180º pela matriz de entrada, e calcular a soma dos produtos em cada posição.

2.8.3 Redes Recorrentes

Assim como ocorreu com as redes convolucionais, as redes neurais recorrentes foram pensadas para resolver um domínio específico de problemas. Além de não serem indicadas para dados espaciais, as redes *feedforward* (*perceptron*) também não apresentam bom desempenho na inferência de dados sequenciais, como é caso das séries temporais. Dados sequenciais demandam uma arquitetura que seja capaz de levar em consideração informações anteriores para fazer predições, num processo frequentemente associado à memória humana [31], [32].

A recorrência, que dá nome a esta arquitetura, é a propriedade que permite às redes recorrentes considerarem elementos anteriores para decidir a saída final. Os neurônios dessa rede são retroalimentados, o que significa que as saídas de uma camada podem ser ligadas à próxima camada e a ela mesma.

Essa característica também permite que as entradas fornecidas, isto é, as amostras, tenham tamanho arbitrário, pois a recorrência dá à rede a capacidade de iterar por cada elemento de uma entrada. Isso torna essa arquitetura naturalmente capaz de processar vídeos, textos, séries temporais e outros tipos de dados com comprimento naturalmente variável.

Embora essa capacidade seja interessante, as redes recorrentes possuem um grave problema conhecido como dissipação ou desaparecimento do gradiente (do inglês, *vanishing gradient*). Por serem arbitrariamente profundas por natureza, as redes recorrentes possuem a tendência de perderem as informações do gradiente no processo de treinamento. Quanto mais profunda a rede, maior será a perda. Este problema é solucionado modificando a unidade fundamental que compõe a rede. As duas propostas mais conhecidas são as unidades LSTM e GRU expostas a seguir.

A unidade de memória de curto prazo longa (do inglês, LSTM – *long short-term memory*) recebe esse nome pela forma como as memórias de curto e longo prazo são codificadas. Em uma rede neural comum, a memória de longo prazo é atingida ajustando os seus pesos ao longo do processo de treinamento. Após o treinamento, esses pesos não mudam, pois as inferências serão feitas com base neles. Por sua vez, a memória de curto prazo corresponde aos sinais de ativação que se propagam pela rede.

A unidade LSTM é projetada para que essas ativações fluam por longos períodos de tempo, isto é, elementos da entrada. Isso significa que a rede mantém uma matriz

de pesos compartilhada que é atualizada enquanto a rede itera sobre os elementos de uma entrada. Essa capacidade de propagação por longos períodos de tempo possibilita que este tipo de rede processe sequências de informações interdependentes de longa distância, como é o caso das séries temporais utilizadas neste desse trabalho.

Outra forma de contornar o problema de perda de dissipação do gradiente é a unidade recorrente fechada (do inglês, GRU – *gated recurrent unit*). Em comparação com as unidades LSTM, as GRUs são mais simples em termos arquiteturais, o que torna as redes compostas por essas unidades mais simples de serem treinadas, porém seu poder de inferência pode ser reduzido dependendo da complexidade dos dados.

A escolha de qual das arquiteturas será empregada é parte do processo de ajuste de hiperparâmetros[12] da rede. Ainda que as GRUs tenham um mecanismo de funcionamento mais simples que o das LSTMs, os resultados obtidos por elas podem ser superiores aos destas.

2.9 Aprendizado profundo

O aprendizado profundo, do inglês *deep learning*, surgiu a partir de pesquisas em inteligência artificial e aprendizado de máquina, sendo uma subárea desta dedicada à criação de modelos profundos de rede neural capazes de tomar decisões precisas baseadas em dados. Em termos técnicos, o termo aprendizado profundo se refere ao conjunto das redes neurais com duas ou mais camadas ocultas [31].

No início, muitas técnicas de aprendizado de máquina, por exemplo, *Support Vector Machines* (SVMs), apresentaram desempenho e eficácia significativos em muitas aplicações do mundo real. No entanto, esses algoritmos requerem que a extração de características seja feita por especialistas para produzirem bons resultados. Conforme a complexidade dos dados aumenta, também se torna cada vez mais desafiador fazer a extração adequada dos recursos a serem utilizados pelo modelo. Esses modelos são essencialmente superficiais, pois consistem em poucas camadas de

[12] O processo de treinamento de uma rede neural envolve uma série de parâmetros específicos a serem configurados além dos parâmetros internos, que são os seus pesos. Os principais hiperparâmetros são: a quantidade de camadas ocultas da rede, o otimizador, o número de épocas de treinamento, o número de amostras por lote. Uma visão mais detalhada e didaticamente descrita sobre os demais hiperparâmetros pode ser vista acessando https://d2l.ai

composição e, por esse motivo, fazem parte do chamado aprendizado superficial, do inglês *shallow learning* [31].

Para contornar os problemas do aprendizado superficial, os esforços de pesquisa em aprendizado de máquina voltaram-se ao aprendizado profundo. Essa técnica é adequada para contextos onde existem grandes volumes de dados complexos, e também quando os dados possuem padrões ocultos ou de difícil extração manual. A abordagem se baseia no conjunto usado no processo de treinamento para identificar e extrair padrões de forma automática, gerando um mapeamento preciso de informações relevantes que resultam em bons resultados de inferência. Pode-se afirmar, dessa forma, que o aprendizado profundo permite que as máquinas processem os dados de forma mais semelhante ao ser humano, estando assim mais próximo do objetivo final da inteligência artificial [31], [32].

2.10 Métricas de desempenho

Os algoritmos de aprendizado de máquina considerados nesse trabalho necessitam de medidas de desempenho para viabilizar a avaliação dos resultados e a comparação com os trabalhos relacionados. As métricas empregadas são aquelas já empregadas nos trabalhos relacionados. Dessa forma, são consideradas a acurácia, a precisão, a sensibilidade, a área sob a curva ROC, a medida-F e o erro quadrático médio [34], [35] e [36].

Ao classificar uma amostra, um classificador apresenta uma entre quatro possibilidades assim definidas:

- Verdadeiro positivo (VP): quando uma amostra pertencente a uma classe A é classificada como sendo da classe A;
- Verdadeiro negativo (VN): quando uma amostra não pertencente à classe A é classificada como não sendo da classe A;
- Falso positivo (FP): quando uma amostra não pertencente à classe A é classificada como sendo da classe A;
- Falso negativo (FN): quando uma mostra pertencente à classe A é classificada como não sendo da classe A.

A partir dessas definições tem-se as métricas de acurácia, precisão e sensibilidade.

A acurácia representa a proporção de acerto total do sistema. Esta medida é obtida conforme a Equação 2.1.

$$acurácia = \frac{VP + VN}{VP + VN + FP + FN} \qquad \text{(Equação 2.1)}$$

A precisão refere-se à capacidade do modelo de classificar amostras realmente negativas como negativas. O cálculo dessa métrica é dado pela Equação 2.2.

$$precisão = \frac{VN}{VN + FP} \qquad \text{(Equação 2.2)}$$

A sensibilidade indica a capacidade que o modelo possui de classificar amostras realmente positivas como positivas. Este indicador é calculado pela Equação 2.3.

$$sensibilidade = \frac{VP}{VP + FN} \qquad \text{(Equação 2.3)}$$

A medida-f (*F-score*) é a média harmônica entre a precisão e sensibilidade e também pode ser interpretada como o quão confiável é a acurácia. Na Equação 2.4 tem-se o cálculo dessa medida.

$$medida - F = \frac{2 * precisão * sensibilidade}{precisão + sensibilidade} \qquad \text{(Equação 2.4)}$$

A curva ROC (*Receiving Operating Characteristics*) é um gráfico bidimensional em que o eixo X apresenta as medidas do complemento da precisão, e o eixo Y as medidas de sensibilidade. Cada dupla de Especificidade e Sensibilidade corresponde a um ponto na curva, e cada ponto é obtido selecionando um limiar que

permite dizer se a predição realizada corresponde à classe positiva ou negativa. Quanto mais a curva ROC se aproxima do canto superior esquerdo do gráfico, melhor é o desempenho do modelo. A área sob a curva ROC (AUC – *Area Under ROC Curve*) é uma medida de desempenho que produz valores entre 0 e 1. Quanto mais próximo de 1 for resultado obtido, melhor será o classificador.

Por fim, o erro quadrático médio (MSE – *Mean Squared Error*), é uma medida útil para a avaliação do erro no processo de aprendizagem. Essa é frequentemente empregada na comparação de resultados de métodos de previsão de séries temporais pois. Diferente do que ocorre nas métricas anteriores, quando menor o valor da MSE, menor é a diferença entre os valores esperados e os valores preditos, o que significa um bom desempenho de predição. Na Equação 2.5 apresenta-se o cálculo do erro quadrático médio.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \qquad \text{(Equação 2.5)}$$

Em que o erro total em relação ao valor predito esperado y_i pode ser calculado pela distância entre o valor $\hat{y}_i$ e y_i, que é o valor predito pelo modelo. O resultado do somatório é então dividido pelas n amostras preditas.

CAPÍTULO 3 - Trabalhos Relacionados

3.1 Considerações iniciais

Neste capítulo são apresentados os trabalhos relacionados ao modelo de predição proposto. O interesse da comunidade de cibersegurança por estratégias de previsão de ataques tem crescido nos últimos anos [33]. Existe uma vasta quantidade de trabalhos voltados à predição de ataques que podem ser agrupados em uma das quatro categorias citadas na seção 2.6. No entanto, entre as várias propostas dedicadas ao tema, foram selecionadas apenas as que se baseiam em redes neurais, por serem as mais próximas da proposta deste trabalho.

3.2 Principais trabalhos relacionados

Os principais trabalhos relacionados ao modelo proposto são apresentados em [34] e [35]. Estes trabalhos propõem, respectivamente, um modelo capaz de prever o volume de ataques de determinado tipo numa *honeypot*, e um modelo capaz de classificar pares de IPs extraídos a partir de dados de fluxo entre classes de ataques pré-definidas.

Como apontado na seção 2.8.3, as redes neurais recorrentes se destacam entre todos os outros métodos do estado da arte com relação à previsão de séries temporais.

Esse fato é explorado em [34], onde os autores propõem um modelo baseado em LSTMs para prever o número de ataques a uma rede de *honeypots* nas próximas horas.

Mais precisamente, os autores propuseram uma variante bidirecional da arquitetura LSTM, em que a rede considera informações passadas e futuras na fase de treinamento. Essa rede bidirecional é a combinação de duas redes unidimensionais: uma aprende as informações do passado, e a outra as informações do futuro. O resultado das duas redes é combinado no cálculo da saída final.

Quanto aos dados, os autores afirmam usar um conjunto de fluxos obtidos de uma *honeypot* no período de aproximadamente um ano. A unidade de tempo considerada para os agrupamentos da série temporal é de uma hora. Isso significa que o objetivo é prever o número de fluxos considerados como ataques nas próximas n horas com base nos dados anteriores. Os autores também compararam o modelo proposto com métodos estatísticos como o modelo auto-regressivo integrado de médias móveis (ARIMA) e o modelo de heterocedasticidade condicional auto-regressiva generalizada (GARCH). A conclusão é que de fato o modelo proposto possui uma acurácia de predição superior à dos modelos estatísticos.

Diante da arquitetura geral do modelo de CSA que este trabalho compõe, o trabalho acima serve como base para a previsão do volume de ataque de cada uma das classes de ataques do sistema de monitoramento de ameaças. Com base nos volumes anteriores de cada classe de ataques detectada pelas assinaturas, o modelo é capaz de prever os próximos n volumes de ataque de acordo com a unidade de tempo considerada no agrupamento desses dados.

Considerando uma granularidade menor, as redes neurais também são capazes de classificar os dados de entrada. Nesse caso, a rede não prevê uma quantidade de valores futuros com base num comportamento histórico, mas, com base nas características de entrada, retorna em sua saída a probabilidade de uma amostra de entrada pertencer a uma das classes aprendidas durante o treinamento. Esta é a proposta de predição apresentada em [35].

Aplicando diferentes arquiteturas de redes neurais profundas, a saber: *perceptron* multicamadas, redes neurais recorrentes e redes LSTM, os autores compararam qual delas apresenta o melhor resultado de classificação de ataque considerando uma base de dados de uma competição de um evento de cibersegurança.

É importante destacar que o modelo proposto em [35] pode ser usado para qualquer tipo de ataque de rede, uma vez que os dados de entrada que compõem as características a serem aprendidas pela rede são apenas o par de endereços IP de origem e destino envolvidos no ataque. Isso é útil no contexto de uma rede de grande porte, onde é desejável prever qual é o tipo mais provável de ataque ao qual um par específico de endereços IP não vistos pela rede podem pertencer.

As métricas utilizadas nos resultados apontam que as redes recorrentes tiveram um desempenho superior em comparação à rede de *perceptron*. Entre as duas arquiteturas recorrentes, o melhor desempenho foi apresentado pela rede composta por unidades LSTM.

O modelo proposto neste trabalho se baseia nas duas abordagens dos trabalhos anteriormente citados. A proposta é obter um modelo de predição dividido em duas partes. A primeira é capaz de prever volumes futuros de ataques, e a segunda contempla a classificação de pares de IPs de origem e destino em uma das quatro classes de eventos maliciosos considerados.

É importante ressaltar que, assim como neste trabalho, os conjuntos de dados utilizados nos modelos são do mundo real. Os autores em [34] e [35] não relataram quaisquer modificações feitas nos dados com o objetivo de facilitar o treinamento dos modelos. Apenas dados incompletos ou não aplicáveis foram removidos do conjunto considerado nos processos que envolvem os modelos.

Outro destaque importante, presente em ambos os trabalhos, é a carência de abordagens que apresentem alta acurácia. Modelos estatísticos comuns, bem como baseados em aprendizado superficial [36], não atingiram bons resultados. Dessa forma, o emprego de métodos baseados em aprendizado profundo produziu uma acurácia superior em relação aos demais.

3.3 Trabalhos relacionados secundários

Embora os trabalhos da seção anterior sejam os mais próximos deste trabalho, existem ainda outros que influenciaram a escolha de outras arquiteturas de redes neurais. Embora os dados de entrada sejam de outra natureza, esses trabalhos também

ajudaram a alicerçar as demais características, considerações e o contexto do modelo proposto.

Em [36], os autores propuseram abordagens baseadas em aprendizado de máquina para predizer comportamentos maliciosos de diversas origens. Foram experimentados algoritmos de aprendizagem superficial e de aprendizado profundo.

Na categoria de aprendizagem superficial, os autores aplicaram o algoritmo de árvores de decisão impulsionadas por gradiente (GBDT) e redes neurais totalmente conectadas (NN), tendo o primeiro resultados ligeiramente melhores que o segundo. Para a aprendizagem profunda, aplica-se uma variante convolucional da rede neural recorrente LSTM (convLSTM).

Os autores afirmaram que as duas abordagens empregadas são complementares. O algoritmo de aprendizagem superficial é aplicado primeiro sobre os dados históricos da base de dados para prever os atacantes mais prováveis para o período em questão. Em seguida, o método baseado em aprendizado profundo é usado para estimar os alertas futuros. A combinação dos dois métodos dessa forma também é interessante, pois evita o desperdício de recursos computacionais, visto que os algoritmos de aprendizado profundo são consideravelmente mais custosos que os de aprendizado superficial.

Apesar de bons resultados, o trabalho foi construído sobre uma única base de dados estática conhecida como Warden, e as fases de treinamento e teste se limitaram apenas aos dados contidos neste conjunto. Os autores também apontaram que outras arquiteturas de redes neurais podem ser potencialmente aplicáveis nesse contexto, como redes neurais convolucionais de uma dimensão e redes de atenção.

Em [37], os autores propuseram um sistema chamado NERDS, que é semelhante ao proposto em [36]. Foram empregados apenas algoritmos de aprendizado superficial para estimar a probabilidade de uma determinada entidade repetir um mesmo ataque em um futuro próximo.

Embora os trabalhos apliquem os mesmos algoritmos, a saber, redes neurais totalmente conectadas e árvores de decisão impulsionadas por gradiente, o enfoque dado é diferente. Os autores em [37] expuseram três argumentos que motivaram a elaboração do modelo de aprendizado de máquina.

O primeiro é relacionado ao grande volume de alertas que os sistemas de segurança geram. Para que um analista tenha condições de lidar com todas essas informações, é necessário sumarizá-las e priorizá-las.

O segundo é baseado na suposição de que ataques recentes serão repetidos pelo mesmo invasor. Essa afirmação não deve ser generalizada, uma vez que é apenas válida para um determinado conjunto de invasores e tipos de ataque. Ataques frequentes podem ser bloqueados automaticamente, enquanto um ataque único, em um ativo crítico, pede uma investigação aprofundada.

O terceiro é que existem muitas fontes de dados de inteligência de ameaças que, além de conterem um alto volume de informações, possuem dados irrelevantes ou de baixa qualidade.

Os autores afirmaram que os problemas elencados acima podem ser resolvidos com um método adequadamente projetado para resumir informações de uma entidade mal-intencionada. Dessa forma, eles apresentaram uma pontuação de probabilidade de mau comportamento futuro, que pode ser entendida como o valor de predição que o algoritmo de aprendizado de máquina retorna para a possibilidade de um ataque se repetir futuramente.

O valor de pontuação pode ser utilizado para priorizar os alertas recebidos. Essa aplicação é útil principalmente no apoio do processo de tomada de decisão do administrador de rede, pois serve como parâmetro de quais casos devem ser resolvidos primeiro, especialmente para grandes infraestruturas.

Como um método de classificação de entidades de rede, a pontuação também pode ser empregada na mitigação de ataques de rede e análise de tráfego. Entidades com pontuação alta podem ser inseridas em uma lista de bloqueio onde todos os itens desta devem ter seu tráfego bloqueado. O valor obtido também pode ser pode ser útil como um dos fatores considerados no processo de decisão em filtros de *spam*[13], dispositivos de mitigação de DDoS ou quaisquer outros algoritmos que reconhecem tráfego malicioso.

Essas observações são diretamente importantes para o contexto deste trabalho uma vez que as arquiteturas empregadas não se limitam àquelas descritas em [34] e

[13] *Spam* é o termo usado para referir-se aos *e-mails* não solicitados, que geralmente são enviados para um grande número de pessoas. Quando o conteúdo é exclusivamente comercial, esse tipo de mensagem é chamada de UCE (do inglês *Unsolicited Commercial E-mail*). (Disponível em antispam.br/conceito)

[35]. Redes convolucionais e LSTMs convolucionais também foram consideradas na metodologia. Além disso, os argumentos apresentados em [37] estão incorporados no sistema de CSA que este trabalho compõe, visto que os fluxos sumarizam as informações da rede, e que as assinaturas não consideram informações de ataques anteriores e também filtram dados irrelevantes.

Embora tenham sido projetadas inicialmente para tarefas envolvendo imagens, as redes neurais convolucionais também podem ser empregadas em tarefas de detecção de ataques como apresentado em [7]. Os autores utilizaram-nas para detectar ataques DDoS e de *malwares* em uma rede com fluxo de dados *Netflow*. Diferente do que foi sugerido em [36], os autores mantiveram a estrutura bidimensional inicial da rede convolucional e converteram os dados de fluxo para uma representação de duas dimensões, empregando uma matriz de correlação circundante. Os resultados obtidos levaram esse tipo de arquitetura de rede neural a ser considerada no modelo de classificação proposto neste trabalho.

Em [33], é apresentado um sistema de previsão de eventos de segurança chamado Tiresias. O conjunto de dados foi formado a partir dos dados coletados durante 27 dias de 740 mil máquinas com produtos de prevenção de intrusão da Symantec instalados, totalizando 3,4 bilhões de eventos.

Os autores defenderam que a propriedade de memória de longo prazo das redes neurais recorrentes é fundamental para realizar de forma precisa a previsão de ataques complexos, com várias etapas e com algum tipo de ruído. Ao se comparar o modelo proposto com outros métodos de previsão, os autores apontaram que as redes neurais se destacaram por possuírem memória de longa duração e a capacidade de filtrar ruídos.

Outro aspecto importante é a influência da duração do período de treinamento. Para verificá-lo, o modelo foi treinado primeiramente com amostras referentes a um dia, e depois com amostras de uma semana. Nos dois casos, o modelo foi avaliado em um período de dez dias. Os resultados apontaram que a precisão média das previsões realizadas pelo modelo treinado no período de uma semana foi apenas 0,3% superior em relação ao treinamento em apenas um dia. Essa informação também é importante para o modelo de predição de volume de ataques proposto, uma vez que é desejável que a unidade de tempo possa ter alguma variação.

Outra limitação é inerente ao conjunto de dados usado. A coleta de dados é passiva, ou seja, os registros correspondem apenas a eventos de segurança captados pelas assinaturas dos *softwares* da Symantec. Eventos bloqueados por outros *softwares* ou que não correspondam a nenhuma assinatura não são registrados.

Como descrito na seção 2.7, as técnicas de previsão de ataques compõem o terceiro nível do modelo de consciência situacional em cibersegurança, em que o objetivo é obter uma visão holística da rede administrada por meio de três estados, a saber: percepção, compreensão e projeção. Em [39], os autores propõem um sistema de conscientização situacional da rede chamado YHSAS. Ele opera com base em grande aquisição de dados e tecnologia de gerenciamento de armazenamento, utilizando métodos de análise de dados, mineração e dedução inteligente para, entre outras tarefas, prever incidentes de segurança e seu desenvolvimento.

Em relação às funções de previsão de ataques, os autores afirmaram que o sistema é capaz prever a propagação de ataques por cavalos de Tróia, DDoS, vírus, *botnets* e ataques APT[14]. Eles também argumentaram que, devido ao contexto de CSA, é difícil prever a situação de segurança da rede com apenas uma tecnologia de previsão.

Dessa forma, os autores lançaram mão de uma arquitetura de previsão que combina vários métodos de previsão. Para o campo de segurança de rede com dados baseados em séries temporais, as previsões podem ser de curto, médio e longo prazo. Cada uma delas necessita de um período de tempo proporcional de histórico de eventos para produzir bons resultados.

3.4 Conclusões sobre os trabalhos relacionados

Embora os trabalhos analisados elaborem soluções de previsão de ataques para contextos diferentes, há um consenso em relação à crescente necessidade da elaboração de métodos efetivos. Os ataques têm se tornado cada vez mais sofisticados, sendo realizados em cada vez mais passos e passam cada vez mais desapercebidos pelos sistemas de detecção.

[14] *Advanced Persistent Threat*: Uma Ameaça Persistente Avançada é um ataque no qual um usuário não autorizado obtém acesso a um sistema ou rede e permanece lá por um longo período de tempo sem ser detectado. (Disponível em digitalguardian.com/blog/what-advanced-persistent-threat-apt-definition)

Existem diversos trabalhos atuais [5] que aplicam métodos baseados nas outras três categorias descritas anteriormente. No entanto, esses métodos são inviáveis para o contexto desse trabalho, seja pela baixa taxa de detecção, seja pela ineficiência deles atuando sobre os dados de fluxo. Por este motivo, os trabalhos voltados a algoritmos de aprendizado de máquina foram selecionados por se adaptarem melhor aos dados de entrada utilizados neste trabalho.

A Tabela 3.1 sumariza as principais informações dos trabalhos considerados nas seções 3.2 e 3.3.

Tabela 3.1 – Resumo comparativo dos trabalhos relacionados

Trabalho	**Dado de entrada**	**Métrica**	**Rede utilizada**
34	Fluxos	MSE=935,07	LSTM bidirecional
35	Fluxos	F-score=93,13%	LSTM
36	Base de eventos	AUC=87%	*Perceptron*
37	Base de eventos	AUC=93%	*Perceptron*
33	Eventos de IDS	Precisão=93%	LSTM

Fonte: elaborado pelo próprio autor.

Apenas [34] e [35] usaram dados de fluxos de fato e, ainda assim, estes são de um volume consideravelmente menor do que o da rede considerada pela metodologia proposta. A métrica considerada para cada trabalho é a que foi obtida para o conjunto de dados com melhor desempenho informado pelos autores, no caso de o trabalho apresentar mais de uma base de dados.

Observa-se que [34] é o único trabalho com o qual seria possível comparar o modelo de previsão de volumes de ataque. No entanto, a métrica empregada pelos autores possui valores elevados pois o conjunto de dados de entrada da rede não foi normalizado, diferente do que ocorre no modelo equivalente proposto neste trabalho. Assim, embora a métrica seja a mesma, os valores não podem ser diretamente comparados.

Diante do exposto, é notória a necessidade de um modelo de previsão de ataques que possua bom desempenho e seja capaz de atuar em um contexto de grandes volumes de dados. Poucos trabalhos apresentam fluxos como entrada do

processamento das redes, e nenhum assim o faz com um grande volume de dados. O modelo proposto é a combinação de dois submodelos de predição, o que torna sua contribuição ainda mais relevante para o estado da arte de modelos de predição de ataques de redes de computadores.

CAPÍTULO 4 - Metodologia

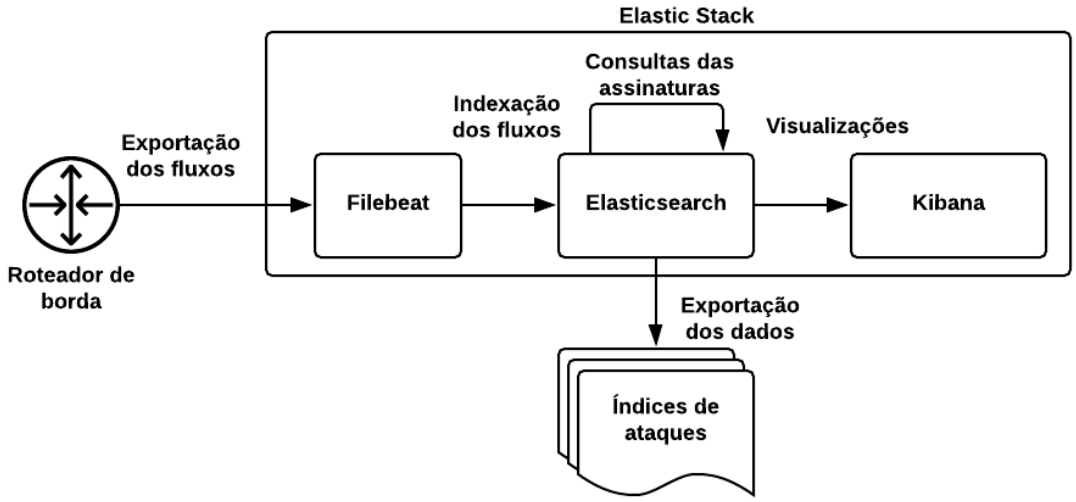
4.1 Considerações iniciais

Neste capítulo, é apresentada a metodologia utilizada para a elaboração do modelo de previsão de ataques proposto. Na seção 4.2, é apresentada a visão geral do sistema de CSA que o modelo proposto neste projeto compõe, bem como seu funcionamento geral; na seção 4.3, descreve-se os dois submodelos que compõem o modelo proposto; na seção 4.4, são descritas as etapas de preparação dos dados de entrada do modelo; na seção 4.5, o submodelo responsável pela classificação de ataques é apresentado em detalhes; na seção 4.6, o submodelo responsável pela previsão do volume de ataques é descrito e, por fim, na seção 4.7 tem-se as considerações finais sobre a metodologia deste trabalho.

4.2 Visão geral do sistema de CSA

Este trabalho compõe um sistema completo de CSA desenvolvido em conjunto com outro membro do laboratório ACME! da UNESP de São José do Rio Preto. Na Figura 4.1 é ilustrado um esquema simplificado da arquitetura desse sistema.

Figura 4.1 – Arquitetura do modelo de CSA



Fonte: elaborado pelo próprio autor

A entrada de dados do sistema consiste exclusivamente de dados de fluxo *Netflow* V9. O roteador de borda localizado na reitoria da universidade exporta os fluxos de toda a rede da universidade para o laboratório ACME!. Os dados brutos ocupam em média 50 GB de armazenamento dos servidores do *Elastic Stack* diariamente. Isso equivale a aproximadamente 2.880.000 fluxos recebidos por hora.

Esses dados são coletados pelo primeiro componente da pilha, o *Filebeat*. Ele é responsável por preparar os dados brutos recebidos de fluxo, mapeando-os e enriquecendo-os com informações geográficas, quando disponíveis. Em seguida, esses dados são indexados pelo *Elasticsearch*.

No *Elasticsearch*, os dados processados são indexados. Este é o componente central da pilha, que de fato armazena as informações de maneira apropriada para que as consultas sejam executadas. A geração de consultas e assinaturas é facilitada pelo componente de visualização *Kibana*, que auxilia na visualização dos dados indexados por meio de diversos tipos de visualizações gráficas. Na Figura 4.2 tem-se a ilustração de um entre os vários exemplos de visualizações possíveis.

Figura 4.2 – Exemplo de visualização de dados do *Kibana*

Fonte: extraído do servidor onde a *Elastic Stack* está instalada

Por meio dos recursos oferecidos pelo *Kibana*, as assinaturas de ataques foram definidas e ajustadas. O *Elasticsearch* possui uma API poderosa para realizar consultas em seus índices. As assinaturas definidas para o sistema de CSA exploram esses recursos em suas definições. O objetivo de cada assinatura de ataque ou evento malicioso é detectar sua ação com base nos fluxos recebidos do roteador da universidade. No momento em que uma assinatura detecta uma atividade maliciosa, esta é registrada no índice do *Elasticsearch* que corresponde a ela.

Com os índices contendo os registros históricos de cada assinatura devidamente preenchidos, esses dados são então exportados para arquivos no formato json. Esses arquivos formam os conjuntos de dados utilizados para o treinamento das redes neurais.

4.3 Visão geral do modelo proposto

Com o intuito de auxiliar administradores de rede na tomada de decisão de contramedidas contra futuras ameaças na rede, a proposta do modelo de previsão de ataques deste trabalho é na verdade dividida em dois submodelos.

O primeiro submodelo é uma rede neural voltada para a tarefa de classificação. Como base em um IP de origem e um IP de destino, a rede é capaz de prever qual é o tipo de ataque ou atividade maliciosa envolvida entre as quatro classes consideradas.

O segundo submodelo é uma rede neural capaz de prever os próximos volumes de um tipo de ataque a serem realizados na rede ao longo de n unidades de tempo futuras.

Antes do treinamento e definição das arquiteturas para cada um dos submodelos, é necessário que os dados sejam preparados. Este processo é descrito na seção 4.4.

4.4 Preparação dos dados

Antes de serem devidamente inseridos como amostras de treinamento para as redes neurais, estes conjuntos de dados de eventos maliciosos detectados precisam ser devidamente processados para a aprendizagem das redes neurais ocorra adequadamente. O Algoritmo 4.1 descreve o processo de preparação dos dados para cada índice exportado do *Elasticsearch*.

Algoritmo 4.1 – Preparação dos dados do índice exportado

Entrada: index.json

1: BD ← obterApenasCaracterísticas(index.json)

2: BD ← removerEndereçosIPv6(BD)

3: BD ← dividirIPsEmOctetos(BD)

4: BD ← normalizarColunas(BD)

Saída: BD

Os dados originais exportados de cada um dos índices do *Elasticsearch* vêm no formato json, com algumas informações de indexação, como por exemplo identificadores, que não são relevantes para o modelo. A linha 1 representa o processo de remoção desses dados irrelevantes, mantendo apenas as colunas que representam alguma característica específica do ataque que seja relevante para o treinamento.

Em seguida, as linhas contendo os endereços IPv6 são removidas pois existem poucas amostras, que além de não poderem ser codificadas da mesma forma que as amostras de IPv4, não possuem um volume suficiente para constituírem uma base de dados à parte.

A maioria das colunas de cada índice corresponde a um valor quantitativo obtido pelas consultas. Para o índice de ataques de varredura, tem-se o número de portas que foram varridas. Para o índice de força bruta, tem-se o número de tentativas de acesso a determinada porta. No entanto, todos os índices possuem um IP de origem e um IP de destino que precisam ser apresentados de uma forma que as redes neurais possam compreender.

Entre as opções mais comuns testadas em [38], o autor concluiu que dividir o endereço IP em quatro octetos é a melhor forma de se apresentar este tipo de dado para a rede neural. Dessa forma, no modelo proposto, os IPs de origem e destino são divididos usando esta estratégia, o que totaliza oito octetos.

Por fim, as colunas restantes são normalizadas como a última etapa de preparação antes do treinamento do modelo. O algoritmo é executado para cada um dos quatro índices considerados neste trabalho, a saber: ataques de força bruta, ataques de varredura, comunicações potencialmente maliciosas envolvendo IPs comprometidos, e comunicações com mineradores de criptomoedas.

4.5 Rede de classificação de ataques

Com os conjuntos de dados devidamente limpos e preparados, a rede neural de classificação pode ser definida. Os únicos campos comuns a todos os índices são os endereços IP e a data do evento. A data não é considerada para a classificação. Dessa forma, os únicos dados mantidos de cada um dos conjuntos de dados são os IPs de origem e destino de cada amostra.

Considerando que o aprendizado ocorre de forma supervisionada, é necessário fazer a rotulagem desses dados. Além disso, cada uma das classes pode conter um número diferente de amostras. Esse desbalanceamento pode prejudicar o desempenho futuro da rede na etapa de inferência. O Algoritmo 4.2 descreve o processo de preparação utilizado neste classificador.

Algoritmo 4.2 – Preparação dos dados para a classificação

```
     Entrada: lista: lista de arquivos contendo os dados de cada classe
1:   min ← 0
2:   para cada classe c em lista:
3:       se numeroAmostras(c) > min:
4:           min ← numeroAmostras(c)
5:   lista ← removerAmostrasAleatoriamente(lista, min)
6:   conjuntoRotulado ← rotularDados(lista)
     Saída: conjuntoRotulado
```

As linhas de 1 a 4 servem para determinar qual é o conjunto com o menor número de amostras. Na linha 5, o procedimento removerAmostrasAleatoriamente elimina aleatoriamente o número de amostras necessárias de cada conjunto, de modo que, ao final, todos possuam o mesmo número de amostras do menor conjunto.

Dessa forma, os conjuntos resultantes são combinados em uma única base de dados final que servirá de entrada para o treinamento do modelo proposto. A rotulagem da linha 6 é feita com base no conhecimento prévio dos arquivos de entrada que compõem o conjunto de dados final.

Os trabalhos considerados no capítulo 3 propõem diversos tipos de arquiteturas de redes neurais artificiais. Dessa forma, para este trabalho, foram consideradas as seguintes arquiteturas, com os parâmetros relevantes de cada uma, para o modelo de classificação:

- *Perceptron* multicamadas: primeira camada com 120 neurônios e camada oculta com 40 neurônios;
- Convolucional com uma dimensão: 64 filtros e máscara de tamanho 4;
- LSTM: 50 unidades;
- GRU: 50 unidades;

- LSTM bidirecional: 50 unidades;
- GRU bidirecional: 50 unidades;
- Convolucional LSTM: 64 filtros e máscara de tamanho 1.

Todas elas possuem uma camada final com quatro neurônios, referentes a cada uma das classes em questão. A melhor configuração encontrada para os demais hiperparâmetros foi a seguinte:

- 5 épocas de treinamento;
- 100 amostras por lote de treinamento;
- Otimizador Adam;
- Entropia cruzada categórica como função de avaliação de perda;
- 20% do conjunto de treinamento inicial reservado para validação.

O Algoritmo 4.3 sumariza o processo de treinamento e obtenção dos resultados para cada arquitetura proposta. O objetivo final é obter uma lista com os resultados do treinamento e validação de cada arquitetura implementada.

Algoritmo 4.3 – Treinamento das redes neurais do classificador

Entrada: conjuntoRotulado, hiperparâmetros, arquiteturas

```
1:    para cada arquitetura a em arquiteturas:
2:        resultados[a] ← treinarValidar(a, hiperparâmetros, conjuntoRotulado)
```

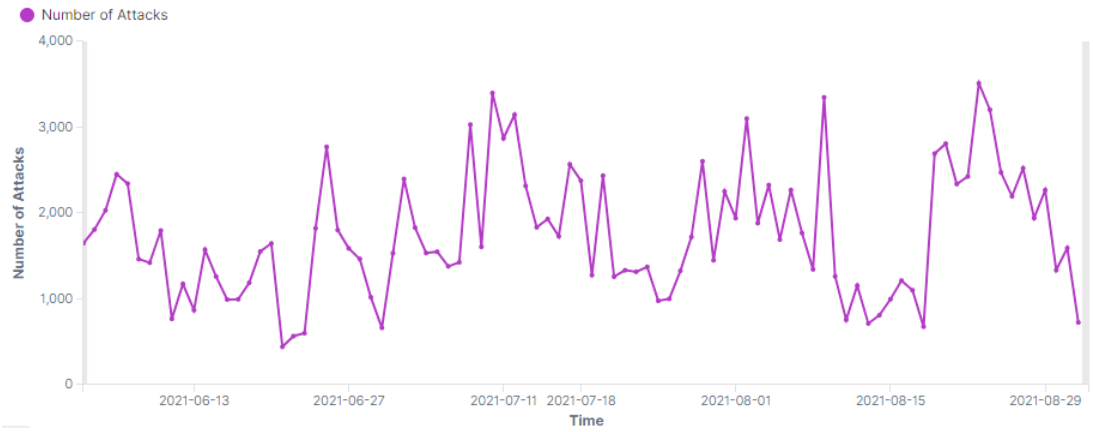
Saída: resultados

4.6 Rede de previsão de volume de novos ataques

O submodelo para classificação de ataques é útil para situações de classificação. No entanto, outro cenário em que é interessante a aplicação de redes neurais é a previsão de séries temporais. Devido ao grande volume de tráfego gerado diariamente por uma rede do porte da UNESP, é esperado que um número considerável de ataques ocorra.

Cada tipo de ataque tem uma frequência de ocorrência. Na Figura 4.3 tem-se um exemplo do volume de ataques de força bruta registrado num período de 90 dias. Em termos visuais, o objetivo é que a rede neural recorrente preveja a continuação da linha, avançando tantos termos no futuro quanto se queira.

Figura 4.3 – Volume de ataque de força bruta num período de 90 dias



Fonte: elaborado pelo próprio autor

As redes neurais recorrentes são o tipo de arquitetura mais indicada para esta tarefa. Embora redes convolucionais também sejam capazes de realizar esse tipo de inferência, os seus resultados são inferiores aos das RNNs [32]. Isso ocorre porque a estrutura das redes recorrentes suporta naturalmente entradas de dados sequenciais.

As unidades de LSTM são capazes de prever a continuação de séries temporais que *perceptrons* multicamadas ou outros métodos estatísticos não conseguiriam. Isso significa que a propriedade de memória de longo prazo dessas redes aprende longas dependências e correlações temporais. As demais abordagens tendem a produzir resultados que pioram muito rapidamente conforme é exigido um longo horizonte de tempo.

Quanto ao processo de treinamento, a rotulação dos dados é diferente. O objetivo é que, dada uma sequência de n elementos anteriores, a rede preveja o valor do elemento n+1. Isso é feito gerando um conjunto de elementos de treinamento a partir da própria série de eventos passados.

Considerando uma sequência inicial com n elementos, utiliza-se uma janela de tempo com tamanho fixo para gerar os dados de treinamento e os rótulos. Considerando uma janela de tamanho m, inicia-se o processo de extração das amostras

e rótulos a partir do intervalo inicial de eventos passados de 0 até m da sequência inicial. Nesse caso, o valor a ser previsto é aquele que está na posição m+1. A segunda amostra corresponde ao intervalo de 1 a m+1, com o rótulo sendo o valor armazenado em m+2, e o processo de extração segue até o final da sequência inicial.

Por sua vez, a sequência de dados inicial que dá origem ao conjunto de treinamento é obtida agregando-se os ataques de acordo com uma unidade de tempo. Agrupar os ataques ocorridos ao longo de um mês significa que a rede neural treinada com esses dados fará previsões para os próximos meses. Agrupar os ataques ocorridos de hora em hora faz com que a rede treinada com esses dados preveja volumes de dados nas próximas horas. No Algoritmo 4.4, são apresentados os passos para a geração dos dados agrupados e dos dados de treinamento para a rede recorrente.

Algoritmo 4.4 – Preparação do conjunto de treinamento da rede recorrente

Entrada: baseAtaque, unidadeTempo, tamanhoHistorico

1: baseAtaque ← ordenarPorData(baseAtaque, ordem=ascendente)

2: serieTemporal ← agruparPorUnidadeTempo(unidadeTempo)

3: conjuntoRotulado ← gerarConjunto (serieTemporal, tamanhoHistórico)

Saída: conjuntoRotulado

Diferente do procedimento de preparação do classificador, para este conjunto de treinamento apenas as datas de cada amostra são relevantes. Como o objetivo é obter uma série temporal a partir desses dados, é necessário garantir na linha 1 que os dados estejam ordenados por ordem ascendente de data. Assim, é possível agrupá-los por unidade de tempo na linha 2, gerando a série temporal com base nesse agrupamento. Por fim, basta preparar o conjunto de treinamento seguindo o processo descrito anteriormente baseado numa janela fixa de tempo.

A unidade de tempo nem sempre pode ser escolhida livremente. Prever o volume de meses ou anos implicaria ter um histórico de dados guardados há um tempo passado suficiente para que a rede pudesse aprender os padrões desses dados. Neste trabalho, o histórico é de aproximadamente um ano de coleta de ataques, o que torna inviável a previsão de meses e anos futuros. Dessa forma, a previsão se concentra em dias e horas futuros.

4.7 Considerações finais

Neste capítulo foram apresentadas as características de funcionamento do modelo proposto. Um panorama da arquitetura do sistema de CSA principal foi apresentado para que se pudesse compreender onde o modelo proposto é útil. As questões referentes à preparação de dados foram apresentadas, de modo que se pudesse compreender como os dados vindos do *Elasticsearch* são transformados nas bases de dados utilizadas no treinamento das redes neurais. Por fim, foram definidas as características e os detalhes de funcionamento dos dois submodelos propostos para a previsão de ataques.

CAPÍTULO 5 - Resultados

5.1 Considerações iniciais

Neste capítulo são apresentados os resultados obtidos na fase de treinamento e teste das redes neurais descritas na metodologia. Todos os testes foram realizados no ambiente online Google Colab, com dois núcleos do processador Intel Xeon com frequência de 2.3GHz, 12GB de memória RAM e aproximadamente 40GB de armazenamento. A placa de vídeo utilizada foi a Nvidia Tesla K80. Na seção 5.2, são apresentados os resultados para o submodelo responsável por classificar ataques; na seção 5.3 são apresentados os resultados obtidos para o submodelo dedicado a prever volumes futuros de cada tipo de ataque considerado; por fim, na seção 5.4 são feitas considerações gerais sobre os resultados obtidos nos dois submodelos.

5.2 Classificador de ataques

Após preparados, os quatro conjuntos de dados possuíam inicialmente a seguintes quantidades de amostras: 153400 amostras de ataques de varredura, 241644 amostras de ataques de força bruta, 354403 amostras de comunicações com IPs comprometidos e 176897 amostras de comunicações com IPs de mineradores. Após a

exclusão aleatória de amostras, todos os conjuntos passaram a ter 153400 amostras, que é número de amostras de conjunto de ataques de varredura, o menor deles.

O conjunto total com as quatro classes juntas possui 613600 amostras e foi dividido nos subconjuntos de treinamento e teste. O conjunto de treinamento recebeu 490880 amostras, e o conjunto de teste 122720 amostras, isto é, 20% do conjunto original foi reservado para os testes. Uma parte do conjunto de treinamento foi reservada para a validação durante o treinamento, que correspondeu a 98176 amostras, ou seja, 20% do conjunto de treinamento.

Todas as principais métricas dos trabalhos relacionados foram consideradas a fim de tornar possível a comparação com os mesmos. A tabela 5.1 sumariza os resultados obtidos no treinamento de cada arquitetura proposta para o classificador.

Tabela 5.1 – Resultados do treinamento das redes neurais consideradas

Arquitetura	**Acurácia**	**Precisão**	**Sensib.**	**F-score**	**AUC**	**Tempo**
Perceptron	**97%**	**97%**	**97%**	**97%**	**99%**	**2,46min**
Conv.	95%	95%	95%	95%	99%	2,88min
LSTM	87%	87%	87%	87%	98%	3,08min
GRU	95%	95%	95%	96%	99%	2,90min
LSTM bid.	**98%**	**98%**	**98%**	**98%**	**99%**	**4,03min**
GRU bid.	**98%**	**98%**	**98%**	**98%**	**99%**	**3,75min**
ConvLSTM	94%	94%	94%	94%	99%	34,20min

Fonte: elaborada pelo próprio autor.

A partir desses resultados, conclui-se que com exceção da rede LSTM, as demais redes possuem medidas de desempenho próximas. Os melhores desempenhos foram alcançados pelas redes bidirecionais. A rede convolucional LSTM necessitou de um tempo significativamente maior que as demais redes para ser treinada, mas não apresentou resultados proporcionais ao tempo consumido.

Considerando a relação entre custo e benefício, a rede de perceptrons é a mais indicada entre as demais por ter o segundo melhor desempenho e o menor tempo de treinamento. O tempo de gasto para treinar cada rede é um fator importante, considerando a necessidade de alteração nos dados do conjunto de treinamento.

Os valores obtidos aplicando as redes no conjunto de testes são similares aos resultados obtidos na fase de treinamento. A Tabela 5.2 representa esses valores.

Tabela 5.2 – Resultados dos testes das redes neurais consideradas

Arquitetura	**Acurácia**	**Precisão**	**Sensib.**	**F-score**	**AUC**
Perceptron	97%	97%	97%	97%	99%
Conv.	95%	95%	95%	95%	99%
LSTM	88%	88%	88%	88%	98%
GRU	96%	96%	96%	96%	99%
LSTM bid.	98%	98%	98%	98%	99%
GRU bid.	98%	98%	98%	98%	99%
ConvLSTM	95%	95%	95%	95%	99%

Fonte: elaborada pelo próprio autor.

Estes resultados comprovam certos aspectos da literatura. Assim como em [35], a rede de perceptrons apresentou desempenho inferior em relação aos classificadores baseados em RNNs. Apenas a área sob a curva ROC se manteve praticamente a mesma em todos os classificadores, não sendo esta uma boa medida de comparação entre eles. Em geral, observa-se que arquiteturas mais complexas, isto é, as CNNs e RNNs são mais apropriadas para a classificação de ataques com base nos IPs de origem e destino.

Embora fosse esperado conforme [33], [34], [35] e [36] que a rede recorrente com a unidade LSTM obtivesse os melhores resultados, isso não ocorreu. No entanto neste problema a unidade GRU se mostrou 8% mais eficiente na classificação em comparação com a LSTM. Embora esse resultado não seja o mais comum, segundo [32] a unidade GRU pode se destacar em alguns casos dependendo do tipo de dados.

As variantes bidirecionais apresentaram os melhores resultados. Embora tenham sido aplicadas em [34] na previsão de volume de ataques, essas arquiteturas se mostraram eficientes até mesmo quando nenhuma informação sobre o histórico dos dados é fornecida. Esse resultado indica que a técnica bidirecional confere mais robustez na classificação.

Embora as métricas indiquem resultados elevados de classificação, as redes com componentes convolucionais não apresentaram os melhores desempenhos. A

arquitetura convolucional LSTM possui ainda o agravante do custo elevado de tempo de treinamento, o que a torna o pior classificador entre todos os considerados.

A proposta de classificação apresentada em [35] é a mais próxima do classificador proposto. O melhor resultado obtido pelos autores foi em uma rede recorrente com medida-F igual a 93,13%, enquanto o classificador deste trabalho apresentou o valor de medida-F igual a 97%. Vale ressaltar que, embora a entrada inicial de ambos os trabalhos seja dados de fluxo, o presente trabalho obtém os dados rotulados por meio de assinaturas previamente definidas; já em [35] os dados foram rotulados por um sistema de detecção de intrusão.

Outro aspecto que contribui para a variação entre o resultado das métricas entre os dois trabalhos é o número de classes. Em [35], devido à rotulagem obtida pelo IDS, foram consideradas 36 classes de ataques, enquanto o classificador proposto considera apenas 4 classes. Quanto maior o número de classes, mais difícil se torna a classificação [31], [32], com o agravante que apenas o IP de origem e o IP de destino foram considerados como características. Assim, é justificável que o classificador proposto em [35] apresente uma valor de medida-F ligeiramente inferior.

5.3 Previsão de volume de ataque

Diferente do que ocorre no classificador, na previsão de volume cada uma das classes é considerada em particular para se realizar a predição. A Tabela 5.3 resume os dados obtidos no processo de treinamento e teste para cada classe.

Tabela 5.3 – Resultados do modelo de predição de volume de ataque

Base de dados	**Contagem de dias**	**Contagem de horas**	**Média quadrada do erro**	
			Treinamento	**Teste**
Varredura	316	4763	0,0027	$5{,}63 \times 10^{-4}$
Força bruta	320	7112	0,0015	0,0767
Comprometidos	53	916	0,0071	0,0122
Mineradores	74	1678	0,0011	$6{,}05 \times 10^{-5}$

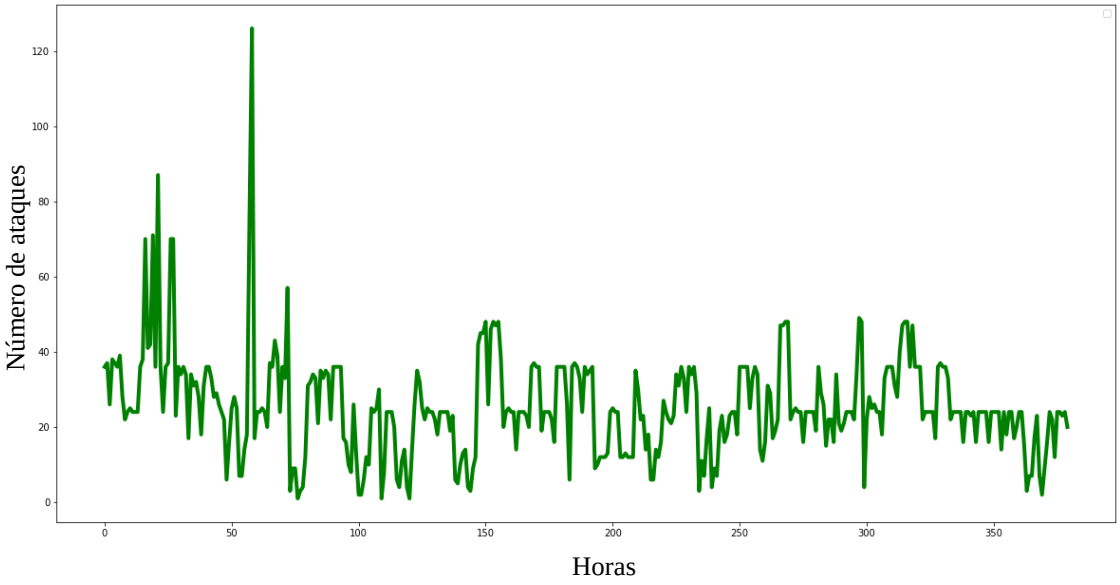
Fonte: elaborado pelo próprio autor.

Devido à complexidade intrínseca do problema de previsão de séries temporais, foram necessárias 200 épocas de treinamento para que houvesse melhores resultados. Isso ocorre pela necessidade da rede ser exposta várias vezes ao conjunto de dados para que ela consiga de fato compreender os padrões da série temporal que deve ser predita. A melhor configuração arquitetural foi de uma rede com duas camadas, sendo a primeira com a unidade LSTM e a segunda com a unidade GRU, ambas com 100 unidades.

Assim como em [34], a melhor unidade de tempo a ser escolhida são horas, pois agrupar o conjunto em dias resulta em uma série temporal com poucos elementos. Embora seja possível prever uma série dessa forma, os resultados tendem a ser comprometidos, mesmo usando redes neurais recorrentes.

Na Figura 5.1 tem-se a representação gráfica de um recorte da série temporal gerada para o conjunto de dados de varredura.

Figura 5.1 – Recorte da série temporal gerada a partir dos dados de varredura



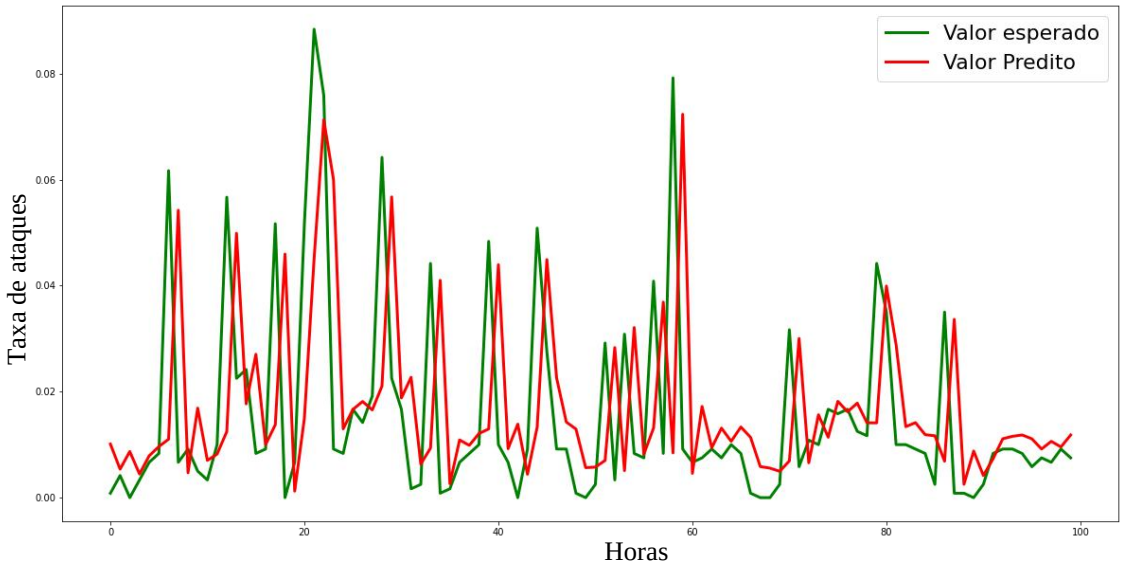
Fonte: elaborada pelo próprio autor.

Após gerada, a série passa pelo processo de geração das amostras descrito na seção 4.6, e então o processo de treinamento é iniciado. Na Figura 5.2 tem-se a representação de uma parte da predição do subconjunto de testes do conjunto de varredura.

Como definido na legenda da Figura 5.2, a linha vermelha representa o valor predito pelo modelo, e a linha verde representa o valor esperado. O recorte que ela representa corresponde a 100 horas de predição, isto é, após ser treinada com os dados de volumes de ataques de varredura passados, a rede é capaz de prever qual será a continuação do volume de ataques.

O trabalho que mais se aproxima da proposta desse submodelo é o presente em [34]. No entanto, o erro médio quadrático, que é a métrica usada em ambos os trabalhos para aferir os resultados de predição das séries temporais, é calculada a partir de um conjunto de dados previamente normalizado, conforme discutido na seção 4.4. Dessa forma, enquanto a menor MSE entre os subconjuntos de dados em [34] foi igual a 935,07, neste trabalho tem-se o menor valor igual a $6{,}05 \times 10^{-5}$ devido a normalização inicial. No entanto, como a MSE só pode ser considerada como medida de comparação para o mesmo conjunto de dados, o confronto entre esses valores com relação ao desempenho da predição não é factível.

Figura 5.2 – Valor predito e valor esperado do conjunto de varredura

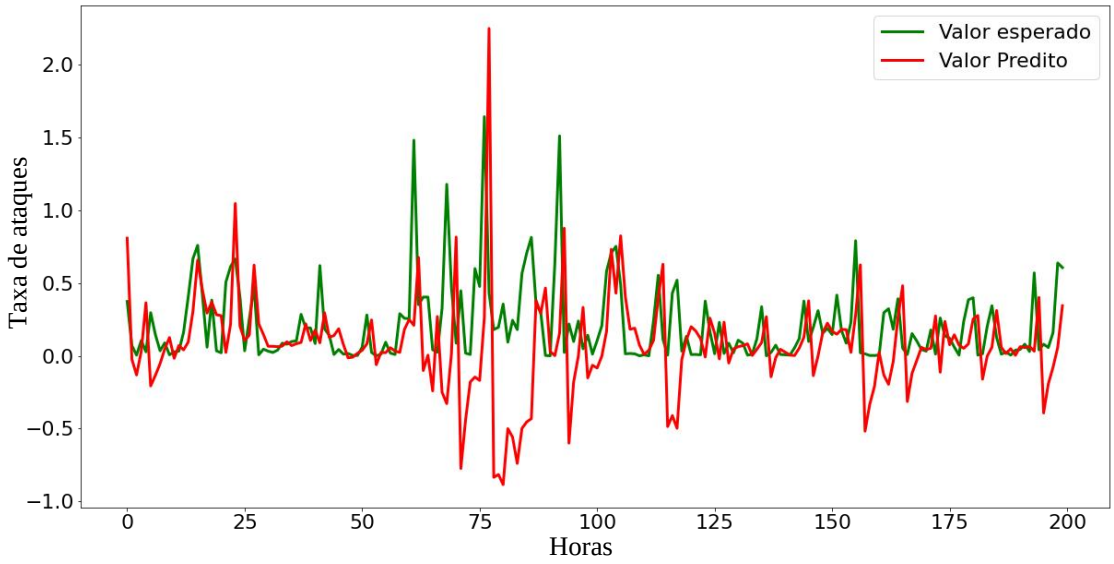


Fonte: elaborado pelo próprio autor

Considerando o nível de complexidade inerente ao problema, nota-se que o modelo é capaz de fornecer boas estimativas para os volumes de ataques futuros. No entanto, assim como em [34], é possível notar que estas redes nem sempre apresentarão uma previsão tão acertada como a apresentada na Figura 5.2. Tem-se um exemplo

desse comportamento na Figura 5.3, em que se observa intervalos de previsão com erros significativos, nesse caso, especificamente no intervalo de 50 a 125.

Figura 5.3 – Recorte de resultado da previsão para os ataques de força bruta



Fonte: elaborada pelo próprio autor.

Contudo, deve ser levando em conta que, diferente de outros métodos, os valores preditos não se deterioraram para valores futuros distantes. Deve-se ainda considerar que estes são dados reais de uma rede de grande porte, no caso um sistema autônomo e que os ataques possuem um comportamento pouco previsível.

5.4 Considerações sobre os resultados obtidos

Considerando os resultados obtidos nos dois submodelos apresentados, é notável a capacidade de inferência de ambos. As redes neurais foram capazes de extrair padrões complexos a partir das amostras de entrada e produzir bons resultados de previsão em cada um dos dois contextos considerados.

A comparação entre as várias arquiteturas de redes neurais consideradas no classificador de ataques contribuiu para corroborar as constantes afirmações presentes na literatura em relação a imprevisibilidade da melhor arquitetura na classificação de um conjunto de dados. Não era esperado que a rede de perceptrons teria um

desempenho elevado, assim como não se pensava que a unidade de LSTM teria os piores resultados. Dessa forma, os experimentos destacaram a necessidade de comparação de várias arquiteturas diferentes para a classificação de dados, não importando o domínio dos mesmos.

A rede recorrente utilizada na previsão de volume de cada uma das atividades maliciosas é um exemplo de como o comportamento futuro das séries temporais é uma tarefa difícil de prever. As relações e padrões internos de cada conjunto histórico possui comportamentos que dificilmente seriam inferidos até mesmo por um especialista. Destaca-se ainda que, conforme os dados da Tabela 5.3, não há relação aparente entre o número de amostras passadas e o resultado da previsão. A MSE resultante para a previsão dos volumes futuros das comunicações com mineradores foi três ordens de grandeza inferior que a dos ataques de força bruta, apesar destes apresentarem um número maior de amostras históricas do que os primeiros.

Após algumas combinações de hiperparâmetros, especialmente a quantidade de neurônios e unidades para cada arquitetura de rede neural, nota-se que não houve necessidade de acrescentar muitas camadas ocultas, de modo que quando as bases de dados precisarem ser geradas novamente, o processo de treinamento não exigirá um elevado tempo de espera.

CAPÍTULO 6 - Conclusões

6.1 Conclusões

O modelo de previsão de ataques para Sistemas Autônomos utilizando *Netflow* proposto neste trabalho tem significativa contribuição para a evolução do estado de segurança de redes de grande porte. A utilização de fluxos de rede como única fonte de informação é uma estratégia promissora, pois outras fontes de dados geram um volume de dados muito maior considerando o contexto de um sistema autônomo, o que demandaria um alto custo computacional para a análise de dados e treinamento do modelo.

O emprego da *Elastic Stack* como sistema agregador é indispensável para a exequibilidade da proposta, uma vez que sua robustez no tratamento dos dados, bem como a facilidade oferecida para obtenção e visualização de informações a partir deles, são pontos cruciais para que as detecções e projeções sejam realizadas e exibidas com eficiência.

As redes neurais artificiais empregadas neste trabalho foram essenciais para a obtenção de um modelo efetivamente adequado nas tarefas de inferência propostas. A capacidade que esses métodos de aprendizado de máquina possuem de extrair informações e relações entre os dados de entrada se confirmou mais uma vez por meio deste trabalho, considerando ainda a alta complexidade inerente a dados de tráfego em redes de computadores. Levando em consideração esses aspectos o modelo foi capaz de apresentar bons resultados para o contexto e os dados disponíveis nesse trabalho.

Em relação ao classificador proposto, considerou-se a comparação do resultado da melhor arquitetura dentre as consideradas ao se comparar com os demais trabalhos. Embora a rede baseada em LSTM do modelo proposto seja a única que não apresentou resultados compatíveis com [33] e [35], é importante destacar que a variante GRU, que é também recorrente obteve resultados melhores e é semelhante a rede LSTM em termos arquiteturais. Além disso, a proposta do classificador é selecionar a rede com melhor desempenho entre todas as consideradas.

Embora o foco deste trabalho seja a etapa de previsão, os resultados aqui obtidos também farão parte de um modelo de CSA que está sendo desenvolvido por uma pesquisadora do Laboratório ACME!. Espera-se que os resultados aqui obtidos possam servir para compor visualizações de dados preditivos num contexto mais abrangente, de modo que as previsões sejam realizadas de forma assertiva e não dispendiosa, que possam de fato ser utilizadas por um administrador de redes para tomar medidas de mitigação efetivas.

6.2 Dificuldades encontradas

Ao longo da elaboração deste projeto, diversas dificuldades foram encontradas. A principal delas foi conseguir encontrar a melhor configuração dos hiperparâmetros das redes neurais sem que o modelo consumisse uma quantidade de tempo elevada no processo de treinamento.

A implementação dos modelos também foi uma dificuldade a ser contornada, uma vez que a definição e das arquiteturas das redes neurais e ajustes das peculiaridades, como os hiperparâmetros, presentes no processo de treinamento das mesmas levou um tempo considerável.

6.3 Trabalhos futuros

Neste trabalho considerou-se a classificação de ataques e a previsão de volumes de ataques futuros como objetos de predição. Existe ainda a possibilidade de prever o

formato completo do ataque, de modo que todas as suas características sejam estimadas por uma rede neural, semelhante ao proposto por uma das seções de [36].

O sistema de CSA do qual este trabalho faz parte foi implementado predominantemente usando as ferramentas da *Elastic Stack*. Dessa forma, seria interessante incorporar o modelo proposto como uma das visualizações do *Kibana*, de modo que as previsões dos volumes de ataque possam ser visualizadas pelos administradores da rede na mesma interface onde as demais visualizações do sistema são consultadas.

Por fim, o ajuste dos hiperparâmetros pode ser testado usando técnicas de aprendizado de máquina automático. As ferramentas que implementam esta técnica são, em teoria, capazes de fornecer um bom ajuste de hiperparâmetros de forma automatizada, no entanto, o tempo despendido na busca desses ajustes pode ser ainda maior do que o ajuste manual realizado por um especialista.

REFERÊNCIAS

[1] PACITTI, T. **Do Fortran... à internet: construindo o futuro através da educação**. 3. ed. atu ed. São Paulo: Pioneira Thomson Learning, 2003.

[2] CENTRO DE ESTUDOS, RESPOSTA E TRATAMENTO DE INCIDENTES DE SEGURANÇA NO BRASIL – CERT.BR. **Estatísticas dos Incidentes Reportados ao CERT.br.** Disponível em: <https://www.cert.br/stats/incidentes/>. Acesso em: 20 ago. 2021.

[3] CABRAL, Carlos; CAPRINO, Willian. **Trilhas em Segurança da Informação: Caminhos e ideias para a proteção de dados**. Brasport, 2015.

[4] DAŞ, Resul; KARABADE, Abubakar; TUNA, Gurkan. **Common network attack types and defense mechanisms**. In: 2015 23nd Signal Processing and Communications Applications Conference (SIU). IEEE, 2015. p. 2658-2661.

[5] HUSÁK, Martin et al. **Survey of attack projection, prediction, and forecasting in cyber security**. IEEE Communications Surveys & Tutorials, v. 21, n. 1, p. 640-660, 2018.

[6] SAATY, Thomas Lorie; VARGAS, Luis Gonzalez. **Prediction, projection, and forecasting: applications of the analytic hierarchy process in economics, finance, politics, games, and sports**. Kluwer Academic Pub, 1991.

[7] LIU, Xiang; TANG, Ziyang; YANG, Baijian. **Predicting Network Attacks with CNN by Constructing Images from NetFlow Data**. In: 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS). IEEE, 2019. p. 61-66.

[8] CISCO. **NetFlow Configuration Guide, Cisco IOS Release 15M&T**. San jose. Disponível em: <https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/netflow/configuration/15-mt/nf-15-mt-book.pdf>. Acesso em: 25 ago. 2021.

[9] VELAN, Petr. **Improving network flow definition: formalization and applicability**. In: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2018. p. 1-5.

[10] HOFSTEDE, Rick et al. **Flow monitoring explained: From packet capture to data analysis with netflow and ipfix**. IEEE Communications Surveys & Tutorials, v. 16, n. 4, p. 2037-2064, 2014.

[11] JADIDI, Zahra et al. **A probabilistic sampling method for efficient flow-based analysis.** Journal of Communications and Networks, v. 18, n. 5, p. 818-825, 2016.

[12] XIE, Xiaosong; WU, Jincheng. **Real-Time Flow Identification Based on Neural Network and OpenFlow Over SDN**. In: 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC). IEEE, 2018. p. 12-16.

[13] CORDERO, Carlos García et al. **Analyzing flow-based anomaly intrusion detection using replicator neural networks**. In: 2016 14th Annual Conference on Privacy, Security and Trust (PST). IEEE, 2016. p. 317-324.

[14] JING, Xuyang; YAN, Zheng; PEDRYCZ, Witold. **Security data collection and data analytics in the Internet: A survey**. IEEE Communications Surveys & Tutorials, v. 21, n. 1, p. 586-618, 2018.

[15] OWASP. **Category:Attack - OWASP.** Disponível em: <https://www.owasp.org/index.php/Category:Attack>. Acesso em: 25 ago. 2021.

[16] CISCO. **Cyber Attack - What Are Common Cyberthreats? – Cisco.** Disponível em: <https://www.cisco.com/c/en/us/products/security/common-cyberattacks.html#~types-of-cyber-attacks>. Acesso em: 25 ago. 2021.

[17] IBM. **Cyber Attacks Explained | IBM.** Disponível em: <https://www.ibm.com/services/business-continuity/cyber-attack>. Acesso em: 25 ago. 2021.

[18] KETTANI, Houssain; WAINWRIGHT, Polly. **On the Top Threats to Cyber Systems**. In: 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT). IEEE, 2019. p. 175-179.

[19] FACHKHA, Claude; DEBBABI, Mourad. **Darknet as a source of cyber intelligence: Survey, taxonomy, and characterization**. IEEE Communications Surveys & Tutorials, v. 18, n. 2, p. 1197-1227, 2015.

[20] NAJAFABADI, Maryam M. et al. **Detection of ssh brute force attacks using aggregated netflow data**. In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). IEEE, 2015. p. 283-288.

[21] NAJAFABADI, Maryam M. et al. **Machine learning for detecting brute force attacks at the network level**. In: 2014 IEEE International Conference on Bioinformatics and Bioengineering. IEEE, 2014. p. 379-385.

[22] GHAFIR, Ibrahim; PRENOSIL, Vaclav. **Blacklist-based malicious ip traffic detection.** In: 2015 Global Conference on Communication Technologies (GCCT). IEEE, 2015. p. 229-233.

[23] APRUZZESE, Giovanni et al. **Identifying malicious hosts involved in periodic communications**. In: 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA). IEEE, 2017. p. 1-8.

[24] Elastic. **ELK Stack: Elasticsearch, Logstash e Kibana | Elastic**. Disponível em: < https://www.elastic.co/pt/what-is/elk-stack >. Acesso em: 29 ago. 2021

[25] Elastic. **O que é o Elasticsearch**. Disponível em: < https://www.elastic.co/pt/what-is/elasticsearch >. Acesso em: 29 ago. 2021

[26] Elastic. **O que é o Kibana**. Disponível em: < https://www.elastic.co/pt/what-is/kibana >. Acesso em: 29 ago. 2021

[27] Elastic. **Beats: Agente de dados para o Elasticsearch | Elastic**. Disponível em: < https://www.elastic.co/pt/what-is/elk-stack >. Acesso em: 29 ago. 2021

[28] FRANKE, U.; BRYNIELSSON, J. **Cyber situational awareness - A systematic review of the literature.** Computers and Security, [s. l.], v. 46, p. 18–31, 2014.

[29] MCCULLOCH, Warren S.; PITTS, Walter. **A logical calculus of the ideas immanent in nervous activity**. The bulletin of mathematical biophysics, v. 5, n. 4, p. 115-133, 1943.

[30] SKANSI, Sandro. **Introduction to Deep Learning: from logical calculus to artificial intelligence**. Springer, 2018.

[31] KELLEHER, J. D. **Deep Learning**. [s.l.] : The MIT Press, 2019.

[32] ZHANG, Aston et al. **Dive into deep learning**. Disponível em: < https://d2l.ai/ >. Acesso em: 30 ago. 2021

[33] SHEN, Y.; MARICONTI, E.; VERVIER, P. A.; STRINGHINI, G. **Tiresias: Predicting security events through deep learning**. In: PROCEEDINGS OF THE ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY 2018, New York, NY, USA. Anais... New York, NY, USA: Association for Computing Machinery, 2018. Disponível em: <https://dl.acm.org/doi/10.1145/3243734.3243811>. Acesso em: 30 ago. 2021.

[34] FANG, Xing et al. **A deep learning framework for predicting cyber attacks rates.** EURASIP Journal on Information security, v. 2019, n. 1, p. 1-11, 2019.

[35] BEN FREDJ, Ouissem et al. **CyberSecurity attack prediction: a deep learning approach**. In: 13th International Conference on Security of Information and Networks. 2020. p. 1-6.

[36] ANSARI, M. S.; BARTOS, V.; LEE, B. **Shallow and Deep Learning Approaches for Network Intrusion Alert Prediction**. Procedia Computer Science, [s. l.], v. 171, n. 2019, p. 644–653, 2020. Disponível em: <https://doi.org/10.1016/j.procs.2020.04.070>

[37] BARTOS, V.; ZADNIK, M.; HABIB, S. M.; VASILOMANOLAKIS, E. **Network entity characterization and attack prediction**. Future Generation Computer Systems, [s. l.], v. 97, p. 674–686, 2019.

[38] SHAO, Enchun. **Encoding IP Address as a Feature for Network Intrusion Detection**. 2019. Trabalho de Conclusão de Curso. Master's Thesis, Purdue University, West Lafayette, Indiana.

[39] HAN, W.; TIAN, Z.; HUANG, Z.; ZHONG, L.; JIA, Y. **System Architecture and Key Technologies of Network Security Situation Awareness System YHSAS**. CMC, [s. l.], v. 59, n. 1, p. 167–180, 2019. Disponível em: <www.techscience.com/cmc>. Acesso em: 30 ago. 2021.

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

TERMO DE REPRODUÇÃO XEROGRÁFICA

Autorizo a reprodução xerográfica do presente Trabalho de Conclusão, na íntegra ou em partes, para fins de pesquisa.

São José do Rio Preto, 28/09/2021

_____

Assinatura do autor