ESCOLA NAVAL

talant de bien faire

Frederico Nunes de Oliveira Correia Gonçalves

Cibersegurança – Ciberrange da Escola Naval

Bases de dados e classificador baseado em redes neuronais

Dissertação para obtenção do Grau de Mestre em Ciências Militares Navais, na especialidade de Engenharia Naval Ramo de Armas e Eletrónica

Alfeite

2023

Frederico Nunes de Oliveira Correia Gonçalves

Cibersegurança – Ciberrange da Escola Naval
Bases de dados e classificador baseado em redes neuronais

Dissertação para obtenção do Grau de Mestre em Ciências Militares Navais, na especialidade de Engenharia Naval Ramo de Armas e Eletrónica

Orientação de: Professor Victor José de Almeida e Sousa Lobo

Coorientação de: Professor Anacleto Cortez e Correia

Coorientação de: Professora Valéria Magalhães Pequeno

O Aluno Mestrando

Assinado por: **Frederico Nunes de Oliveira Correia Gonçalves**
Num. de Identificação: 15175651
Data: 2023.08.29 17:07:38+01'00'

CHAVE MÓVEL

[Frederico Nunes de Oliveira Correia Gonçalves]

O Orientador

Assinado por: **VICTOR JOSÉ DE ALMEIDA E SOUSA LOBO**
Num. de Identificação: 09302569
Data: 2023.08.29 18:46:36+01'00'

[Victor José de Almeida e Sousa Lobo]

Alfeite

2023

Epígrafe

"Research is formalized curiosity.

It is poking and prying with a purpose."

Zora Neale Hurston

Agradecimentos

Com o finalizar desta etapa, resta-me deixar alguns agradecimentos aos que me ajudaram e apoiaram a chegar até aqui.

Começo por agradecer ao meu orientador Professor Victor Lobo que se mostrou sempre disponível com a partilha generosa do seu vasto conhecimento e experiência.

Ao Professor Anacleto Cortez e Correia, que desempenhou um papel fundamental como coorientador nesta dissertação. Ressalvo a sua inestimável disponibilidade, incentivo, conselhos e paciência, qualidades fundamentais para a conclusão deste trabalho académico.

À Professora Valéria Pequeno que aceitou pela primeira vez o desafio de orientar a dissertação de um Aspirante da Marinha Portuguesa e que sempre se mostrou proativa e disponível no progresso do projeto.

Não podia deixar de agradecer aos meus pais e à minha irmã pelo carinho e apoio constantes ao longo desta grande caminhada.

Agradeço à minha namorada pelo seu apoio incondicional, compreensão, carinho e pela constante presença tanto bons como nos mais difíceis momentos.

Por fim, agradeço ao Curso Contra-Almirante Manuel Armando Ferraz, com uma especial dedicatória à classe de Engenheiros Navais – Ramo de Armas e Eletrónica. Camaradas que percorreram esta caminhada ao meu lado e assim iremos seguir “na eternidade do amanhã”.

Os meus sinceros agradecimentos a todos.

Resumo

O aumento crescente de ciberataques tem se tornado um problema cada vez mais grave num mundo cada vez mais digital e dependente de dispositivos conectados e vulneráveis. Os ciberataques estão a tornar-se cada vez mais sofisticados e frequentes, representando uma ameaça significativa para governos, empresas e indivíduos. A proteção contra estes ataques é crucial para a integridade dos sistemas e a privacidade das informações. A utilização de técnicas avançadas, como a aprendizagem de máquina, desempenha um papel fundamental na deteção e prevenção de ciberataques, permitindo uma resposta mais eficiente e proativa diante das ameaças digitais. Portanto, a pesquisa realizada nesta dissertação contribui para o avanço da cibersegurança, oferecendo soluções para a proteção contra ciberataques no contexto da Marinha Portuguesa.

A presente dissertação faz parte do projeto Ciberrange da Escola Naval, e que tem como foco o estudo de bases de dados e a construção de classificadores para deteção de ciberataques. Além disso, enquadra o presente trabalho na literatura existente nas temáticas de ataques e ameaças, técnicas de aprendizagem de máquina e métodos de avaliação da solução. Também é realizado um estudo das bases de dados *open-source* disponíveis na área da cibersegurança.

O trabalho é enquadrado no contexto da segurança organizacional, mais especificamente na Marinha Portuguesa. Destacam-se os pontos-chave fundamentais que um ciberrange deve ter, de acordo com os interesses de uma organização que visa proteger-se a si mesma e aos seus membros.

Por fim, a dissertação desenvolve, analisa e compara classificadores baseados em redes neurais com o objetivo de detetar ciberataques, utilizando como suporte a base de dados UNSW-NB15. Os resultados obtidos mostram uma precisão de 0.9301 na classificação binária, 0.8211 na classificação multi-classe e 0.8358 na classificação multi-classe com redução de ataques minoritários.

Palavras-Chave: Cibersegurança, Ciberrange, Aprendizagem máquina, Bases de dados, Redes neuronais, Sistemas de deteção de intrusões

Abstract

The increasing rise of cyberattacks has become an increasingly serious problem in an increasingly digital world that is dependent on connected and vulnerable devices. Cyberattacks are becoming more sophisticated and frequent, posing a significant threat to governments, companies, and individuals. Protection against these attacks is crucial for system integrity and information privacy. The use of advanced techniques such as machine learning plays a fundamental role in detecting and preventing cyberattacks, enabling a more efficient and proactive response to digital threats. Therefore, the research conducted in this dissertation contributes to the advancement of cybersecurity by offering solutions for protection against cyberattacks in the context of the Portuguese Navy.

This dissertation is part of the “Ciberrange da Escola Naval” and focuses on database analysis and classifier construction for cyberattack detection. It also contextualizes this work within the existing literature on attack and threat analysis, machine learning techniques, and solution evaluation methods. Additionally, an examination of open-source databases available in the field of cybersecurity is conducted.

The work is framed within the context of organizational security, specifically the Portuguese Navy. Key fundamental points that a cyber range should have are highlighted, according to the interests of an organization seeking to protect itself and its members.

Finally, the dissertation develops, analyzes, and compares neural network-based classifiers with the objective of detecting cyberattacks, using the UNSW-NB15 database as support. The results obtained show an accuracy of 0.9301 in binary classification, 0.8211 in multi-class classification, and 0.8358 in multi-class classification with the reduction of minority attacks.

Keywords: Cybersecurity, Cyber Range, Machine Learning, Databases, Neural Networks, Intrusion Detection Systems

Índice

Epígrafe III

Agradecimentos V

Resumo VII

Abstract IX

Índice XI

Índice de Figuras XIII

Índice de Tabelas XV

Lista de abreviaturas e acrónimos XVII

Introdução 3

1. Enquadramento 3

2. Objeto da dissertação 4

3. Investigação 4

3.1. Metodologia científica 5

4. Estrutura da dissertação 7

Capítulo 1: Revisão da literatura 11

1.1. Ameaças e ataques 12

1.2. Técnicas de aprendizagem máquina 14

1.2.1. Técnicas de aprendizagem profunda 16

1.3. Bases de dados 19

1.4. Métodos de avaliação da solução 26

Capítulo 2: Cibersegurança no contexto das organizações 33

2.1. Importância 33

2.2. Marinha portuguesa 34

Capítulo 3: Ciberrange 39

Capítulo 4: Classificador baseado em redes neuronais 45

4.1. Descrição das ferramentas 46
4.2. Processamento dos dados 46
4.2.1. Classificação multi-classe 46
4.2.2. Classificação binária 50
4.3. Descrição dos modelos 51
4.3.1. FFNN 51
4.3.2. DNN 52
4.3.3. LSTM 53
4.3.4. CNN-LSTM 54
4.4. Treino dos modelos 56
4.5. Métodos de avaliação da solução 56
Capítulo 5: Validação dos resultados 61
5.1. Classificação multi-classe reduzida 62
5.2. Classificação multi-classe – ataques originais 67
5.3. Classificação binária 72
Conclusões 81
1. Trabalho futuro 84
Referências bibliográficas 85
ANEXOS 95
1. Anexo A – Tipos de ciberataques 95
2. Anexo B – Métodos de avaliação da solução 102
3. Anexo C – Soluções de aprendizagem máquina 105
4. Anexo D – Técnicas de aprendizagem máquina 107
5.4. Anexo D.1 – Técnicas de aprendizagem profunda 112

Índice de Figuras

Figura 1: Paralelismo da metodologia adotada com os capítulos da presente dissertação 7

Figura 2: Esquema representativo das funcionalidades do ciberrange no contexto da presente dissertação 34

Figura 3: Distribuição de classes no conjunto de treino multi-classe com redução de ataques 48

Figura 4: Distribuição de classes no conjunto de teste multi-classe com redução de ataques. 48

Figura 5: Distribuição das correlações *pearson* entre as diversas colunas para a classificação multi-classe com redução dos ataques 49

Figura 6: Esquema representativo da arquitetura FFNN adotada 52

Figura 7: Esquema representativo da arquitetura RNN adotada 54

Figura 8: Esquema representativo da arquitetura CNN-LSTM adotada 55

Figura 9: *Loss* e exatidão durante o treino dos classificadores multi-classe com ataques reduzidos 64

Figura 10: Matrizes de confusão das diferentes redes neuronais de classificação multi-classe com os ataques reduzidos 65

Figura 11: *Loss* e exatidão durante o treino dos classificadores multi-classe com ataques originais 69

Figura 12: Matrizes de confusão das diferentes redes neuronais de classificação multi-classe com os ataques originais 70

Figura 13: *Loss* e exatidão durante o treino dos classificadores binários 74

Figura 14: Matrizes de confusão das diferentes redes neuronais de classificação binária 75

Índice de Tabelas

Tabela 1: Diretrizes do *Design-Science Research* 6
Tabela 2: Tipos de ataques avaliados com aprendizagem máquina 12
Tabela 3: Técnicas de aprendizagem máquina usadas na deteção de ataques 15
Tabela 4: Técnicas de aprendizagem profunda nos documentos analisados 17
Tabela 5: Bases de dados baseadas em tráfego de rede nos artigos analisados 21
Tabela 6: Bases de dados baseadas em tráfego de internet nos artigos analisados 22
Tabela 7: Bases de dados baseadas em tráfego móvel nos artigos analisados 23
Tabela 8: Comparação da presença de bases de dados nos artigos analisados 24
Tabela 9: Comparação da abordagem da matriz de confusão nos artigos analisados 27
Tabela 10: Ataques presentes na base de dados UNSW-NB15 46
Tabela 11: Colunas da base de dados UNSW-NB15 adotadas 47
Tabela 12: Métricas na FFNN de classificação multi-classe reduzida 65
Tabela 13: Métricas na DNN de classificação multi-classe reduzida 65
Tabela 14: Métricas na RNN de classificação multi-classe reduzida 66
Tabela 15: Métricas na CNN-LSTM de classificação multi-classe reduzida 66
Tabela 16: Métricas na FFNN de classificação multi-classe – ataques originais 70
Tabela 17: Métricas na DNN de classificação multi-classe – ataques originais 71
Tabela 18: Métricas na RNN de classificação multi-classe – ataques originais 71
Tabela 19: Métricas na CNN-LSTM de classificação multi-classe – ataques originais 71
Tabela 20: Métricas na FFNN de classificação binária 75
Tabela 21: Métricas na DNN de classificação binária 75
Tabela 22: Métricas na RNN de classificação binária 75
Tabela 23: Métricas na CNN-LSTM de classificação binária 76
Tabela 24: Comparação dos resultados da literatura na classificação multi-classe 77
Tabela 25: Comparação dos resultados da literatura na classificação binária 77

Lista de abreviaturas e acrónimos

AUC	*Area Under Curve*
CNN	*Convolutional Neural Network*
DDoS	*Distributed Denial of Service*
DNN	*Deep Neural Network*
DoS	*Denial of Service*
DSR	*Design Science Research*
FDR	*False Discovery Rate*
FFNN	*Feed-Forward Neural Network*
FN	*False Negative*
FNR	*False Negative Rate*
FOR	*False Omission Rate*
FP	*False Positive*
FPR	*False Positive Rate*
IoT	*Internet of Things*
LSTM	*Long Short-Term Memory*
MiTM	*Mas-in-The-Middle*
NPV	*Negative Predictive Value*
RNN	*Recorrent Neural Network*
ROC	*Receiver Operating Characteristic*
SI	Sistemas de Informação
TN	*True Negative*
TNR	*True Negative Rate*
TP	*True Positive*
TPR	*True Positive Rate*

Introdução

1. Enquadramento
2. Objeto da dissertação
3. Investigação
4. Estrutura da dissertação

Introdução

1. Enquadramento

A cibersegurança é um tema de crescente preocupação no mundo de hoje. A ascensão da tecnologia e da Internet trouxe oportunidades e benefícios sem precedentes, mas também criou ameaças e desafios. Os ciberataques têm se tornado cada vez mais comuns e podem ter consequências devastadoras para indivíduos, organizações e a economia como um todo.

A importância da cibersegurança tornou-se especialmente evidente nos últimos anos, à medida que os ciberataques se tornaram mais frequentes e mais sofisticados. *Hackers* e cibercriminosos desenvolvem constantemente novas táticas e técnicas para violar sistemas de segurança e roubar dados confidenciais. As consequências dos ciberataques podem ser graves, desde perdas financeiras a danos reputacionais e até danos físicos.

O contexto em que os ciberataques ocorrem também está a mudar. À medida que cada vez mais pessoas dependem das tecnologias e serviços digitais, aumenta o potencial de ciberataques causarem danos. Além disso, a utilização crescente de dispositivos ligados à Internet e da Internet das Coisas (IoT, Internet of Things) cria vulnerabilidades que podem ser exploradas por piratas informáticos e cibercriminosos.

O impacto dos ciberataques pode ser sentido em toda a economia, afetando diariamente empresas de todas as dimensões e setores. Para além disso, os ciberataques podem resultar no roubo de propriedade intelectual valiosa, dados de clientes e informações financeiras. Isto pode levar a perdas financeiras significativas, bem como danos à reputação de uma empresa e à confiança do cliente.

Para organizações e empresas, as consequências dos ciberataques podem ser particularmente graves. Um ciberataque bem-sucedido pode comprometer dados sensíveis, interromper operações e danificar infraestruturas críticas. Os custos de recuperação de um ciberataque podem ser significativos, tanto em termos de recursos financeiros como de tempo e esforço necessários para restaurar sistemas e operações.

Dada a importância crescente da cibersegurança, é essencial desenvolver estratégias e instrumentos eficazes de proteção contra ciberataques. Instrumentos estes que permitam fornecer informações valiosas sobre os padrões e comportamentos dos cibercriminosos, permitindo que organizações e empresas antecipem e respondam a ameaças potenciais de

forma mais eficaz. O projeto Ciberrange da Escola Naval é um exemplo promissor de como as tecnologias podem ser usadas para melhorar a cibersegurança e proteger contra ciberataques num mundo cada vez mais digital.

2. Objeto da dissertação

A crescente ameaça de ciberataques deixou claro que as organizações devem estar preparadas para se defender de potenciais ameaças. Para o fazer de forma eficaz, as organizações precisam de ter entre todos os seus elementos um conhecimento profundo dos tipos de ataques que podem enfrentar, bem como as habilidades, ferramentas e técnicas necessárias para se defender contra eles. Estas necessidades precisam de vir a ser colmatadas com estratégias eficazes de cibersegurança para as organizações, existindo assim a necessidade de investir na formação e em infraestruturas de defesa contra potenciais ciberataques.

O objetivo da presente dissertação é investigar estratégias que consigam colmatar a necessidade de cibersegurança de uma organização através de uma infraestrutura que permita a formação dos vários elementos pertencentes a essa organização. Em particular pretende-se usar bases de dados e classificadores baseados em redes neuronais para detetar padrões e comportamentos de cibercriminosos de forma a antecipar e responder a ameaças potenciais de forma mais eficaz.

3. Investigação

Para melhorar a preparação para a cibersegurança organizacional, deve ser construída uma infraestrutura eficaz que permita aos indivíduos pertencentes à organização aprender sobre ciberataques e guarnecer estes mesmos indivíduos com as competências necessárias para defender a organização.

Esta infraestrutura deve incluir oportunidades de aprendizagem colaborativa para promover a sensibilização, a preparação e uma cultura de segurança. A infraestrutura deve ainda ser utilizada para recolher dados de forma a alimentar um modelo classificador baseado em redes neurais, com a capacidade de detetar e analisar padrões e comportamentos de cibercriminosos.

A presente dissertação irá focar-se na construção de um classificador baseado em redes neuronais, com a capacidade de detetar e analisar padrões e comportamentos de

cibercriminosos, bem como a análise das bases de dados existentes mais relevantes no âmbito do projeto Ciberrange da Escola Naval.

De forma a desenvolver o classificador baseado em redes neuronais no âmbito do Ciberrange da Escola Naval, algumas questões devem ser respondidas, nomeadamente:

- Quais são as práticas utilizadas em classificadores semelhantes construídos em projetos anteriores?
- Como identificar a melhor infraestrutura para a recolha de dados e ainda capaz de promover a preparação e proteção de uma organização relativamente à cibersegurança?
- Quais são os elementos-chave de uma infraestrutura que recolhe dados e apoia a aprendizagem colaborativa de preparação para a cibersegurança?
- Como pode ser construído o classificador baseado em redes neuronais a partir de uma base de dados?
- O que indicam, que limitações refletem e como podem ser comparados a projetos anteriores os resultados obtidos pelo classificador?

3.1. **Metodologia científica**

A metodologia científica adotada na presente dissertação é uma adaptação da metodologia *Design Science Research* (DSR) (Hevner et al., 2004). DSR é uma metodologia de investigação utilizada na área dos Sistemas de Informação (SI) que visa criar artefactos e conhecimentos inovadores que abordem problemas práticos. A DSR procura projetar, desenvolver e avaliar sistemas de informação ou soluções para problemas, combinando conhecimento científico com experiência prática num *design*. Um documento DSR normalmente segue um formato estruturado que descreve o problema, a solução proposta e a avaliação da solução, seguindo sete diretrizes propostas por (Hevner et al., 2004) como demonstra Tabela 1.

A presente dissertação, no Capítulo 4: Classificador baseado em redes neuronais, descreve a construção do artefacto na forma de um modelo (Diretriz 1), oferecendo uma solução para o problema apresentado anteriormente na Introdução - detetar padrões e comportamentos de cibercriminosos de forma a antecipar e responder a ameaças potenciais de forma mais eficaz (Diretriz 2). O artefacto é avaliado utilizando-se diversos métodos, conforme descrito no Capítulo 5: Validação dos resultados (Diretriz 3).

Ao longo desta dissertação, especialmente na Introdução e no Capítulo 2: Cibersegurança

no contexto das organizações, são evidenciadas as contribuições e a relevância que o presente trabalho possui no âmbito da aprendizagem máquina aplicada à cibersegurança (Diretriz 4).

A construção e avaliação do artefacto baseiam-se na aplicação dos métodos apresentados no Capítulo 1: Revisão da Literatura, os quais são discutidos de forma detalhada nos Capítulos 4 e 5 (Diretriz 5).

A busca pelo aperfeiçoamento do artefacto envolveu a utilização de diversos métodos com o objetivo de alcançar uma solução que seja aplicável no contexto real do problema (Diretriz 6).

A presente dissertação apresenta a solução de maneira eficaz ao longo dos diferentes capítulos, sendo acessível tanto para o público com conhecimentos técnicos na área de tecnologia, quanto para o público voltado para a gestão organizacional (Diretriz 7).

DIRETRIZES	**DESCRIÇÃO**
Diretriz 1: *Design* como um artefacto	DSR deve produzir um artefacto viável na forma de uma construção, modelo, método ou instanciação.
Diretriz 2: Relevância do problema	O objetivo da DSR é desenvolver soluções baseadas em tecnologia para problemas importantes e relevantes.
Diretriz 3: Avaliação do *design*	A utilidade, qualidade e eficácia de um artefacto devem ser rigorosamente demonstradas por meio de métodos de avaliação bem executados.
Diretriz 4: Contribuições da pesquisa	A DSR eficaz deve fornecer contribuições claras e verificáveis nas áreas do artefacto, fundações de *design* e/ou metodologias de *design*.
Diretriz 5: Rigor de pesquisa	A DSR depende da aplicação de métodos rigorosos tanto na construção quanto na avaliação do artefacto de *design*.
Diretriz 6: *Design* como um processo de busca	A pesquisa por um artefacto eficaz requer a utilização dos meios disponíveis para alcançar os fins desejados enquanto satisfaz as leis no ambiente do problema.
Diretriz 7: Comunicação da pesquisa	A DSR deve ser apresentada de maneira eficaz tanto para o público orientado para a tecnologia quanto para o público orientado para a gestão.

Tabela 1: Diretrizes do *Design-Science Research*

Note-se que a DSR é uma metodologia que assenta no processo iterativo onde tanto os capítulos que contêm a apresentação das alternativas como o *design* em si são revisitados e "alimentados" com o que se avaliou em capítulos posteriores (Hevner et al., 2004; Takeda et al., 1990), questão que será evidenciada na Figura 1 que ilustra o paralelismo entre os capítulos da presente dissertação e a modelagem de processos de *design* proposta por (Takeda et al., 1990).

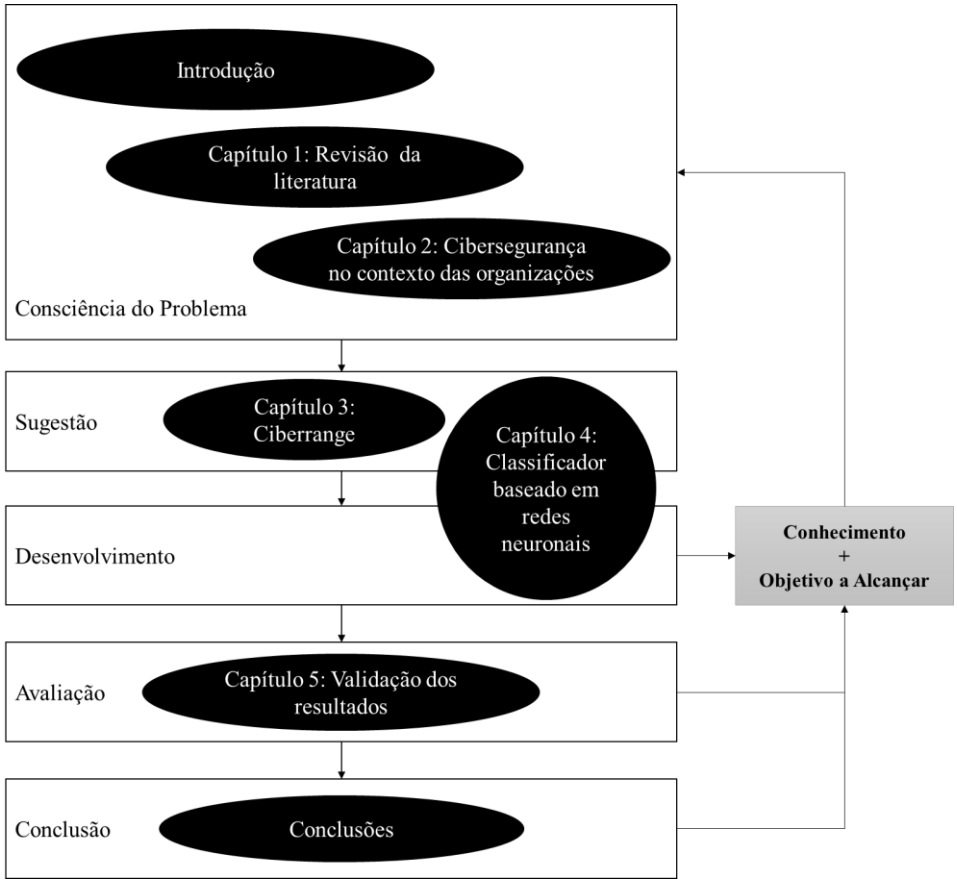

Figura 1: Paralelismo da metodologia adotada com os capítulos da presente dissertação

4. Estrutura da dissertação

Nesta secção, serão enumerados os elementos essenciais que compõem a estrutura da presente dissertação.

- Introdução - A dissertação começará pela introdução onde será abordada a relevância do tema, o contexto dos ciberataques, que prejudicam a economia, empresas e organizações, discutida a motivação, a razão da dissertação ser realizada e levantada a questão principal da dissertação, as suas subquestões associadas, bem como um paralelismo com a metodologia científica utilizada.
- Capítulo 1: Revisão da Literatura - A revisão da literatura fará uma abordagem a documentos utilizados como fonte para a realização da presente dissertação expondo o nível de estudo profundo/grande formalismo nos temas que darão lugar às secções Ameaças e ataques, Técnicas de aprendizagem máquina, Bases de dados e Métodos de avaliação da solução.
- Capítulo 2: Cibersegurança no contexto das organizações - Este capítulo abordará a necessidade de resolver o problema apresentado no contexto das organizações,

mais especificamente na Marinha Portuguesa.

- Capítulo 3: Ciberrange - O capítulo contemplará o levantamento dos requisitos que deve ter a infraestrutura.
- Capítulo 4: Classificador baseado em redes neuronais – O capítulo em questão irá apresentar o classificador baseado em redes neuronais como solução para detetar padrões e comportamentos de cibercriminosos a partir de uma base de dados.
- Capítulo 5: Validação dos resultados - Neste capítulo irão ser apresentados os resultados obtidos pela solução apresentada para o problema inicial.
- Conclusões: As conclusões resumirão os resultados da pesquisa respondendo às subquestões que foram apresentadas na introdução, fornecendo ainda recomendações para pesquisas futuras.

Cap. 1: Revisão da Literatura

1.1. Ameaças e ataques

1.2. Técnicas de aprendizagem máquina

1.3. Bases de Dados

1.4. Métodos de avaliação da solução

Capítulo 1: Revisão da literatura

O presente capítulo tem como propósito rever e analisar criticamente a literatura existente, bem como a investigação relevante para o tema de estudo.

O foco será nos temas relacionados com ameaças e ataques, técnicas de aprendizagem máquina, bases de dados e métodos de avaliação de soluções de aprendizagem máquina. Estes temas são cruciais no campo da cibersegurança e aprendizagem máquina. e o seu estudo detalhado e análise formal são necessários para o desenvolvimento de soluções eficazes neste domínio.

No presente capítulo preferiu-se predominar a presença de artigos relevantes face a cada tema ao invés de uma explicação detalhada de cada tema, visto que esta explicação já se encontra disponível na literatura apresentada. Assim sendo, ao longo do capítulo, cada tema irá ser acompanhado por uma abordagem a documentos utilizados como fonte para a realização da presente dissertação expondo a presença de cada tema em cada documento analisado. A seleção dos documentos analisados teve em vista escolher artigos bastante citados na literatura, de diferentes partes do mundo, diferentes organizações e plataformas de forma a alargar o leque da literatura analisada.

A secção Ameaças e Ataques explorará os principais tipos de ameaças e ciberataques que as organizações enfrentam atualmente.

A secção Técnicas de Aprendizagem Máquina fornecerá uma discussão dos vários métodos de aprendizagem máquina usados para cibersegurança.

A secção Bases de Dados discutirá os vários repositórios de dados usados para treino de modelos de aprendizagem máquina no âmbito da cibersegurança.

Por fim, a secção Métodos de Avaliação da Solução irá analisar os vários métodos e técnicas utilizados para avaliar a eficácia das soluções de cibersegurança. Serão discutidas as métricas utilizadas para medir o desempenho dessas soluções nos diferentes documentos analisados.

No geral, a revisão da literatura realizada tem como objetivo fornecer uma visão abrangente dos principais temas relacionados à cibersegurança e aprendizagem máquina. O estudo aprofundado e a análise formal destes temas servirão de base sólida para o desenvolvimento dos capítulos seguintes.

1.1. Ameaças e ataques

A ameaça e o ataque são conceitos relacionados, no entanto distintos. Uma ameaça caracteriza-se por uma possível ação ou evento que possa comprometer a segurança ou integridade de sistemas computacionais, redes ou dados. A ameaça pode ser intencional ou não, representando um risco ou perigo potencial. (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020; Vazhakkat, 2022)

Por outro lado, um ataque corresponde a uma ação ou incidente mal-intencionado que causa dano a sistemas computacionais redes ou dados. O ataque pode ser passivo ou ativo.

As ameaças acabam por ser precursoras dos ataques, pois representam as vulnerabilidades por onde um invasor pode explorar.

Artigo	(Wazid et al., 2022)	(Li et al., 2021)	(Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)	(Berman et al., 2019)	(Bahassi et al., 2022)	(Dasgupta et al., 2022)	**Presente Dissertação**
Malware	✓	✓	✓	✓	✓	✓	✓
DoS	✓	✓	✓	✓	✓	✓	✓
MITM	✓	✗	✗	✗	✗	✓	✓
***Eavesdropping* (*Sniffing*/*Snooping*)**	✓	✗	✗	✗	✓	✓	✓
Ataque de acesso privilegiado	✓	✗	✗	✗	✗	✓	✓
Ataques dia-zero	✓	✗	✓	✓	✗	✓	✓

Tabela 2: Tipos de ataques avaliados com aprendizagem máquina

De acordo com os artigos que foram tomados como base de pesquisa para a realização deste capítulo da presente dissertação apresenta-se a Tabela 2 com comparação da presença do tema nos vários artigos. Ressalva-se que existem outras maneiras possíveis de englobar os tipos de ciberataques.

Nos artigos analisados o *malware* destaca-se em (Li et al., 2021) e em (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020). (Li et al., 2021) apresenta estudo bastante aprofundado, analisando vários métodos de deteção de *malware* com base na aprendizagem profunda e comparando-os com os métodos tradicionais de aprendizagem máquina. Já (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020) é um artigo focado apenas em deteção de intrusão, deteção de *malware* e deteção de *spam*, sendo assim bastante completo relativamente ao tópico em questão. Compara a existência deste tema em documentos anteriores e ainda compara o que existe sobre *malware* em computadores e *smart devices* em vários aspetos como ferramentas, tendências, técnicas e métodos.

Em (Berman et al., 2019) e em (Akhtar & Feng, 2022) o estudo não se revela tão aprofundado como nos documentos destacados no parágrafo anterior. Ainda assim (Berman et al., 2019) refere como a deteção e classificação de *malware* é realizada comparando a forma como é feita em diferentes artigos anteriores, apresentando detalhes sobre como foi realizada nos artigos analisados. (Akhtar & Feng, 2022) é um artigo em torno de um modelo específico para detetar intrusões *malware*. Explica em que consiste *malware* e estuda aprofundadamente comparando o seu modelo em diferentes variantes.

Relativamente a (Wazid et al., 2022), explica em que consiste o ataque, mas é escasso na comparação de modelos de aprendizagem máquina para deteção de *malware*.

Denial of Service (DoS) destaca-se em (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020) onde para além de ser dada a explicação do ataque são mencionadas técnicas de aprendizagem máquina usadas para detetar este ataque em artigos anteriores (árvores de decisão, redes neuronais, naive bayes e SVM), com as percentagens de precisão associadas.

Em (Li et al., 2021) são mencionados ao longo do documento alguns modelos já utilizados para detetar este tipo de ataque, no entanto não se apresenta tão bem estruturado como em (Shaukat et al., 2020).

Relativamente a (Wazid et al., 2022), é explicado em que consiste o ataque, no entanto refere apenas um modelo usado para o detetar.

Apresentando consideravelmente menos informação relativamente aos outros documentos analisados tem-se (Berman et al., 2019) que refere muito pouco o tipo DoS, referindo apenas um modelo capaz de detetar vários ataques incluindo DoS e (Akhtar & Feng, 2022) que apresenta um modelo específico de deteção que não cobre DoS.

(Wazid et al., 2022) é o único dos documentos que apresenta informação relativa a 1.1. *Mas-in-The-Middle* (MiTM), Eavesdropping (Sniffing ou Snooping) e Ataque de acesso privilegiado onde é apenas apresentada uma explicação em que consiste o ataque.

Os ataques dia-zero em (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020) são referidos principalmente como um desafio futuro. São indicados como alvo de pesquisa para deteção de intrusão em computadores e dadas referências de modelos capazes de detetar estes ataques em dispositivos moveis.

Em (Berman et al., 2019) e (Wazid et al., 2022) são apenas mencionados, não existindo referência a estes ataques nos restantes documentos.

(Bahassi et al., 2022) e (Dasgupta et al., 2022) são artigos com um capítulo destinado a destacar os ataques mais utilizados. No caso de (Dasgupta et al., 2022) é dada uma explicação com elevado formalismo dos ataques mencionados. Já em (Bahassi et al., 2022) a explicação dos ataques mencionados não é tão aprofundada e alguns dos ataques considerados no nosso artigo não são mencionados ou apenas indicados.

1.2. Técnicas de aprendizagem máquina

As técnicas de aprendizagem máquina têm uma vasta gama de aplicações, desde visão computacional e processamento de linguagem natural, até cuidados de saúde e finanças. A escolha da técnica depende da tarefa específica e do tipo de dados que estão a ser analisados.

Neste capítulo irão ser expostas algumas soluções de aprendizagem máquina relevantes no que toca a aplicar estas técnicas na cibersegurança.

De acordo com os artigos que foram tomados como base de pesquisa das técnicas de aprendizagem máquina da presente dissertação apresenta-se a Tabela 3 com a comparação do tema abrangido pelos vários artigos.

Artigo	(Li et al., 2021)	(Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)	(Bahassi et al., 2022)	(Dasgupta et al., 2022)	(Shaukat, Luo, Varadharajan, Hameed, Chen, et al., 2020)	**Presente Dissertação**
Support vector machine	✗	✓	✓	✓	✓	✓
Árvore de decisão	✓	✓	✓	✓	✓	✓
K-vizinhos	✓	✓	✓	✓	✗	✓
Floresta aleatória	✓	✓	✗	✓	✓	✓
Clustering	✓	✓	✓	✓	✓	✓
Naive Bayes	✓	✓	✗	✓	✓	✓

Tabela 3: Técnicas de aprendizagem máquina usadas na deteção de ataques

Nos artigos analisados destaca-se o (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020) que inclui de forma esclarecedora vários tipos de aprendizagem máquina. Em (Li et al., 2021) são mencionados alguns métodos ao longo dos artigos

O artigo (Bahassi et al., 2022) possuí um capítulo destinado a destacar técnicas de aprendizagem máquina mais utilizadas. É dada uma explicação das técnicas utilizadas bem como ataques a que os diferentes modelos se adequam.

(Dasgupta et al., 2022) é um artigo com elevado formalismo tanto na explicação de cada técnica como na apresentação de referências de modelos que utilizam as diferentes técnicas, no entanto há que ressalvar que há técnicas que são menos referidas.

(Shaukat, Luo, Varadharajan, Hameed, Chen, et al., 2020) fornece uma breve explicação de cada técnica e ainda várias referências de artigos acompanhadas com informações sobre as precisões obtidas. Fornece também tabelas bastante completas que ilustram a utilização das diferentes técnicas na cibersegurança.

1.2.1. **Técnicas de aprendizagem profunda**

A aprendizagem profunda é um ramo da aprendizagem máquina que envolve o treino de redes neurais artificiais para aprender com dados e dar previsões ou decisões com base nesses dados. Os modelos de aprendizagem profunda são projetados para aprender e melhorar automaticamente a partir da experiência sem instruções explícitas de um programador.

Os algoritmos de aprendizagem profunda usam várias camadas de nós ou neurónios interconectados para aprender representações cada vez mais complexas dos dados de entrada. A rede neuronal é tipicamente treinada num grande conjunto de dados de exemplos, com o objetivo de maximizar as previsões corretas.

A aprendizagem profunda alcançou um desempenho de última geração em muitas tarefas, mas requer grandes quantidades de dados e recursos computacionais para treino. Além disso, os modelos de aprendizagem profunda podem ser difíceis de interpretar e, por vezes, podem cometer erros ou desvios inesperados.

A aprendizagem profunda mostra um grande potencial na deteção de ciberataques devido ao facto de poder aprender padrões complexos de dados, como tráfego de rede ou *logs* do sistema.

De acordo com os artigos que foram tomados como base de pesquisa das técnicas de aprendizagem profunda da presente dissertação apresenta-se a Tabela 4 com a comparação do tema abrangido pelos vários artigos.

Artigo	(Akhtar & Feng, 2022)	(Li et al., 2021)	(Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)	(Berman et al., 2019)	(Gümüsbas et al., 2020)	(Ferrag, Maglaras, Janicke, et al., 2019)	(Ferrag, Maglaras, Moschoyiannis, et al., 2019)	(Dasgupta et al., 2022)	(Shaukat, Luo, Varadharajan, Hameed, Chen, et al., 2020)	**Presente Dissertação**
Rede neuronal feed forward	✗	✓	✓	✗	✗	✓	✓	✓	✗	✓
Rede neuronal recorrente	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Rede neuronal profunda	✗	✓	✓	✓	✗	✓	✓	✓	✗	✓
Rede neuronal convolucional	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Memória longa de curto prazo (Long short-term memory)	✓	✓	✗	✓	✓	✓	✓	✓	✗	✓
Rede Deep belief	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Autocodificadores	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Máquina de Boltzmann Restrita	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Tabela 4: Técnicas de aprendizagem profunda nos documentos analisados

(Akhtar & Feng, 2022) foca-se especificamente no seu modelo CNN-LSTM, referindo esporadicamente outros métodos.

Nos artigos (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020) e (Li et al., 2021) há uma larga abrangência dos métodos de aprendizagem profunda. É feita uma breve explicação das técnicas acompanhadas por esquemas, sendo também falado em alguns modelos que fizeram uso das técnicas.

(Berman et al., 2019) apresenta os métodos são mais aprofundados que os artigos anteriores. Realiza a explicação do funcionamento das técnicas acompanhadas por esquemas, e ainda são dados vários artigos com as diferentes implementações.

(Gümüsbas et al., 2020) realiza uma revisão do que tem sido o uso de algumas das técnicas ao longo dos tempos. São também mencionadas outras técnicas ao longo do artigo, no entanto com um grau de formalismo menor.

(Ferrag, Maglaras, Janicke, et al., 2019) e (Ferrag, Maglaras, Moschoyiannis, et al., 2019) apresentam abordagens muito idênticas das técnicas de aprendizagem profunda. Realizam uma revisão onde incluem uma explicação dos resultados obtidos com modelos de outros artigos que utilizam as diversas técnicas. Possuem também uma secção dedicada à explicação acompanhado por esquemas e exemplos do algoritmo base de algumas das técnicas

Efetuam ainda um estudo onde comparam separadamente a performance de RNN, CNN, DNN e DBN, RBM, DBM, DA contra vários tipos de ataques.

(Shaukat, Luo, Varadharajan, Hameed, Chen, et al., 2020) apresenta uma breve explicação e refere artigos que utilizam as técnicas por si apresentadas. A destacar as DBN, que apresentam um número considerável de artigos comparativamente às outras técnicas de aprendizagem profunda apresentadas em (Shaukat, Luo, Varadharajan, Hameed, Chen, et al., 2020).

Após a análise dos diversos artigos irão ser mencionadas no âmbito da deteção de intrusão e cibersegurança algumas das técnicas de aprendizagem máquina que constituem uma maior relevância no que toca ao que se procura realizar na presente dissertação. As técnicas de aprendizagem máquina que se seguem têm desempenhado um papel significativo na construção de classificadores para deteção de intrusões. O foco será nas técnicas de aprendizagem profunda visto que nos diversos artigos analisados, revelaram obter melhores resultados no âmbito da construção de classificadores para deteção de intrusões.

I. Rede Neural Feed Forward (FFNN)

No âmbito da deteção de intrusão e cibersegurança a FFNN destaca-se o modelo de (Kasongo & Sun, 2019) que com uma taxa de aprendizagem de 0,05 e 30 neurónios em 3 camadas ocultas, a avaliação de desempenho mostrou que o sistema proposto atinge uma precisão de 99,69% (Ferrag, Maglaras, Moschoyiannis, et al., 2019).

II. Rede Neural Profunda (DNN)

No âmbito da deteção de intrusão e cibersegurança destaca-se o artigo bastante citado de (Tang et al., 2016) que treina os dados com diferentes taxas de aprendizagem e concluí que uma taxa de aprendizagem de 0,001 teve um desempenho mais eficaz do que outros com a maior curva característica de operação do recetor (AUC). (Kim et al., 2017) utilizou quatro camadas ocultas com 100 neurónios/nós usando a função ReLU como função de ativação e o método de otimização estocástico para treino do modelo que alcança uma precisão de aproximadamente 99%. (Ferrag, Maglaras, Moschoyiannis, et al., 2019).

III. **Rede Neural Convolucional (CNN)**

No âmbito da deteção de intrusão e cibersegurança destaca-se o artigo bastante citado (Fu et al., 2016) que usa uma rede neural convolucional para aprender os padrões intrínsecos de comportamentos fraudulentos, especialmente para deteção de fraudes com cartões de crédito (Ferrag, Maglaras, Moschoyiannis, et al., 2019). Há a referir também o artigo mais recente (Akhtar & Feng, 2022) que cria um modelo CNN-LSTM com uma precisão de deteção de 99% depois de restrigir o seu dataset às características mais relevantes com a justificação de reduzir o overfitting.

IV. **Rede Neuronal Recorrente (RNN)**

No âmbito da deteção de intrusão e cibersegurança destaca-se o artigo bastante citado de (C. Yin et al., 2017) que utilizou uma RNN com três indicadores de desempenho, incluindo precisão, taxa de falsos positivos e taxa de verdadeiros positivos. Referem também que o desempenho de deteção de anomalias tem maior precisão com taxa de aprendizagem = 0,1 e 80 nós nas camadas ocultas. Há também a referir um sistema de deteção de intrusão multicanal que usa *Long short-term memory* (LSTM) descritas por (Jiang et al., 2020). O modelo é relatado como 99,23% taxa de deteção com um falso alarme taxa de 9,86% e uma precisão de 98,94% (Ferrag, Maglaras, Moschoyiannis, et al., 2019).

1.3. Bases de dados

Os dados são um componente crucial de qualquer sistema de aprendizagem máquina, e a cibersegurança não é exceção. Os investigadores de cibersegurança precisam de acesso a grandes quantidades de dados para desenvolver e avaliar modelos e soluções de aprendizagem máquina.

Esta secção fornecerá uma visão geral de algumas das bases de dados (*datasets*) de cibersegurança mais utilizados de acordo com os artigos que foram tomados como base de pesquisa para a presente secção. Apresentam-se as Tabela 5, Tabela 6, Tabela 7 e Tabela 8 com a comparação do tema abrangido pelos vários artigos. A divisão adotada foi adaptada do artigo (Ferrag, Maglaras, Moschoyiannis, et al., 2019) e tem em conta o tipo de tráfego recolhido presente em cada base de dados, no entanto ressalva-se que a divisão das bases de dados pode ser feita de outra forma uma vez que certas bases de dados contêm diversos tipos de dados recolhidos ou até mesmo usando outros critérios. Certos conjuntos de dados não têm presença assinalada em nenhum dos artigos analisados por se tratarem ou de bases de dados

menos populares ou por serem bases de dados recentes com poucos estudos realizados.

O artigo (Akhtar & Feng, 2022) menciona apenas que utilizou no seu modelo os datasets contidos no repositório *Kaggle*.

(Li et al., 2021) aprofunda as características e onde já foram utilizados os *datasets* KDDCUP 99 e NSL-KDD. (Li et al., 2021) menciona ainda alguns *datasets* utilizados para construção de modelos de aprendizagem profunda em outros artigos.

(Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020) apresenta uma breve explicação de alguns *datasets* usados frequentemente, não referindo pormenores sobre cada um dos conjuntos de dados.

(Berman et al., 2019) menciona com formalismo os *datasets* KDDCUP 99 e CTU-13 referindo aplicações relevantes destes conjuntos de dados em modelos de aprendizagem máquina. Revela ainda os conjuntos de dados *Alexa Top Sites*, *Comodo, Contagio* e *Virus Share*, *Virus Total*, DREBIN e *Microsoft* usados frequentemente com o objetivo de examinar modelos de deteção de malware, fornecendo também breves informações sobre outros *datasets*, com um formalismo menor. A destacar a menção do *CERT Insider Threat Dataset*, um conjunto de dados para deteção de ameaças internas.

Em (Gümüsbas et al., 2020) são apresentadas as características de vários *datasets*, sendo em alguns apresentadas as referências de modelos que utilizaram os *datasets*.

(Ferrag, Maglaras, Moschoyiannis, et al., 2019) revela-se um artigo bastante completo que refere as características dos *datasets*, mencionando vários modelos que utilizam os diferentes *datasets*. Apresenta também uma tabela comparativa com os datasets e as suas características gerais. Algumas bases de dados são apenas mencionadas na tabela comparativa. Estudos no próprio artigo utilizam alguns dos *datasets*.

(Dasgupta et al., 2022) e (Shaukat, Luo, Varadharajan, Hameed, Chen, et al., 2020) dão um breve contexto dos *datasets*, e apresentam vários modelos de outros artigos ao longo do artigo, acompanhado por tabelas comparativas.

(Mijwil et al., 2023) realiza a explicação de algumas bases de dados acompanhada com um gráfico que apresenta o número de amostras para cada ataque em cada um dos *datasets* falados. Compara ainda estudos onde menciona datasets para além dos analisados.

(Alshaibi et al., 2022) apresenta uma tabela com uma pequena explicação para cada base de dados apresentada.

Artigo	(Akhtar & Feng, 2022)	(Li et al., 2021)	(Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)	(Berman et al., 2019)	(Gümüşbas et al., 2020)	(Ferrag, Maglaras, Moschoyiannis, et al., 2019)	(Dasgupta et al., 2022)	(Shaukat, Luo, Varadharajan, Hameed, Chen, et al., 2020)	(Mijwil et al., 2023)	(Alshaibi et al., 2022)	Presente Dissertação
Base de dados baseada em tráfego de rede											
ADFA/ADFA-Linux (*ADFA IDS Datasets | UNSW Research*, 2013)	✗	✓	✓	✗	✓	✓	✓	✓	✗	✗	✓
AWID2018 (CSE-CIC-IDS2018) (*IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2018)	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓
CAIDA (*CAIDA Data - Completed Datasets - CAIDA*, 2020)	✗	✗	✗	✗	✓	✓	✓	✗	✗	✓	✓
CDX (*"Cyber Defense Exercise (CDX) 2009 Data" by Erik Dean, Thomas Cook et Al.*, 2009)	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓
CERT Insider Threat Dataset v6.2 (*Insider Threat Test Dataset*, 2016)	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓
CIC Bell DNS 2021 (*CIC-Bell-DNS 2021 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2021)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
CIC DoS (*DoS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2017)	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓
CICIDS2017 (*IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2017)	✗	✓	✓	✗	✓	✓	✗	✗	✓	✗	✓
CTU-13 (*The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background Traffic. — Stratosphere IPS*, 2013)	✗	✗	✓	✓	✓	✓	✓	✗	✗	✓	✓
DARPA IDS (*1998 DARPA Intrusion Detection Evaluation Dataset | MIT Lincoln Laboratory*, 1998)	✗	✗	✓	✗	✓	✓	✓	✓	✗	✓	✓
DDoS Evaluation Dataset (CIC-DDoS2019) (*DDoS 2019 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2019)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
ISCX (ISCXIDS2012) (*IDS 2012 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2012)	✗	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓
KDDCUP 99 (*KDD Cup 1999 Data*, 1999)	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Kyoto (*Traffic Data from Kyoto University's Honeypots*, 2016)	✗	✓	✗	✗	✓	✓	✓	✗	✗	✗	✓
NSL-KDD (*NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2009)	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SpamBase (*UCI Machine Learning Repository: Spambase Data Set*, 1999)	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓
SSHCure (*SSH Datasets - SimpleWiki*, 2014)	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓
UNSW-NB-15 (*The UNSW-NB15 Dataset | UNSW Research*, 2021)	✗	✓	✓	✗	✓	✓	✓	✗	✓	✗	✓
VirusShare (*VirusShare.Com*, n.d.)	✗	✗	✓	✓	✗	✗	✗	✓	✗	✗	✓

Tabela 5: Bases de dados baseadas em tráfego de rede nos artigos analisados

Artigo	(Akhtar & Feng, 2022)	(Li et al., 2021)	(Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)	(Berman et al., 2019)	(Gümüşbas et al., 2020)	(Ferrag, Maglaras, Moschoyiannis, et al., 2019)	(Dasgupta et al., 2022)	(Shaukat, Luo, Varadharajan, Hameed, Chen, et al., 2020)	(Mijwil et al., 2023)	(Alshaibi et al., 2022)	Presente Dissertação
Base de dados baseada em tráfego internet											
Botnet (*Botnet 2014 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2014)	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✓
CIC MalMem 2022 (*Malware Memory Analysis | Datasets | Canadian Institute for Cybersecurity | UNB*, 2022)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Comodo Antivirus Database (*Comodo Anti Malware Database Latest Version & Additions 2022*, 2022)	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✓
Darknet 2020 (*Darknet 2020 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2020)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
DNS over HTTPS (CIRA-CIC-DoHBrw2020) (*DoHBrw 2020 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2020)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Evasive-PDF Mal 2022 (*CIC-Evasive-PDFMal2022 | Datasets | Canadian Institute for Cybersecurity | UNB*, 2022)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Heritrix (*Heritrix - Home Page*, 2010)	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓
HTTP CSIC-2010 (*HTTP DATASET CSIC 2010*, 2010)	✗	✗	✓	✗	✓	✗	✓	✓	✗	✗	✓
ISOT (*Botnet and Ransomware Detection Datasets | ISOT Research Lab*, 2010)	✗	✗	✓	✗	✓	✓	✓	✗	✗	✓	✓
MAWI (*MAWI Working Group Traffic Archive*, 2023)	✗	✗	✗	✗	✓	✓	✗	✗	✗	✓	✓
Microsoft Malware (BIG 2015) (*Microsoft Malware Classification Challenge (BIG 2015) | Kaggle*, 2015)	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✓
Tor-nonTor (ISCXTor2016) (*Tor 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2016)	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✓
UGR (*UGR'16 Dataset*, 2016)	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓
UMASS (*Home Page - UMass Trace Repository*, n.d.)	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓
URL (ISCX-URL2016) (*URL 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2016)	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✓

Tabela 6: Bases de dados baseadas em tráfego de internet nos artigos analisados

Artigo	(Akhtar & Feng, 2022)	(Li et al., 2021)	(Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)	(Berman et al., 2019)	(Gümüsbas et al., 2020)	(Ferrag, Maglaras, Moschoyiannis, et al., 2019)	(Dasgupta et al., 2022)	(Shaukat, Luo, Varadharajan, Hameed, Chen, et al., 2020)	(Mijwil et al., 2023)	(Alshaibi et al., 2022)	Presente Dissertação
Base de dados baseada em tráfego móvel											
Adagio (*GitHub - Hgascon/Adagio: Structural Analysis and Detection of Android Malware*, 2013)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Android Adware and General Malware Dataset (CIC-AAGM2017) (*Android Adware 2017| Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2017)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Android Malware (CICAndMal2017) (*Android Malware 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2017)	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗	✓
Android Validation (*Android Validation | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2014)	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗	✓
CCCS-CIC-AndMal2020 (*AndMal 2020 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2020)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
CICMalDroid 2020 (*MalDroid 2020 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2020)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Drebin (*GitHub - Sontung/Drebin-Malwares: Malware Detection Using the Drebin Dataset*, 2014)	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✓
Investigation of the Android Malware (CIC-InvesAndMal2019) (*Investigation on Android Malware 2019 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2019)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
ISCX Android Botnet dataset 2015 (*Android Botnet 2015 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2015)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Kharon Malware (*Kharon Malware Dataset*, 2016)	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓
Malheur (*GitHub - Rieck/Malheur: A Tool for Automatic Analysis of Malware Behavior*, 2011)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

Tabela 7: Bases de dados baseadas em tráfego móvel nos artigos analisados

Artigo	(Akhtar & Feng, 2022)	(Li et al., 2021)	(Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)	(Berman et al., 2019)	(Gümüsbas et al., 2020)	(Ferrag, Maglaras, Moschoyiannis, et al., 2019)	(Dasgupta et al., 2022)	(Shaukat, Luo, Varadharajan, Hameed, Chen, et al., 2020)	(Mijwil et al., 2023)	(Alshaibi et al., 2022)	Presente Dissertação
Base de dados baseada em tráfego IoT											
Bot-IoT (*The Bot-IoT Dataset | UNSW Research*, 2018)	✗	✗	✓	✗	✗	✓	✗	✗	✓	✓	✓
DS2OS Traffic Traces (*DS2OS Traffic Traces | Kaggle*, 2018)	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓
CIC IoT Dataset 2023 (*IoT Dataset 2023 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2023)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Enriching IoT Datasets (*Enriched Dataset | Datasets | Canadian Institute for Cybersecurity | UNB*, 2021)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
TON_IoT (*The TON_IoT Datasets | UNSW Research*, 2016)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
WSN-DS (*WSN-DS | Kaggle*, 2016)	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
Base de dados baseada em rede elétrica e controlos industriais											
ICS cyber attack (*Tommy Morris - Industrial Control System (ICS) Cyber Attack Datasets*, 2015)	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓
IEEE 300-bus power test system (*IEEE 300-Bus System*, n.d.)	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓
LBNL (*LBNL Power Data | Berkeley Lab Cybersecurity R&D*, 2016)	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✓
Base de dados baseada em rede privada virtual											
VPN-nonVPN (ISCXVPN2016) (*VPN 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2016)	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓
Outros											
Contagio Malware Dump (*Contagio*, 2023)	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✓
Kaggle (*Kaggle: Your Machine Learning and Data Science Community*, 2023)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Machine Learning-Based NIDS Datasets (*ML-Based NIDS Datasets*, 2022)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

Tabela 8: Comparação da presença de bases de dados nos artigos analisados

Após a análise dos diversos artigos irão ser mencionados alguns dos conjuntos de dados mais relevantes e que têm desempenhado um papel significativo no desenvolvimento e avaliação de sistemas de deteção de intrusão. Todos os repositórios de dados apresentados são *open-sorce*.

I. CIC-IDS-2017 e CSE-CIC-IDS2018

ICIC IDS 2017 (*IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2017) e CSE-CIC-IDS2018 (Sharafaldin et al., 2018) são conjuntos de dados de deteção de intrusão de rede que foi criado pelo Instituto Canadiano de Cibersegurança (CIC) em 2017 e 2018 respetivamente. Os conjuntos de dados são constituídos por tráfego de rede benigno e maligno. Incluem vários tipos de tráfego de rede, como HTTP, FTP e SSH, e inclui ataques como DoS, verificação e ataques de força bruta. CSE-CIC-IDS2018 (*IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2018) é uma versão atualizada do conjunto de dados anterior. Inclui um maior número de registos e mais tipos de ataques, como ataques à Web e ataques específicos da IoT (Yadav et al., 2020)

II. NSL-KDD

NSL-KDD (Tavallaee et al., 2009) é um conjunto de dados de referência para deteção de intrusão de rede que foi criado através do processamento do conjunto de dados original KDD Cup 1999 (*NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, 2009). Contém uma versão pré-processada do conjunto de dados KDD Cup com dimensionalidade reduzida e um número menor de registos, devido à remoção de erros e repetições (Yadav et al., 2020). O conjunto de dados inclui vários tipos de ataques como DOS, R2L, U2R, entre outros, juntamente com rótulos que indicam se o tráfego é normal ou um ataque. No entanto não deixa de ser um repositório de dados antigo uma vez que é uma filtragem dos dados do *dataset* KDD Cup 1999.(Yadav et al., 2020)

III. UNSW-NB-15

UNSW-NB15 (Moustafa & Slay, 2015) (Moustafa et al., 2019) (Moustafa & Slay, 2016) (Moustafa et al., 2017) (Sarhan et al., 2020) é um conjunto de dados recolhido no Australian Centre for Cyber Security (ACCS) pelo grupo de investigação em cibersegurança em 2015. Os dados brutos de 100 GB foram recolhidos usando as ferramentas TCP dump e Ixia PerfectStorm, contendo tanto o tráfego normal da rede como os ataques. Os dados foram gerados ao longo de dois períodos diferentes de simulação, um de 15 horas e outro de 16

horas.

Na base de dados existem aproximadamente 2,5 milhões de registos, com 49 atributos extraídos usando as ferramentas *Bro-IDS*, *Argus* e alguns algoritmos recém-desenvolvidos.

As características do UNSW-NB15 estão divididas em cinco conjuntos: características temporais, características de conteúdo, características de fluxo, características básicas e características originárias adicionais. Além dessas características, há duas variáveis alvo: *attack_cat*, que pode ser o estado normal ou o nome da categoria de ataque, e *label*, que é 1 para tráfego anormal e 0 para tráfego normal. Existem nove tipos de ataques, incluindo Worms, Shellcode, Reconnaissance, Analysis, Generic, Backdoor, DoS, Exploits e Fuzzers (Yadav et al., 2020).

Estes conjuntos de dados têm sido amplamente utilizados na comunidade de investigação e têm desempenhado um papel crucial no desenvolvimento e avaliação de sistemas de deteção de intrusão.

A base de dados selecionada para o treino e teste dos classificadores desenvolvidos nesta dissertação é a UNSW-NB15. Esta escolha baseia-se na ampla notoriedade e referência desta base de dados na literatura, tendo sido citada em diversos artigos. Desta forma proporciona-se um valioso suporte para comparação com os resultados que serão obtidos neste estudo. A base de dados revela-se desafiante devido ao seu conjunto pré-selecionado de amostras criado pelos autores, conjunto esse que irá ser utilizado no treino e teste dos modelos na presente dissertação. Além disso, a UNSW-NB15 é uma base de dados concebida num ciberrange, assemelhando-se ao que se pretende criar no projeto ciberrange da Escola Naval.

Possuí uma grande diversidade de ataques que servem de apoio para o crescimento contínuo da plataforma ciberrange, permitindo a adição progressiva de novos tipos de ataques, ampliando a capacidade da plataforma para replicar de maneira eficaz os cenários presentes na base de dados selecionada.

1.4. Métodos de avaliação da solução

Avaliar o desempenho de modelos de aprendizagem máquina é uma etapa crucial para garantir que os modelos são eficazes em aplicações do mundo real. Ao utilizar métodos de avaliação apropriados, os modelos de aprendizagem máquina podem ser otimizadas para melhor desempenho e precisão, levando a soluções mais eficazes para problemas do mundo real.

A presente secção tem como objetivo fornecer uma visão geral dos vários métodos de

avaliação de soluções utilizados em aprendizagem máquina.

De acordo com os artigos que foram tomados como base de pesquisa para a realização deste capítulo da presente dissertação apresenta-se a Tabela 9 com a comparação da presença do tema nos vários artigos analisados.

Artigo	(Wazid et al., 2022)	(Akhtar & Feng, 2022)	(Li et al., 2021)	(Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)	(Berman et al., 2019)	(Powers, 2007)	(Hnamte et al., 2023)	**Presente Dissertação**
Termos TP, TN, FP, FN	✓	✓	✗	✓	✓	✓	✓	✓
Matriz de Confusão	✗	✓	✗	✓	✓	✗	✗	✓
Precisão	✓	✓	✓	✓	✓	✓	✓	✓
***Recall* / *True Positive Rate* (TPR)**	✓	✓	✓	✓	✓	✓	✓	✓
Exatidão / *Accuracy*	✓	✓	✓	✓	✓	✓	✓	✓
Especificidade / *True Negative Rate* (TNR)	✗	✗	✗	✓	✓	✓	✗	✓
Taxa de Erro Global	✗	✗	✗	✓	✗	✗	✗	✓
Fall Out / *False Positive Rate* (FPR)	✗	✗	✓	✓	✓	✓	✓	✓
Miss Rate / *False Negative Rate* (FNR)	✗	✗	✓	✓	✗	✓	✗	✓
***False Discovery Rate* (FDR)**	✗	✗	✗	✓	✗	✗	✗	✓
***False Omission Rate* (FOR)**	✗	✗	✗	✓	✗	✗	✗	✓
***Negative Predictive Value* (NPV)**	✗	✗	✗	✗	✓	✗	✗	✓
F1-*Score*	✓	✗	✓	✓	✓	✗	✓	✓
G-*Mean*	✗	✗	✗	✓	✗	✗	✗	✓
Curva de Características Operacionais Recebidas (ROC)	✗	✗	✗	✓	✓	✓	✗	✓
Área Sob a Curva (AUC)	✗	✗	✓	✓	✓	✓	✗	✓

Tabela 9: Comparação da abordagem da matriz de confusão nos artigos analisados

Para os termos verdadeiros positivos (TP, sigla inglesa), verdadeiros negativos (TN, sigla inglesa), falsos positivos (FP, sigla inglesa) e falsos negativos (FN, sigla inglesa) a maioria

dos artigos apresenta uma explicação razoável, onde a diferença está no formalismo que utiliza. No caso de (Berman et al., 2019) é mostrada apenas a tabela clássica de explicação da matriz de confusão. No entanto há a referir que em (Li et al., 2021) e em (Akhtar & Feng, 2022) os termos são pouco ou nada referidos.

Relativamente às métricas da matriz de confusão (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020) revelou-se o artigo mais completo com um elevado número de métricas para análise da matriz de confusão. Em (Li et al., 2021) e em (Akhtar & Feng, 2022) não há nenhuma explicação individual para métricas específicas, sendo que algumas das métricas são utilizadas na comparação entre modelos mencionados ao longo dos artigos.

O artigo (Powers, 2007) fornece uma visão crítica das métricas tradicionais de avaliação, fornecendo uma avaliação mais completa e informada de classificadores baseados em redes neurais.

Um artigo recente de (Hnamte et al., 2023) descreve o desenvolvimento de um modelo de deteção de intrusão, que destaca a adequação das métricas utilizadas com base na revisão da literatura realizada.

(Wazid et al., 2022) e (Berman et al., 2019) apresentam também algumas métricas com um formalismo elevado, no entanto não apresentam uma gama tão vasta quanto (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020).

Pela análise dos diversos artigos conclui-se que as métricas mais favoráveis para a avaliação e comparação dos modelos classificadores baseados em redes neuronais são a F1-*score*, Exatidão, Precisão, *False Positive Rate* (FPR)/ *False Alarm Rate* e *True Positive Rate* (TPR). As métricas selecionadas têm sido amplamente utilizadas para avaliar a eficácia dos sistemas de deteção de intrusão (Hnamte et al., 2023).

No Capítulo 1: Revisão da Literatura respondeu-se à questão: quais são as práticas utilizadas em classificadores semelhantes construídos em projetos anteriores? Constata-se a presença de diversas abordagens empregadas em projetos prévios para desenvolver classificadores. Começou-se com a identificação de variados tipos de ataques, verificando-se que existem múltiplas maneiras de categorizá-los. Foram adotadas as categorias de Malware, DoS, MITM, Eavesdropping (Sniffing/Snooping), Ataque de acesso privilegiado e Ataques dia-zero e apresentada a presença do uso em modelos de aprendizagem máquina de cada tipo de ataques em artigos selecionados.

Avaliou-se o desempenho de técnicas de aprendizado de máquina empregadas em projetos anteriores, e determinou-se que as abordagens mais adequadas para o contexto do projeto ciberrange da Escola Naval seriam FFNN, DNN, LSTM e CNN-LSTM.

Realizou-se uma análise das bases de dados de código aberto disponíveis, onde foi selecionada a base de dados UNSW-NB15 por ser a mais pertinente no contexto do projeto.

Também examinamos as métricas frequentemente utilizadas na literatura para avaliar os modelos classificadores criados. Desta análise, concluímos que as métricas mais apropriadas seriam F1-*score*, Exatidão, Precisão, Taxa de Falsos Positivos (FPR)/Taxa de Alarmes Falsos e Taxa de Verdadeiros Positivos (TPR).

Cap. 2: Cibersegurança no contexto das organizações

Capítulo 2: Cibersegurança no contexto das organizações

2.1. Importância

A cibersegurança tornou-se um problema crítico na era digital atual, onde organizações de todos os tamanhos enfrentam ameaças crescentes de *hackers*, cibercriminosos e outras entidades mal-intencionadas (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020). Como tal, é essencial que as organizações compreendam a importância de proteger os seus dados e informações contra estas ameaças. Neste capítulo, discutiremos a relevância da cibersegurança no contexto das organizações, a importância de proteger segredos, o papel dos *hackers* e a importância de ter um ambiente de treino numa organização.

A atual era digital testemunhou um aumento exponencial de ciberameaças, incluindo *hacking*, violações de dados, ataques de *ransomware*, entre outros. O relatório anual da IBM (IBM, 2023) refere que em 2022, Portugal foi o terceiro país da Europa com mais ciberataques (9% dos ataques em toda a Europa). Várias organizações são atacadas, não escapando nem as grandes empresas como a Vodafone que em 2022 sofreu um grande ataque que afetou os serviços de rede móvel da empresa (Caçador, 2022; Vodafone Portugal, 2022).

As organizações enfrentam agora desafios sem precedentes para manter os seus dados seguros. As organizações possuem dados confidenciais, incluindo segredos comerciais, informações financeiras, dados de clientes, entre outros (Goutam, 2015). Se estes dados são comprometidos, podem ter consequências graves para a organização, incluindo perda de negócios, danos à reputação e implicações legais. Portanto, é crucial que as organizações protejam os seus segredos de entidades mal-intencionadas (Goutam, 2015).

Os *hackers* são indivíduos ou grupos que exploram vulnerabilidades em sistemas informáticos para obter acesso não autorizado. Os *hackers* podem causar danos significativos às organizações roubando informações confidenciais, corrompendo dados ou interrompendo operações (E. S. Raymond, 2013). Como tal, as organizações devem estar vigilantes e tomar as medidas adequadas para prevenir ou mitigar o impacto dos ciberataques.

Uma das formas mais eficazes de proteção contra ciberataques é formar o pessoal sobre como identificar e responder a potenciais ameaças. Ter um ambiente de treino dedicado, como o Ciberrange, pode fornecer ao pessoal cenários realistas para praticar e aperfeiçoar as suas habilidades (Karjalainen & Kokkonen, 2020; Vykopal et al., 2017). Da mesma forma

que pode ajudar a garantir que o pessoal está mais bem equipado para lidar com ciberameaças, reduzindo assim o risco de ataques bem-sucedidos.

2.2. Marinha portuguesa

O Ciberrange da Escola Naval será uma infraestrutura de treino projetada para fornecer cenários realistas para o treino de pessoal em cibersegurança. A infraestrutura será fundamental para a Marinha, pois permitirá que o pessoal desenvolva e aperfeiçoe as suas habilidades num ambiente seguro e controlado. Esta infraestrutura pode ajudar a preparar a Marinha para potenciais ciberataques, que podem ter graves consequências para a segurança nacional. Ao investir na infraestrutura Ciberrange, a Marinha pode garantir que seu pessoal esteja mais bem equipado para lidar com ciberameaças e proteger-se contra possíveis ataques.

Além disso, o Ciberrange pode ser usado para testar novas tecnologias e estratégias, como por exemplo a recolha de dados na presente dissertação, proporcionando assim à Marinha uma vantagem na deteção de intrusão e ciberataques.

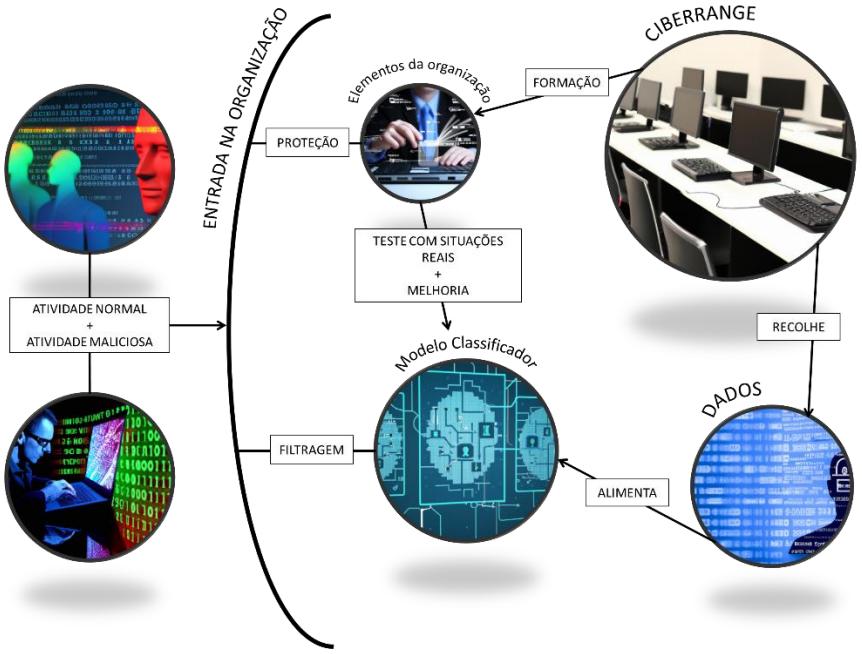

Figura 2: Esquema representativo das funcionalidades do ciberrange no contexto da presente dissertação

Nota: Imagens do esquema são geradas com inteligência artificial pela ferramenta DALL-E

A Figura 2 apresenta em forma de esquema a ideia do projeto ciberrange, onde o ciberrange em si, funciona como uma infraestrutura que serve tanto de formação dos elementos da organização, como de recolha de dados da atividade. Estes dados recolhidos

alimentam um modelo classificador que por sua vez é testado em ambiente controlado pelos elementos da organização, permitindo a constante melhoria do classificador e novamente das competências dos elementos da organização. Assim a atividade tanto normal, como maliciosa é filtrada pelo modelo classificador e a organização está protegida por elementos conscientes para as ameaças exteriores que poderão causar dados à organização.

Ao longo do capítulo 2: Cibersegurança no contexto das organizações procurou-se responder à questão: como identificar a melhor infraestrutura para a recolha de dados e ainda capaz de promover a preparação e proteção de uma organização relativamente à cibersegurança? Analisaram-se as exigências de uma instituição relativamente ao treino do pessoal para a deteção e reação a ciberameaças, chegou-se à conclusão de que a implementação de um ciberrange representaria a infraestrutura mais apropriada para fomentar a preparação e salvaguarda da organização no âmbito da cibersegurança, ao mesmo tempo em que possibilitaria a recolha de dados.

Dentro deste capítulo, também foi possível constatar a relevância da cibersegurança no contexto de uma organização, especialmente no caso da Marinha Portuguesa. Constatou-se que a cibersegurança é uma questão crítica para as organizações e é essencial proteger os dados sensíveis contra ciberameaças. Os *hackers* podem causar danos significativos às organizações, e é crucial ter um ambiente de treino para treinar o pessoal para identificar e responder a ameaças potenciais. O Ciberrange da Escola Naval é uma infraestrutura essencial que pode ajudar a Marinha a desenvolver e melhorar as suas capacidades de cibersegurança. É fundamental investir nesta infraestrutura para garantir que a Marinha esteja preparada para lidar com ciberameaças e proteger a segurança nacional.

Por último, foi apresentado e detalhado um esquema visual que representa a conceção do ciberrange, destacando suas capacidades e a troca interna de informações.

Cap. 3: Ciberrange

Capítulo 3: Ciberrange

Um ciberrange é um ambiente controlado e isolado projetado para treino e exercícios de cibersegurança. Proporciona uma plataforma segura e realista para os profissionais de cibersegurança ou elementos de uma organização desenvolverem e melhorarem as suas competências de defesa contra ciberameaças (Park et al., 2022; Urias et al., n.d.; Vykopal et al., 2017). Nesta secção, irão ser exploradas as características essenciais que um ciberrange deve ter para simular eficazmente cenários de cibersegurança do mundo real e fornecer experiências de treino enriquecedoras.

(Karjalainen & Kokkonen, 2020) refere que as características mais relevantes num ciberrange são:

Realismo: O realismo é uma característica crucial de ciberrange. O ciberrange deve replicar com precisão redes, sistemas e aplicações do mundo real, incluindo as suas configurações, vulnerabilidades e interações. Isto garante que os exercícios de formação estão o mais próximos possível de cenários da vida real, proporcionando oportunidades de aprendizagem práticas e relevantes.

Ambiente Isolado e Controlado: O ciberrange deve ser isolado das redes e sistemas de suporte para evitar qualquer impacto acidental ou intencional nos ambientes operacionais. Deve dispor de controlos de acesso sólidos e de mecanismos de monitorização que garantam um ambiente controlado para os exercícios de formação. Desta forma tem-se um ambiente seguro sem representar riscos para os sistemas ou dados reais.

Simulação da Internet: O ciberrange deve ter a capacidade de simular a Internet global com suas estruturas e serviços. A simulação da Internet permite cenários de treino realistas que envolvem a análise e a resposta a ciberameaças de diferentes tipos, bem como melhor simular as vulnerabilidades associadas à mesma, imitando a infraestrutura onde os ataques são realizados no mundo real.

Criação de Tráfego de Utilizador e Rede: O ciberrange deve ser capaz de gerar tráfego de utilizador e rede realista para simular cenários do mundo real. Deve simular diferentes tipos de tráfego de rede, como navegação na Web, comunicação por e-mail, transferências de arquivos e outras atividades que ocorrem em ambientes de rede típicos. Isto ajuda os elementos da organização a ganhar experiência prática na vigilância e análise do tráfego de rede para identificar potenciais ciberameaças e vulnerabilidades.

Execução e Simulação de Ataques: O ciberrange deve permitir a execução e simulação de

vários tipos de ciberataques. Deve conter exercícios ofensivos e defensivos, onde os elementos da organização podem simular e responder aos diferentes cenários de ciberataque. O ciberrange deve fornecer uma ampla gama de ferramentas e técnicas de ataque para imitar as ameaças do mundo real, incluindo *malware*, *ransomware*, ataques *Distributed Denial of Service* (DDoS), ataques de engenharia social, entre outros. Desta forma permitir-se-à formação prática na deteção, mitigação e resposta a diferentes tipos de ciberataques.

Infraestruturas das organizações: O ciberrange deve incluir ambientes organizacionais variados, como diferentes tipos de redes, sistemas e aplicações usualmente encontrados nas organizações. Assim permite-se que os elementos da organização entendam os desafios e vulnerabilidades únicas associadas a diferentes ambientes organizacionais e desenvolvam competências de segurança específicas para cada cenário.

Colaboração: O ciberrange deve ter a capacidade de colaboração e cooperação com outras plataformas de formação, isto é, a capacidade de partilhar e trocar recursos, cenários e dados de treino com outras plataformas de treino. A colaboração permite exercícios conjuntos, *red teaming* e partilha de informações, o que melhora a experiência de aprendizagem e ajuda os elementos da organização a desenvolverem competências de trabalho em equipa e coordenação na resposta a ciberameaças.

Planeamento, Execução, Monitorização e Análise: O ciberrange deve fornecer ferramentas e capacidades para planear, executar, monitorizar e analisar exercícios de cibersegurança. Desta forma são necessários recursos como planeamento de exercícios, criação de cenários, execução de exercícios, vigilância e registo de atividades e análise detalhada de resultados. Assim garante-se que os elementos da organização projetem, executam e avaliam exercícios de cibersegurança com base em cenários da vida real, fornecendo *feedback* para melhorar as suas habilidades e conhecimentos.

Ao longo do Capítulo 3: Ciberrange respondeu-se à questão quais são os elementos-chave de uma infraestrutura que recolhe dados e apoia a aprendizagem colaborativa de preparação para a cibersegurança, levantada na introdução. Ficou evidente que um ciberrange, a fim de ser uma infraestrutura de formação para os membros de uma organização e ao mesmo tempo efetuar uma recolha de dados relevante num ambiente que simule o mundo real, deve possuir elementos-chave cruciais. São eles: realismo, ambiente isolado e controlado, deve ter a capacidade de simular a internet, capacidade de criar tráfego de utilizador e de rede, executar e simular ataques, simular a infraestrutura da organização, capacidade de colaboração e cooperação com outras plataformas e por fim ter ferramentas que permitam planeamento, execução, monitorização e análise da atividade no ciberrange. A incorporação destes

elementos-chave no ciberrange da Escola Naval torna-se muito desafiante, pelo que se antecipa que sejam gradualmente incorporadas e aprimoradas ao longo do desenvolvimento do projeto, visando no futuro obter um ciberrange robusto, dinâmico e funcional.

Cap. 4: Classificador baseado em redes neuronais

4.1. Descrição das ferramentas

4.2. Processamento dos dados

4.3. Descrição dos modelos

4.4. Treino dos modelos

4.5. Métodos de avaliação da solução

Capítulo 4: Classificador baseado em redes neuronais

A construção de modelos classificadores eficientes e precisos torna-se fundamental para a deteção e prevenção de ciberataques. Como já foi verificado no Capítulo 1: Revisão da Literatura, as redes neurais têm se destacado como uma poderosa abordagem no campo da aprendizagem de máquina, possibilitando a criação de modelos capazes de identificar padrões e comportamentos maliciosos em grandes volumes de dados.

O presente capítulo tem como objetivo apresentar a arquitetura e os procedimentos utilizados na construção de modelos classificadores de ciberataques baseados em redes neuronais. Serão abordados conceitos fundamentais das redes neurais, bem como os passos seguidos para a construção dos modelos. Além disso, serão exploradas técnicas de pré-processamento de dados e avaliação de desempenho dos modelos, com o objetivo de melhorar a deteção e minimizar falsos positivos.

O capítulo começa pela descrição das ferramentas utilizadas na construção dos modelos, de seguida descreve o processamento inicial dos dados. Descreve separadamente as arquiteturas abordadas bem como os parâmetros utilizados no treino e compilação dos modelos. Por fim, refere as métricas de avaliação selecionadas para comparar os modelos entre si, bem como outros modelos já existentes.

A base de dados selecionada são os 10% da base de dados UNSW-NB15 pré-selecionados pelos autores para treino e teste. A base de dados é abordada de três formas sendo elas a classificação binária, isto é, entre amostra normal e de ataque e a classificação multi-classe, com o objetivo de detetar o tipo de ataque. A Classificação multi-classe divide-se ainda numa classificação com todos os tipos de ataques existentes e uma classificação com redução de ataques onde minoritários que representam menos de 2% do conjunto de dados são removidos.

A base de dados UNSW-NB15 é uma das mais desafiantes da literatura e a sua escolha para treino e teste dos modelos da presente dissertação deve-se a ser uma base de dados muito utilizada na literatura, como já foi referido na Revisão da Literatura, sendo constituída por dados relativamente recentes e com conjuntos de treino e teste com amostras selecionadas.

O código utilizado encontra-se disponível na plataforma *github* através da referência (*Correiagoncalves/Classificador_UNSWNB15*, 2023).

4.1. Descrição das ferramentas

O processamento, análise da base de dados, treino e teste dos modelos classificadores são conduzidos usando Python versão 3.10.4 com recurso e apoio das bibliotecas numpy (*NumPy User Guide — NumPy v1.25 Manual*, n.d.), pandas (*User Guide — Pandas 2.0.3 Documentation*, n.d.), matplotlib (*Users Guide — Matplotlib 3.7.2 Documentation*, n.d.), seaborn(*User Guide and Tutorial — Seaborn 0.12.2 Documentation*, n.d.),.sklearn (*User Guide: Contents — Scikit-Learn 1.3.0 Documentation*, n.d.) e tensorflow(*TensorFlow Core*, n.d.).

4.2. Processamento dos dados

4.2.1. Classificação multi-classe

O processamento da classificação multi-classe foi realizado tanto com as amostras originais de ataques da base de dados UNSW-NB15 (*Normal*, *Generic*, *Exploits*, *Fuzzers*, *DoS*, *Reconnaissance*, *Analysis*, *Backdoor*, *Shellcode*, *Worms*) como procedendo à remoção de ataques minoritários que representavam menos de 2% da base de dados. Os ataques removidos foram *Analysis*, *Backdoor*, *Shellcode*, *Worms*.

A Tabela 10 apresenta os tipos de ataques bem como as quantidades de amostras para cada tipo antes de ser procedido a qualquer processamento no conjunto de treino.

Tipo de ataque	Nº. de amostras	Explicação
Normal	56000	Tráfego de rede considerado normal.
Generic	40000	Ataques genéricos que não se enquadram em outras categorias específicas.
Exploits	33393	Ataques que usam vulnerabilidades conhecidas para obter acesso não autorizado.
Fuzzers	18184	Envio de entradas inválidas ou aleatórias com o de forma a encontrar comportamentos inesperados.
DoS	12264	Sobrecarga de um sistema, serviço ou rede para que se torne inacessível para usuários legítimos
Reconnaissance	10491	Obtenção de informações sobre uma rede ou sistema alvo de forma a explorar fraquezas na segurança.
Analysis	2000	Recolha e análise de informações sobre uma rede ou sistema alvo de forma a identificar vulnerabilidades ou obter informações sensíveis
Backdoor	1746	Acesso não autorizado a um sistema ou rede contornando por autenticação e segurança.
Shellcode	1133	Código malicioso inserido em *shell*.
Worms	130	Programas com propagação automática através de redes.

Tabela 10: Ataques presentes na base de dados UNSW-NB15

I. Remoção de ataques minoritários

O processamento dos dados da base de dados começa por remover as colunas que são menos relevantes, devido ao facto que a literatura já existente revelou benéfico um número mais reduzido de colunas na base de dados adotada. Entre as várias abordagens possíveis de seleção das melhores colunas, adotaram-se as colunas selecionadas pelo estudo realizado por (Y. Yin et al., 2023) que utilizou um método híbrido de seleção de colunas chamado IGRF-RFE. Este modelo selecionou as melhores colunas da base de dados em questão, com o objetivo de obter os melhores resultados num classificador semelhante ao que se está a construir na presente dissertação. A Tabela 11 apresenta as colunas selecionadas para treino e teste dos classificadores.

COLUNA	EXPLICAÇÃO
DUR	Duração total da gravação
PROTO	Tipo de protocolo (como TC, UDP)
SERVICE	Serviço (como http, ftp, smtp, ssh, dns e ftp-data)
STATE	Estado do pacote e seu protocolo dependente (tal como ACC, CLO e CON)
SPKTS	Contagem de pacotes de origem para destino
DPKTS	Contagem de pacotes do destino para a origem
SBYTES	Bytes de origem para destino
DBYTES	Bytes do destino para a origem
RATE	Taxa de transferência da conexão
STTL	Tempo de vida da origem ao destino
DTTL	Tempo de vida do destino à origem
DLOAD	Bits de destino por segundo
DLOSS	Pacotes de destino retransmitidos ou descartados
SINPKT	Instante de chegada do interpacote de origem (mSec)
DINPKT	Instante de chegada interpacote de destino (mSec)
DJIT	Desvio de destino (mSec)
TCPRTT	Tempo de ida e volta de configuração da conexão TCP, a soma de 'synack' e 'ackdat'
SYNACK	Tempo de configuração da conexão TCP, o tempo entre o SYN e os pacotes SYN_ACK
ACKDAT	Tempo de configuração da conexão TCP, o tempo entre o SYN_ACK e os pacotes ACK
SMEAN	Média do tamanho do pacote de fluxo transmitido pelo src
DMEAN	Média do tamanho do pacote de fluxo transmitido pelo dst
CT_STATE_TTL	N.º para cada state de acordo com Intervalo de valores de sttl e dttl
CT_DST_SRC_LTM	N.º de registos do mesmo srcip e do dsport em 100 registos de acordo com o fim da gravação
LABEL	Destinção entre amostra normal ou de ataque
ATTACK_CAT	Tipo de ataque

Tabela 11: Colunas da base de dados UNSW-NB15 adotadas

Realizou-se a conversão *OneHot* para dados binários das colunas categóricas da base de dados.

Tinha-se o objetivo de obter o mesmo número de amostras de ataque e de amostras normais, sendo que a base de dados UNSW-NB15 apresenta um número muito mais elevado de amostras de ataque, mais concretamente 56000 amostras normais e 119341 amostras de ataque.

Os dados foram separados entre amostras normais e amostras de ataque de forma a

equilibrar os dados de treino. Numa primeira fase foram baralhados e de seguida replicadas as amostras normais usando a técnica SMOTE fazendo com que os dados de treino passassem de 32,88% de amostras normais e 67,12% de amostras de ataque para 50% de amostras normais e 50% de amostras de ataque. Obteve-se assim nesta fase do processamento, após a remoção das amostras de ataques minoritários o mesmo número de amostras normais e de ataque, 114332, como ilustra a Figura 3: Distribuição de classes no conjunto de treino multi-classe com redução de ataques. Os dados de teste não foram equilibrados ou replicados, de forma a manter as amostras originais e assim equiparar os resultados obtidos com outros modelos da literatura existente, como ilustra a Figura 4.

Figura 3: Distribuição de classes no conjunto de treino multi-classe com redução de ataques

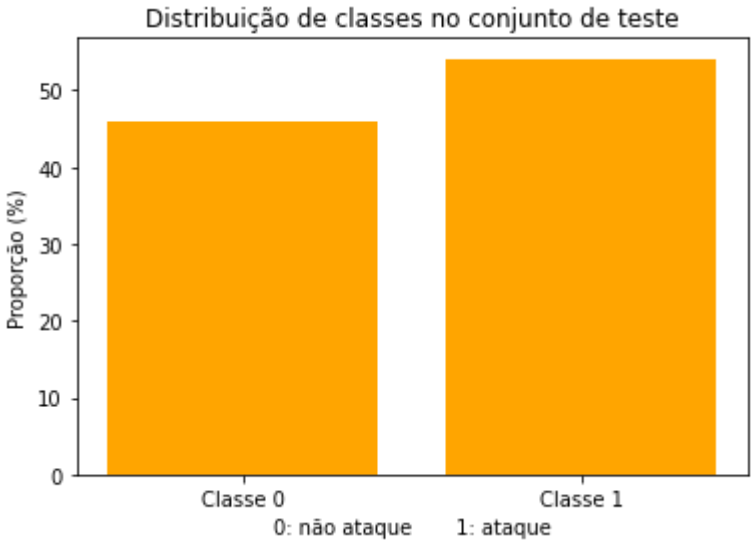

Figura 4: Distribuição de classes no conjunto de teste multi-classe com redução de ataques

Para os dados de validação o conjunto de teste foi dividido ao meio. É de referir que a base

de dados UNSW-NB15, ao contrário de outros conjuntos de dados não apresenta amostras redundantes (Moustafa & Slay, 2016).

Após a verificação das correlações *pearson* entre as caraterísticas apresentadas na base de dados procedeu-se à remoção de 2 colunas com uma correlação superior a 95%. As colunas removidas foram 'dbytes', 'dloss'. Este processo permite evitar que os dados estejam sobre correlacionados, o que pode afetar a precisão e a interpretação dos modelos de aprendizagem máquina. Removeu-se também a coluna 'label' por não ser relevante para a classificação do tráfego de rede em questão. A Figura 5 ilustra graficamente a distribuição das correlações *pearson* entre as diversas colunas para a classificação multi-classe com redução dos ataques.

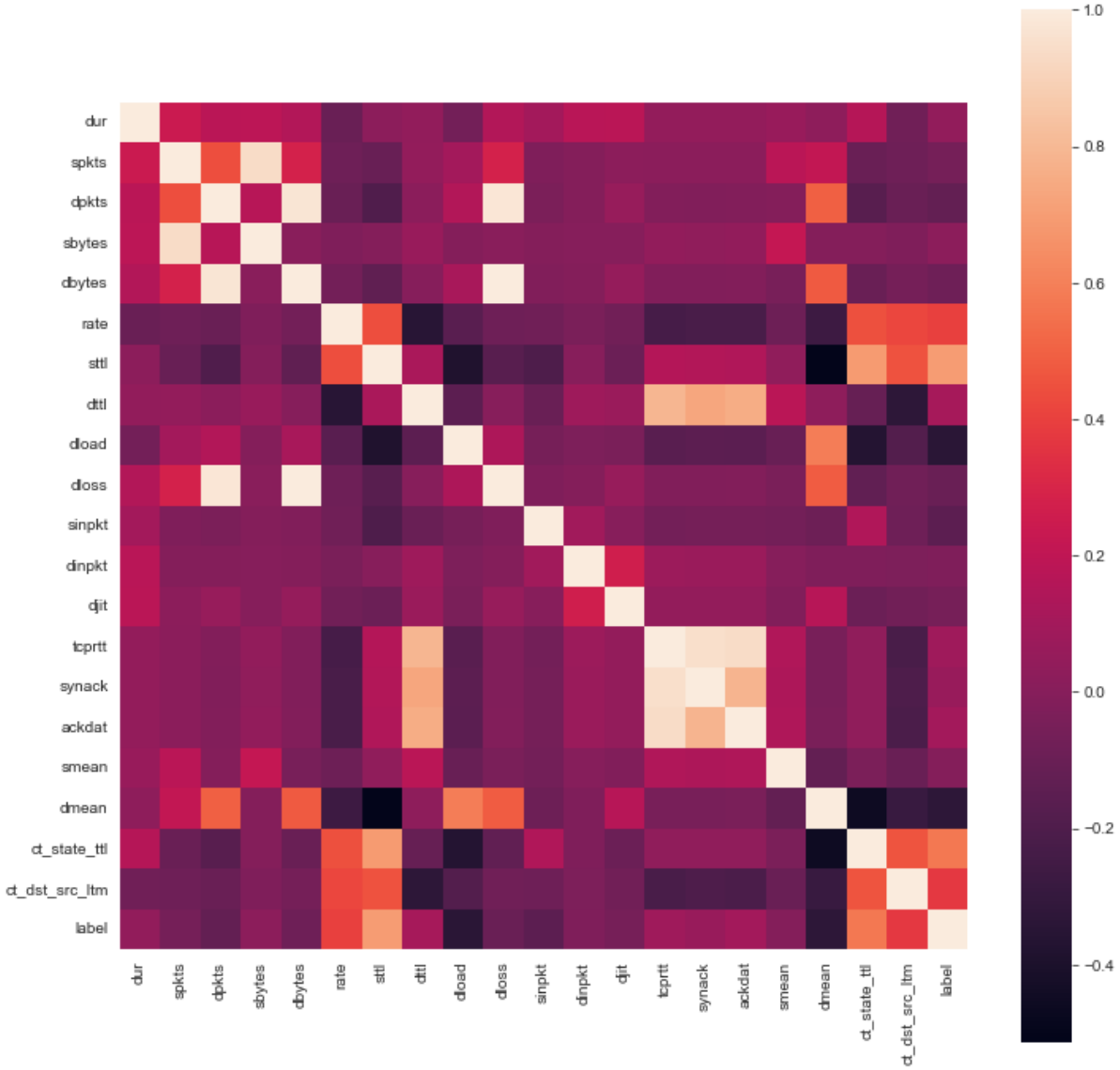

Figura 5: Distribuição das correlações *pearson* entre as diversas colunas para a classificação multi-classe com redução dos ataques

Nas colunas numéricas contínuas em que a transformação logarítmica com adição de 1

('log1p') refletia uma maior correlação com a coluna alvo ('attack_cat'), procedeu-se à transformação, de forma a tornar a distribuição dos dados numa distribuição mais normal com o objetivo de tornar o modelo mais robusto e preciso. As colunas numéricas contínuas foram ainda normalizadas com Z-*Score* com o objetivo de normalizar as diferentes escalas utilizadas nas colunas numéricas contínuas.

O processamento dos dados para a classificação multi-classe com redução dos ataques minoritários termina com 228664 amostras de treino e com o mesmo número de amostras de validação e teste, 40325.

II. **Ataques originais**

O processamento dos dados para a classificação com dados originais é em tudo semelhante ao realizado anteriormente, a destacar as pequenas diferenças nas proporções entre amostras de ataque e normais que eram respetivamente 31,94% e 68.06%. Obteve-se da mesma forma que anteriormente o mesmo número de amostras normais e de ataque, 119341, número mais elevado que no processamento anterior justamente devido a não terem sido removidos ataques minoritários.

O processamento dos dados para a classificação multi-classe com os ataques originais termina com 238682 amostras de treino e com o mesmo número de amostras de validação e teste, 41166.

4.2.2. **Classificação binária**

O processamento dos dados da base de dados para a classificação binária é em tudo idêntico ao processamento da classificação multi-classe com ataques originais. Na classificação binária não é relevante remover os ataques minoritários uma vez que os modelos classificadores apenas terão de distinguir entre amostras normais e de ataque.

Na classificação binária foi removida a coluna 'attack_cat' e foi tomada como coluna alvo a coluna 'label'.

4.3. Descrição dos modelos

De forma a alcançar os melhores resultados quanto às classificações entre ataque ou não ataque e distinção do tipo de ataque foram testadas quatro arquiteturas de redes neuronais: FFNN, DNN, RNN-LSTM e LSTM-CNN. As redes neurais são compostas por camadas interconectadas de neurónios artificiais que processam e transformam dados de entrada para produzir previsões de saída. Os modelos visam aprender as relações entre as características de entrada e as variáveis-alvo através de aprendizagem supervisionada.

4.3.1. **FFNN**

O modelo com rede *feed-forward* (FFNN) segue uma arquitetura sequencial, o que significa que cada camada é conectada à próxima camada de forma *feed-forward*, isto é, a informação flui da camada de entrada através das camadas ocultas e, finalmente, para a camada de saída.

A camada de entrada é responsável por receber os dados de entrada. A dimensionalidade da camada de entrada é igual ao número de atributos nos dados de treino.

A primeira camada oculta da FFNN é uma camada densa com 70 unidades. Cada unidade nesta camada recebe entrada de todas as unidades na camada anterior e usa a função de ativação da unidade linear retificada (ReLU) (1).

$$\mathrm{ReLU}(x) = \max(0, x) \tag{1}$$

A função de ativação ReLU introduz não-linearidade ao modelo, permitindo aprender padrões e relações complexas nos dados (Lomuscio & Maganti, 2017).

A segunda camada e terceira camadas ocultas são ambas camadas densas com 60 e 30 unidades, respetivamente, e ativação ReLU.

A camada de saída da DNN é uma camada densa com um número de unidades igual ao número de atributos da variável alvo. Aplica a função de ativação *softmax* (2), que normaliza a saída numa distribuição de probabilidade para cada classe, permitindo que o modelo faça previsões.

$$\mathrm{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} x_j} \tag{2}$$

Cada unidade na camada de saída representa a probabilidade da classe correspondente, permitindo que o modelo faça previsões com base na maior probabilidade.

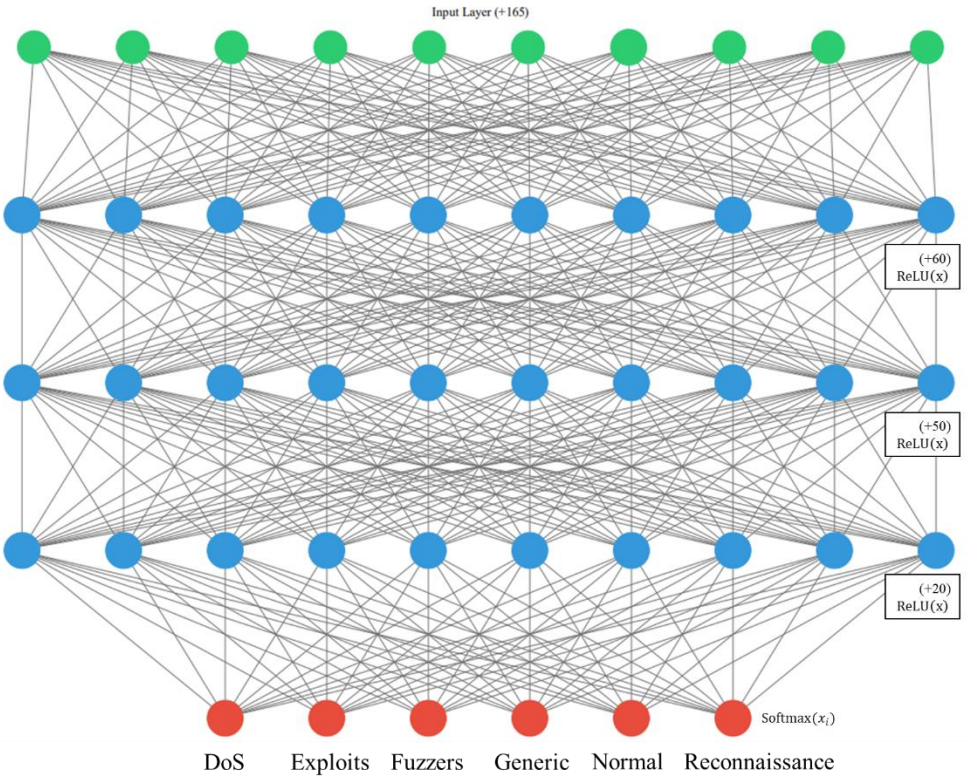

Figura 6: Esquema representativo da arquitetura FFNN adotada

No diagrama da Figura 6, cada camada é representada por um conjunto de nós azuis, com a indicação do número de unidades ou neurónios dessa camada em falta no esquema à direita. As linhas representam o fluxo de informação através da rede, desde a camada de entrada até à camada de saída. As funções de ativação (ReLU (1) e *Softmax* (2)) são especificadas ao lado das respetivas camadas.

É de ressalvar que no esquema a representação é feita para a classificação multi-classe reduzida, sendo que a camada de saída diferirá para as abordagens de classificação binária (apenas dois nós na camada de saída) e classificação multi-classe com ataques originais (nós a mais para os ataques minoritários)

4.3.2. **DNN**

O modelo com rede neuronal profunda (DNN) é em tudo semelhante ao anterior FFNN, com a adição de camadas ocultas de forma a obter uma rede mais profunda.

As primeiras três camadas possuem 70, 60 e 30 neurónios, respetivamente, todas com ativação ReLU (1), equivalente à rede FFNN anterior. De seguida apresentam seis mais camadas ocultas com 256, 384, 384, 256, 128 e 64 neurónios respetivamente.

A rede DNN termina da mesma maneira que a rede FFNN, uma camada de saída densa

com um número de unidades igual ao número de atributos da variável alvo com a aplicação da função de ativação *softmax* (2).

O diagrama da rede DNN é semelhante ao da rede FFNN representado na Figura 6, com a adição das seis camadas já referidas anteriormente, antes da camada de saída.

4.3.3. **LSTM**

O modelo com uma rede neural recorrente baseada em LSTM segue uma arquitetura muito semelhante à DNN. A arquitetura mantém-se sequencial e a camada de entrada mantém a dimensionalidade determinada pelo número de atributos dos dados de treino.

A primeira camada oculta é uma camada LSTM com 70 unidades. Cada unidade nesta camada recebe as entradas de todas as unidades na camada anterior e tem uma conexão recorrente que tem em conta a sequência de dados. Uma técnica de regularização de desistência (*dropout*) com uma taxa de 0,2 é aplicada após esta camada, que define aleatoriamente uma fração de unidades de entrada para 0 em cada atualização durante o treino, evitando o overfitting (Wager et al., 2013).

A segunda camada oculta é outra camada LSTM com 60 unidades. Como a primeira camada, também considera a sequência de dados, no entanto não tem uma camada de *dropout* associada.

A terceira camada oculta é uma camada densa com 30 unidades e ativação ReLU. Esta camada não tem uma conexão recorrente porque é aplicada após as camadas LSTM. Tem como objetivo aprender representações mais abstratas e complexas de dados.

A camada de saída é em tudo semelhante à apresentada na DNN.

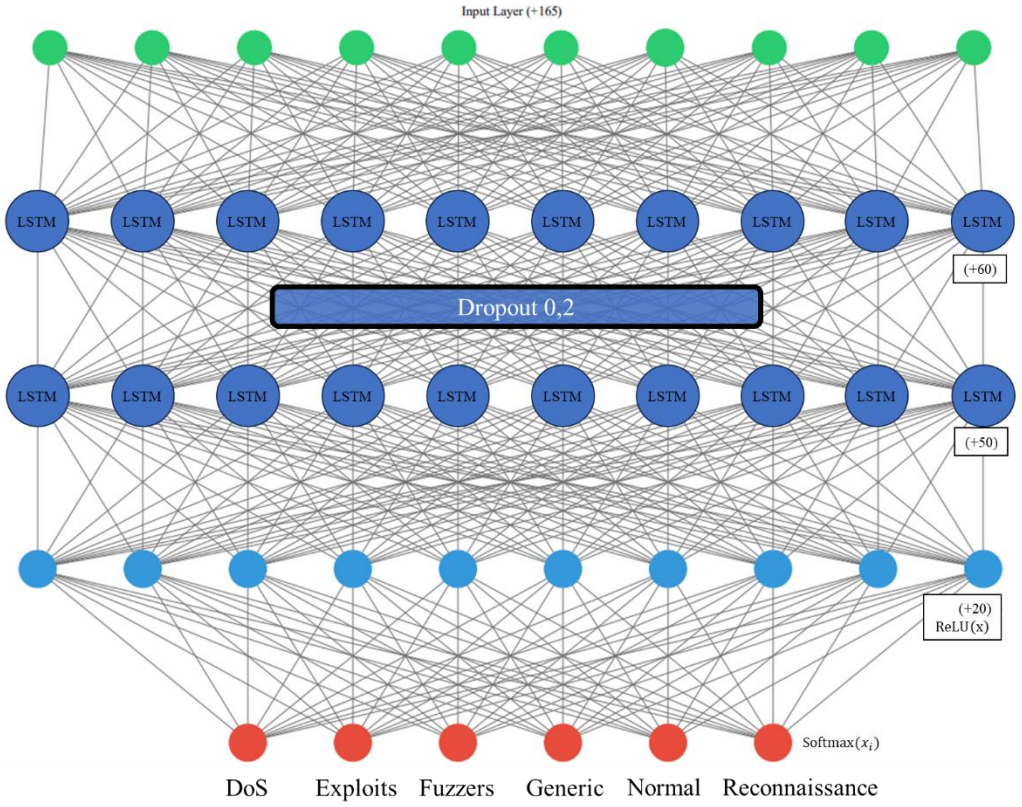

Figura 7: Esquema representativo da arquitetura RNN adotada

No diagrama da Figura 7, cada camada é representada por um conjunto de nós azuis, com a indicação do número de unidades ou neurónios dessa camada em falta no esquema à direita.

Os nós com inscrição “LSTM” representam nós *Long short-term memory*.

As linhas representam o fluxo de informação através da rede, desde a camada de entrada até à camada de saída. As funções de ativação (ReLU (1)e *Softmax* (2)) são especificadas ao lado das respetivas camadas. A camada de desistência é mostrada como Dropout (taxa).

É de ressalvar que no esquema a representação é feita para a classificação multi-classe reduzida, sendo que a camada de saída diferirá para as abordagens de classificação binária (apenas dois nós na camada de saída) e classificação multi-classe com ataques originais (nós a mais para os ataques minoritários)

4.3.4. **CNN-LSTM**

O modelo CNN-LSTM é uma adaptação da arquitetura usada por (Akhtar & Feng, 2022), consistindo numa combinação de camadas convolucionais e recorrentes. Semelhante aos modelos anteriormente descritos, a arquitetura é sequencial e a dimensionalidade da camada de entrada é determinada pelo número de atributos dos dados de treino.

A camada de entrada é composta por uma camada convolucional com 32 filtros, cada um

dos quais aplica uma operação de convolução sobre os dados de entrada. A camada de entrada mantém a função de ativação ReLU e a dimensão determinada pela forma dos dados de treino.

Após a camada convolucional, uma camada de pool (*MaxPooling*) com um tamanho de pool de 2 é adicionada. Esta camada reduz a dimensionalidade dos dados, mantendo as características mais relevantes.

Em seguida, uma camada recorrente de memória de longo prazo (LSTM) com 128 unidades é adicionada. Esta camada é configurada de forma que a camada LSTM não devolve sequências completas, mas apenas o último estado oculto.

Por fim, a camada de saída é uma camada densa (totalmente conectada) com um número de unidades igual ao número de classes da variável de destino. Usa a função de ativação *sigmoid* (3), que produz uma saída entre 0 e 1 para cada classe, representando a probabilidade de a amostra pertencer a cada classe.

$$sigmoid\ (\mathrm{x}) = \frac{1}{1 + \mathrm{e}^{-\mathrm{x}}} \tag{3}$$

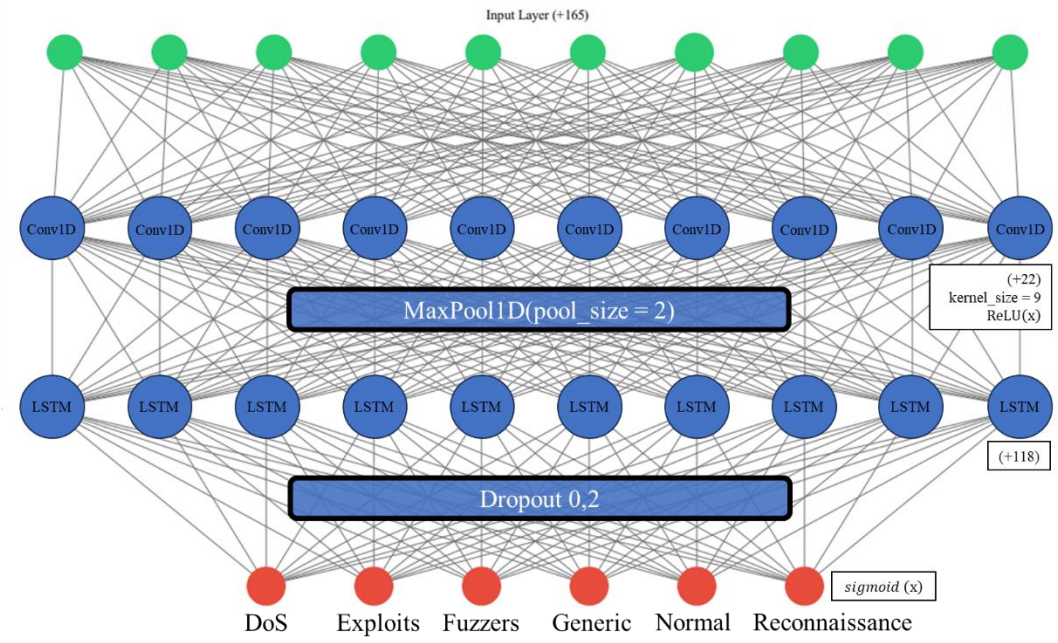

Figura 8: Esquema representativo da arquitetura CNN-LSTM adotada

No diagrama da Figura 8, cada camada é representada por um conjunto de nós azuis, com a indicação do número de unidades ou neurónios dessa camada em falta no esquema à direita.

Os nós com inscrição "Conv1D" representam filtros convolucionais, bem como os nós com a inscrição "LSTM" representam nós *Long short-term memory*.

As linhas representam o fluxo de informação através da rede, desde a camada de entrada

até à camada de saída. As funções de ativação (ReLU (1) e *Sigmoid* (3)) são especificadas ao lado das respetivas camadas. A camada de desistência é mostrada como Dropout (taxa).

É de ressalvar que no esquema a representação é feita para a classificação multi-classe reduzida, sendo que a camada de saída diferirá para as abordagens de classificação binária (apenas dois nós na camada de saída) e classificação multi-classe com ataques originais (nós a mais para os ataques minoritários)

4.4. Treino dos modelos

A base de dados utilizada para treino e teste dos classificadores foi a UNSW-NB15 já referida anteriormente.

Durante o treino, os pesos e os desvios dos modelos são ajustados usando o algoritmo de retropropagação, que minimiza a função de perda (*loss*) atualizando os parâmetros do modelo. A função de perda entropia cruzada categórica para classificação multiclasse, mesmo na classificação binária e é a usada por ter obtido melhores resultados que a entropia cruzada binária. Todos os classificadores são otimizados com o otimizador *adam*.

Os modelos são ajustados aos dados de treino por um número máximo de 100 *epochs*, estando aplicada uma paragem antecipada (*EarlyStoping*), caso a função *loss* não melhore nas últimas 30 *epochs*. O treino é realizado em lotes de tamanho 256 (*batch-size*). Os dados de validação são usados para avaliar o desempenho do modelo durante o treino e auxiliar na paragem antecipada. Após a paragem, os melhores pesos do modelo são restaurados, dando fim ao treino.

4.5. Métodos de avaliação da solução

De forma a realizar a avaliação dos diferentes modelos classificadores são utilizadas métricas calculadas com os termos TP, TN, FP, FN, como mencionado na secção 1.4 Métodos de avaliação da solução. Segundo a revisão da literatura realizada, as métricas mais favoráveis são F1-*score* (5), Exatidão (4), Precisão (6), *False Positive Rate* (FPR)/*False Alarm Rate* (7) e *Recall*/*True Positive Rate* (TPR) (8).

$$Exatidão = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$F_{1_{score}} = 2 \times \frac{precisão \times recall}{precisão + recall} \quad (5)$$

$$Precisão = \frac{TP}{TP + FP} \quad (6)$$

$$FPR = \frac{FP}{FP + TN} \quad (7)$$

$$TPR = \frac{TP}{TP + FN} \quad (8)$$

No presente Capítulo 4: Classificador baseado em redes neuronais entendeu-se como pode ser construído o classificador baseado em redes neuronais a partir de uma base de dados. Verificou-se que a construção de um classificador pode ser feita por meio de diferentes arquiteturas, cada uma com as suas próprias características, acompanhadas por uma base de dados utilizada para o treino e teste. Neste estudo, optou-se por empregar a base de dados UNSW-NB15, como já referido anteriormente, juntamente com as arquiteturas FFNN, DNN, LSTM e CNN-LSTM. Estas arquiteturas foram configuradas com os mesmos parâmetros de treino, tais como o otimizador *adam*, um limite máximo de 100 *epochs*, com uma paragem antecipada caso a função de perda (*loss*) não apresente melhoria nas últimas 30 épocas, e um tamanho de *batch-size* de 256.

Foram desenvolvidos classificadores tanto para cenários binários como para multi-classe, empregando tanto os ataques originais quanto uma versão reduzida da base de dados que excluiu os ataques minoritários.

Validação dos resultados

5.1. Classificação multi-classe reduzida

5.2. Classificação multi-classe – ataques originais

5.3. Classificação binária

Capítulo 5: Validação dos resultados

A validação dos resultados obtidos é uma etapa relevante no desenvolvimento de modelos classificadores de ciberataques baseados em redes neurais. Neste capítulo, serão apresentados os resultados de validação utilizados para avaliar o desempenho dos modelos desenvolvidos, considerando os diferentes cenários de classificação e diferentes arquiteturas apresentadas no Capítulo 4:.

Inicialmente, serão abordados os resultados da classificação multi-classe reduzida, onde os ataques minoritários, representando menos de 2% do conjunto de dados, foram removidos. Serão apresentados gráficos de *loss* e exatidão para os dados de teste e validação ao longo das *epochs* de treino, permitindo analisar a convergência dos modelos e a capacidade de generalização alcançada. Serão também apresentados os resultados do modelo aplicado aos dados de teste, nomeadamente a matriz de confusão e os valores das métricas selecionadas.

Em seguida, serão explorados os resultados da classificação multi-classe com todos os tipos de ataques existentes. Novamente, serão fornecidos gráficos de loss e exatidão, desta vez considerando o conjunto completo de ataques. Além disso, serão apresentadas as matrizes de confusão, permitindo uma análise mais detalhada do desempenho dos modelos em cada classe de ataque, e os valores das métricas selecionadas.

Por fim, serão discutidos os resultados da classificação binária, focando na distinção entre amostras normais e de ataque. Os gráficos de loss e exatidão serão fornecidos, juntamente com as matrizes de confusão específicas para essa tarefa de classificação e os valores das métricas selecionadas.

A avaliação do desempenho dos modelos será feito com base em métricas previamente selecionadas, já referidas na secção 4.5. Estas métricas permitirão comparar os modelos desenvolvidos neste trabalho com outros modelos existentes na literatura, fornecendo uma visão abrangente do seu desempenho e eficácia na deteção e prevenção de ciberataques.

5.1. Classificação multi-classe reduzida

Nesta secção aborda-se a classificação multi-classe reduzida, uma abordagem que visa a deteção dos principais tipos de ataques presentes na base de dados UNSW-NB15, enquanto reduz os ataques minoritários que representam menos de 2% do conjunto de dados. Através desta redução, procura-se simplificar o problema de classificação, concentrando-se nos ataques mais relevantes para a deteção e prevenção de ciberataques.

Na Figura 9 apresentam-se os gráficos de *loss* e exatidão obtidos com os dados de treino e validação para as diferentes arquiteturas descritas no Capítulo 4:.

Pela análise dos gráficos de *loss* nos dados de validação podemos observar que as arquiteturas de rede neuronal FFNN e DNN apresentam a estabilização da *loss* por volta de 0.43, sendo importante referir a tendência para a descida na rede FFNN que nas últimas *epochs* chega aos valores de 0.41. As arquiteturas RNN e CNN-LSTM, apesar de uma curva mais inconstante, apresentam de estabilização ligeiramente mais baixos de *loss* para os dados de validação, chegando aos valores de 0.39 na RNN e de 0.40 na CNN-LSTM.

Pela comparação dos gráficos *loss* de validação e de treino podemos verificar que as curvas apresentam algum distanciamento em todas as arquiteturas o que pode indicar algum *overfitting*, situação a ser melhorada.

Quando aos gráficos de exatidão observa-se a FFNN mantém uma exatidão com pouca evolução com valores por volta dos 0.81. As restantes redes apresentam curvas que se vão aproximando dos 0.82 a destacar a RNN que chega a ter valores de 0.83 em algumas das suas épocas. Pela comparação dos gráficos de exatidão dos dados de validação e dados de treino observamos que em todas as arquiteturas estes gráficos mantêm-se distantes, com exceção da FFNN, onde os dados de treino não chegaram a valores de exatidão tão altos comparativamente às outras arquiteturas.

Através da observação das matrizes de confusão obtidas na classificação multi-classe reduzida podemos verificar que diferentes redes apresentam resultados semelhantes, com algumas das arquiteturas com melhor desempenho a detetar certos tipos de ataques.

No geral observamos que os tipos de ataques que ofereceram mais dificuldade de identificação foram o DoS, *Fuzzers* e *Normal*, destacando-se os *DoS* que em todas as arquiteturas foram muito confundidos com *Exploits*. O tipo *Fuzzers* foi identificado várias vezes como *Normal* e *Exploits* e o tipo Normal foi identificado bastante como *Fuzzers*.

A referir o desempenho da rede FFNN com resultados valores ligeiramente menores nos focos de erro, no entanto com piores resultados nos ataques corretamente identificados, o que

revela valores mais “espalhados” pela matriz.

Quando às redes DNN, RNN e CNN-LSTM têm um desempenho semelhante com a DNN a destacar-se na identificação dos tipos de ataques *Generic*, e *Reconnaissance Normal*, a RNN a identificar melhor os *Fuzzers* e *Exploits* e por fim a CNN-LSTM com nenhuma identificação de ataque em destaque específico, mas sempre próxima dos melhores valores.

Observando as métricas exatidão, podemos verificar que todos os modelos apresentam um desempenho semelhante, com valores variando entre 0,8250 e 0,8358. Portanto, não há uma arquitetura claramente superior com base apenas nestes resultados, obrigando a verificação das restantes métricas.

Em termos de precisão, *recall*, F1-*score* e FPR, os modelos FFNN, DNN, RNN e CNN-LSTM apresentam desempenho semelhante para a maioria dos tipos de ataque.

A taxa de FPR é relativamente baixa para a maioria dos ataques, sugerindo que o modelo consegue evitar classificar erroneamente instâncias negativas como positivas.

A diferença mais notável é o *recall* para o tipo de ataque DoS, que é baixo em todos os modelos, indicando a dificuldade que todos os modelos têm em detetar este tipo de ataque.

Destaca-se o modelo RNN para o tipo de ataque *Generic* com uma precisão muito próxima de 1 e um TPR muito baixo, apesar de estes valores serem também bastante elevados nos outros modelos.

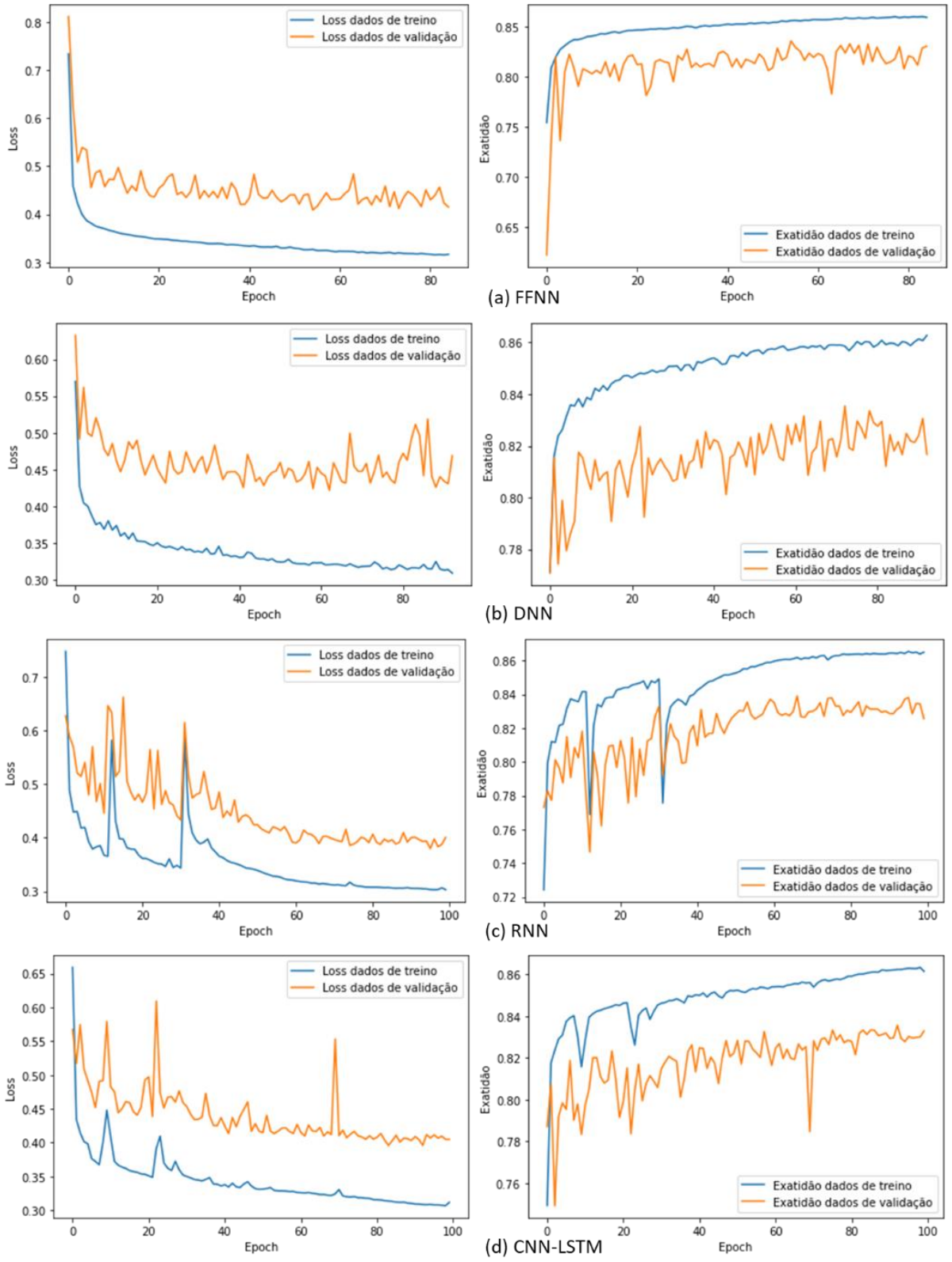

Figura 9: *Loss* e exatidão durante o treino dos classificadores multi-classe com ataques reduzidos

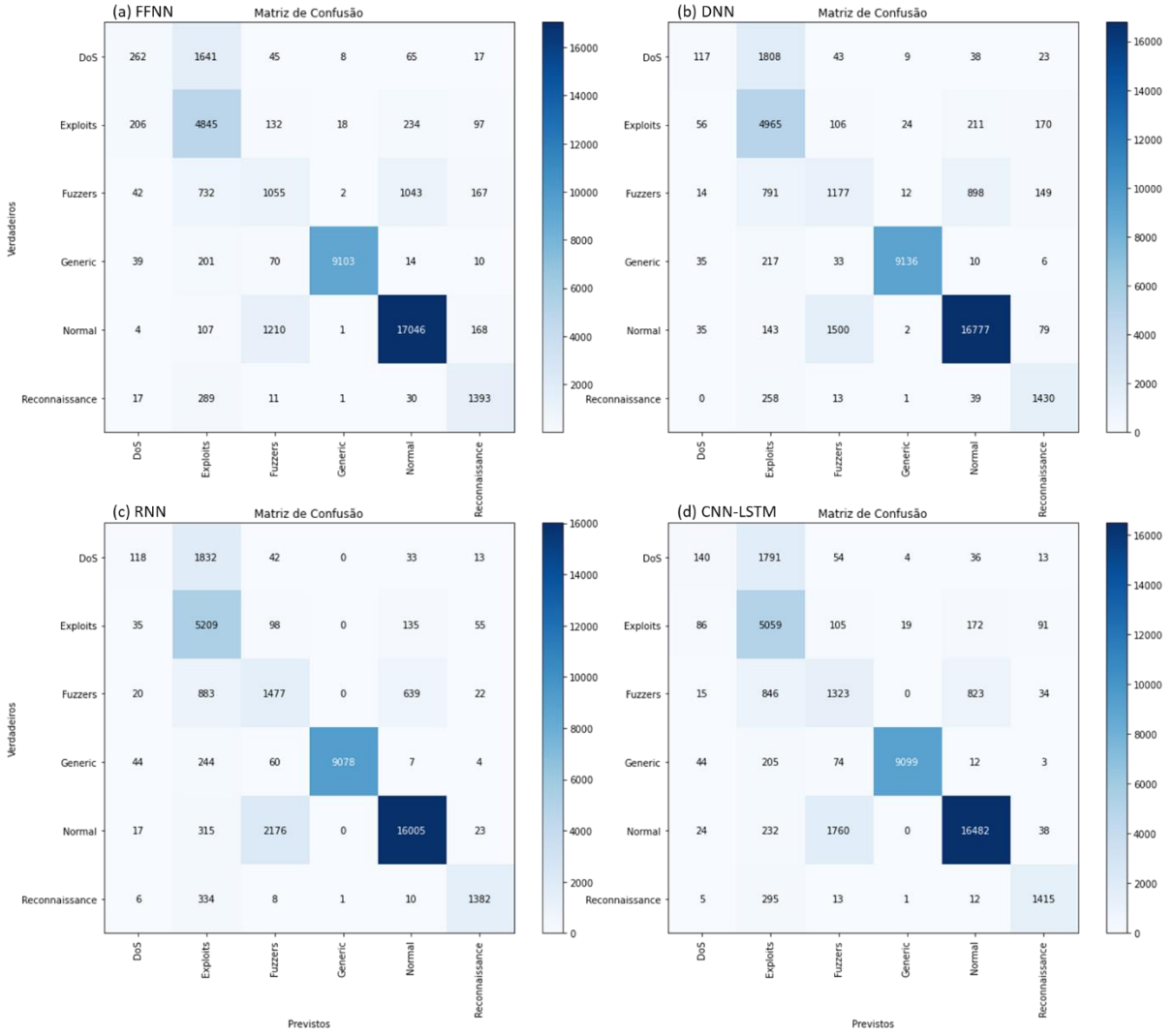

Figura 10: Matrizes de confusão das diferentes redes neuronais de classificação multi-classe com os ataques reduzidos

TIPO DE ATAQUE	PRECISÃO	RECALL/TPR	F1-SCORE	FPR	EXATIDÃO
DOS	0.4596	0.1285	0.2009	0.0080	
EXPLOITS	0.6200	0.8758	0.7260	0.0854	
FUZZERS	0.4182	0.3469	0.3792	0.0394	0.8358
GENERIC	0.9967	0.9646	0.9803	0.0010	
NORMAL	0.9248	0.9196	0.9222	0.0636	
RECONNAISSANCE	0.7522	0.8001	0.7753	0.0119	

Tabela 12: Métricas na FFNN de classificação multi-classe reduzida

TIPO DE ATAQUE	PRECISÃO	RECALL/TPR	F1-SCORE	FPR	EXATIDÃO
DOS	0.4553	0.0574	0.1020	0.0037	
EXPLOITS	0.6068	0.8975	0.7241	0.0925	
FUZZERS	0.4098	0.3870	0.3981	0.0455	0.8333
GENERIC	0.9947	0.9681	0.9813	0.0016	
NORMAL	0.9335	0.9051	0.9191	0.0549	
RECONNAISSANCE	0.7701	0.8214	0.7949	0.0111	

Tabela 13: Métricas na DNN de classificação multi-classe reduzida

TIPO DE ATAQUE	PRECISÃO	RECALL/TPR	F1-SCORE	FPR	EXATIDÃO

DOS	0.4917	0.0579	0.1036	0.0032	0.8250
EXPLOITS	0.5908	0.9416	0.7260	0.1037	
FUZZERS	0.3825	0.4857	0.4280	0.0639	
GENERIC	0.9999	0.9620	0.9806	3.238e-05	
NORMAL	0.9510	0.8635	0.9051	0.0378	
RECONNAISSANCE	0.9219	0.7938	0.8531	0.0030	

Tabela 14: Métricas na RNN de classificação multi-classe reduzida

TIPO DE ATAQUE	PRECISÃO	RECALL/TPR	F1-SCORE	FPR	EXATIDÃO
DOS	0.4459	0.0687	0.1190	0.0045	0.8312
EXPLOITS	0.6003	0.9145	0.7248	0.0968	
FUZZERS	0.3974	0.4351	0.4154	0.0538	
GENERIC	0.9974	0.9642	0.9805	0.0007	
NORMAL	0.9398	0.8892	0.9138	0.0484	
RECONNAISSANCE	0.8877	0.8128	0.8486	0.0046	

Tabela 15: Métricas na CNN-LSTM de classificação multi-classe reduzida

5.2. Classificação multi-classe – ataques originais

Nesta secção, irá ser discutida a abordagem da classificação multi-classe com ataques originais, uma estratégia que tem como objetivo identificar e classificar os diferentes tipos de ataques presentes na base de dados UNSW-NB15, sem exclusão ou redução, procurando detetar o maior número possível de tipos de ataques.

Ao analisarmos os gráficos apresentados na Figura 7, percebe-se que a FFNN teve o treino interrompido após 54 *epochs* devido à falta de melhoria na *loss* dos dados de validação em 30 épocas consecutivas. Assim, a FFNN possui um treino mais curto, mas ainda comparável às restantes arquiteturas.

Ao observarmos os gráficos de *loss*, verificamos que o comportamento das curvas é bastante semelhante à abordagem de classificação multi-classe reduzida, com a exceção de apresentarem valores ligeiramente mais elevados, como era de esperar, devido à presença de mais tipos de ataques e menos amostras para deteção nesses ataques. Destacam-se os valores de *loss* obtidos pela RNN e pela CNN-LSTM, que são ligeiramente mais baixos do que os das outras arquiteturas, chegando a 0.50.

Quanto aos gráficos de precisão, mais uma vez, as arquiteturas apresentam um comportamento semelhante, com todos os valores próximos a 0.80.

Em todas as redes ao compararmos as curvas de treino e validação para os valores de *loss* e precisão, observamos alguma distância entre elas, o que indica a presença de *overfitting* que precisa ser melhorado.

Observando as matrizes de confusão da classificação multi-classe com os ataques originais da base de dados UNSW-NB15 é evidente a influencia da presença dos ataques minoritários. Devido à fraca presença destes ataques a sua identificação é bastante reduzida em todas as arquiteturas.

As matrizes de confusão das diferentes redes neuronais revelam-se bastante semelhantes ao observado na classificação multi-classe reduzida, sendo que como era de esperar apresentam um pior desempenho na identificação na maioria dos ataques. No entanto destacam-se o aumento dos valores de amostras Normal corretamente identificadas comparativamente à classificação multi-classe reduzida, devido ao aumento de amostras de ataque.

Pela observação das tabelas podemos verificar que os valores de precisão, *recall*/TPR, F1-*score* e FPR são nulos (0) ou perto de nulos para os ataques Analysis, Backdoor e Worms. Isto indica que o modelo não conseguiu classificar corretamente estes ataques. A destacar os

modelos DNN e CNN-LSTM que conseguem obter valores ainda que baixos, significativos nos tipos de ataques Backdoor e Worms. Para os restantes ataques, os valores das métricas variam, sendo os mais altos para o ataque Generic, indicando uma melhor classificação em comparação com os outros ataques. A taxa de FPR é baixa para a maioria dos ataques, indicando uma boa capacidade dos modelos em evitar classificar erradamente uma amostra negativa como positiva.

Mais uma vez a exatidão apresenta valores bastante semelhantes, entre 0.8106 e 0.8211, nas diferentes arquiteturas, não havendo uma arquitetura claramente superior com base apenas nestes resultados.

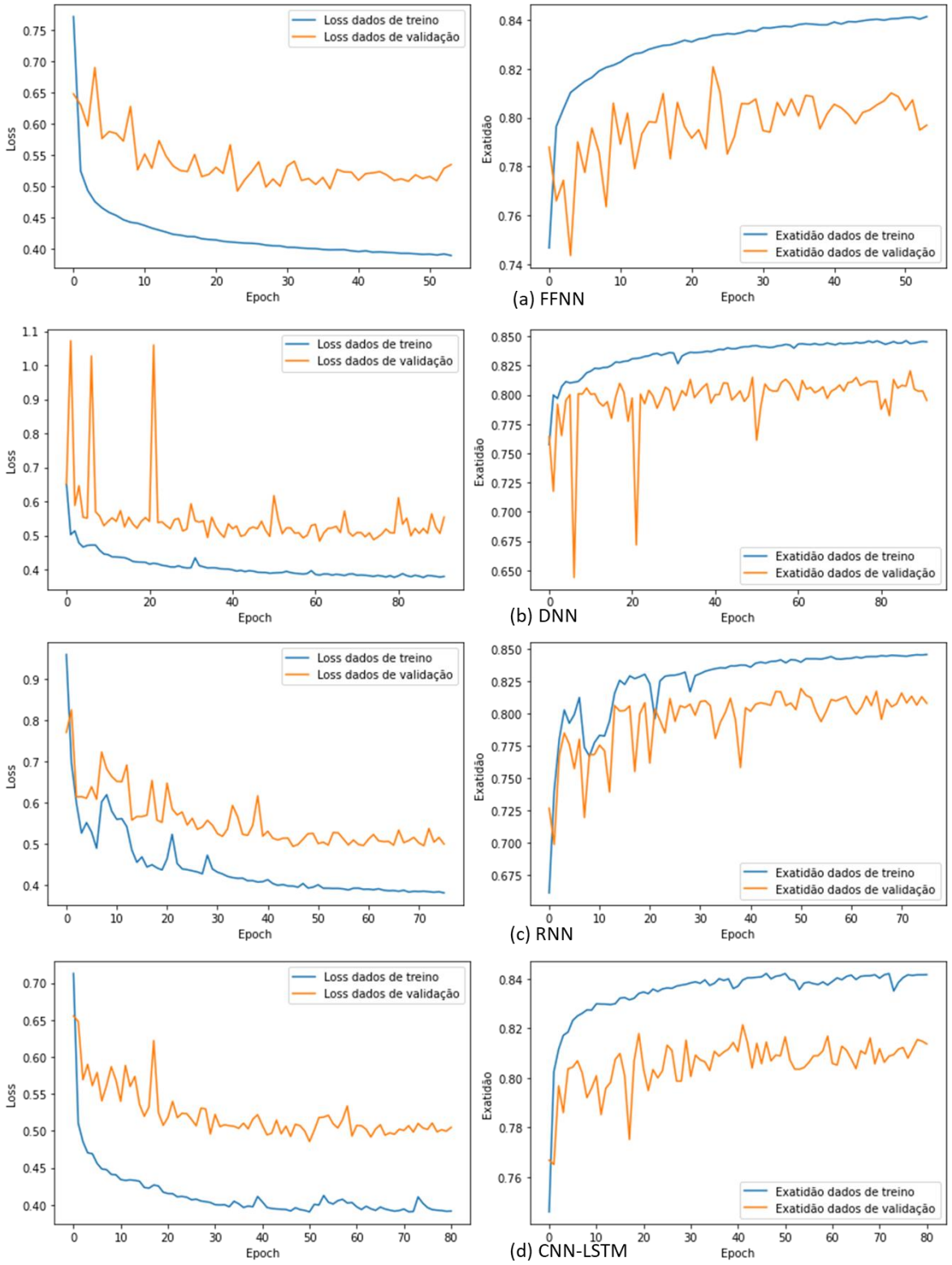

Figura 11: *Loss* e exatidão durante o treino dos classificadores multi-classe com ataques originais

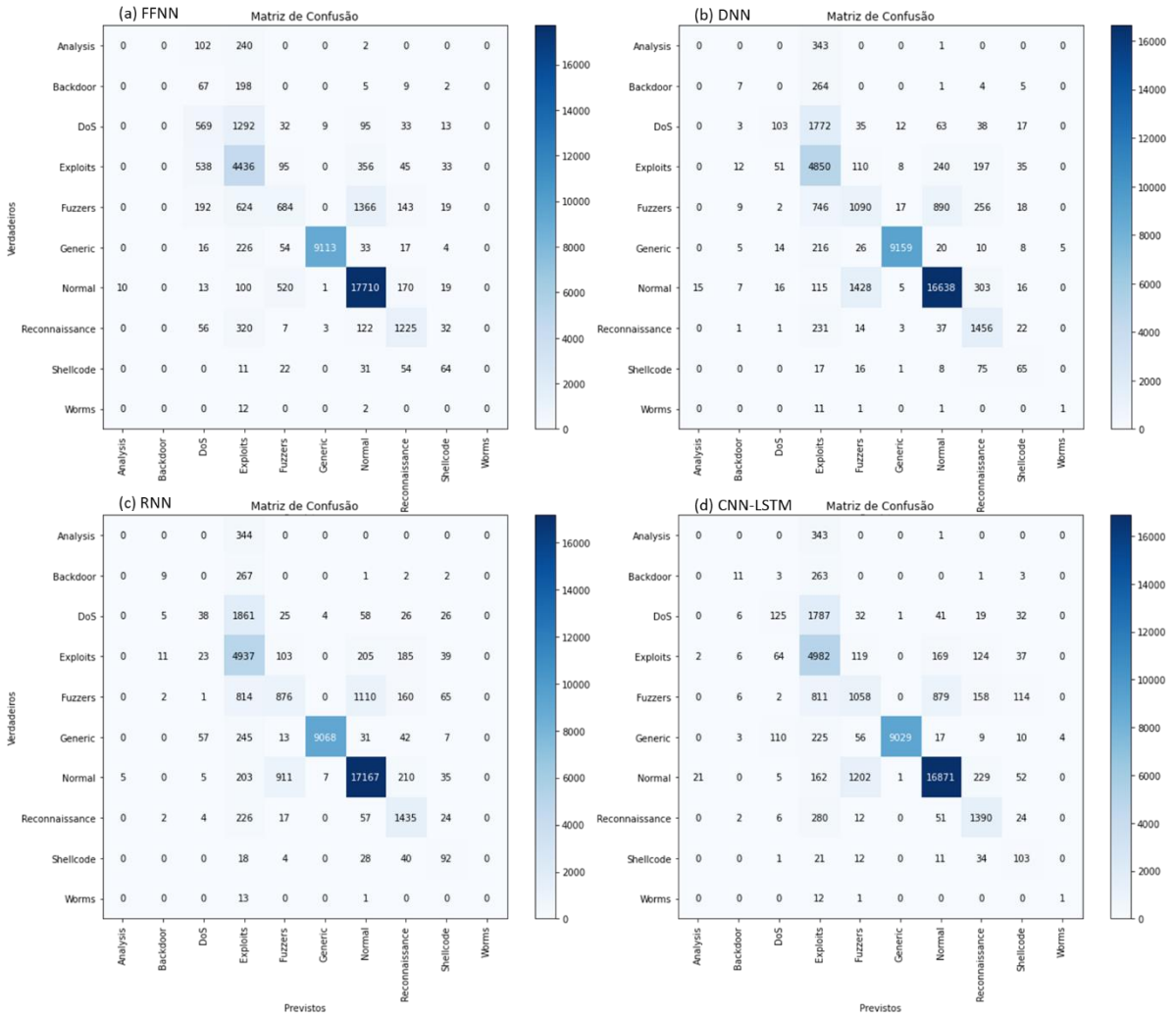

Figura 12: Matrizes de confusão das diferentes redes neuronais de classificação multi-classe com os ataques originais

TIPO DE ATAQUE	PRECISÃO	RECALL/TPR	F1-SCORE	FPR	EXATIDÃO
ANALYSIS	0	0	0	0.0002	0.8211
BACKDOOR	0	0	0	0	
DOS	0.3664	0.2785	0.3165	0.0252	
EXPLOITS	0.5947	0.8061	0.6847	0.0848	
FUZZERS	0.4837	0.2259	0.3080	0.0191	
GENERIC	0.9986	0.9630	0.9805	0.0004	
NORMAL	0.8980	0.9551	0.9257	0.0889	
RECONNAISSANCE	0.7223	0.6941	0.7079	0.0120	
SHELLCODE	0.3441	0.3516	0.3478	0.0030	
WORMS	0	0	0	0	

Tabela 16: Métricas na FFNN de classificação multi-classe – ataques originais

TIPO DE ATAQUE	PRECISÃO	RECALL/TPR	F1-SCORE	FPR	EXATIDÃO
ANALYSIS	0	0	0	0.0004	0.8106
BACKDOOR	0.1591	0.0249	0.0431	0.0009	
DOS	0.5508	0.0504	0.0924	0.0021	
EXPLOITS	0.5663	0.8813	0.6895	0.1042	
FUZZERS	0.4007	0.3600	0.3793	0.0427	
GENERIC	0.9950	0.9678	0.9813	0.0015	
NORMAL	0.9295	0.8973	0.9131	0.0557	
RECONNAISSANCE	0.6225	0.8249	0.7096	0.0224	
SHELLCODE	0.3494	0.3571	0.3533	0.0030	
WORMS	0.1667	0.0714	0.1000	0.0001	

Tabela 17: Métricas na DNN de classificação multi-classe – ataques originais

TIPO DE ATAQUE	PRECISÃO	RECALL/TPR	F1-SCORE	FPR	EXATIDÃO
ANALYSIS	0	0	0	0.0001	0.8167
BACKDOOR	0.3103	0.0320	0.0581	0.0005	
DOS	0.2969	0.0186	0.0350	0.0023	
EXPLOITS	0.5530	0.8971	0.6842	0.1119	
FUZZERS	0.4495	0.2893	0.3520	0.0281	
GENERIC	0.9988	0.9583	0.9781	0.0003	
NORMAL	0.9201	0.9258	0.9229	0.0659	
RECONNAISSANCE	0.6833	0.8130	0.7426	0.0169	
SHELLCODE	0.3172	0.5055	0.3898	0.0048	
WORMS	0	0	0	0	

Tabela 18: Métricas na RNN de classificação multi-classe – ataques originais

TIPO DE ATAQUE	PRECISÃO	RECALL/TPR	F1-SCORE	FPR	EXATIDÃO
ANALYSIS	0	0	0	0.0006	0.8155
BACKDOOR	0.3235	0.0391	0.0698	0.0006	
DOS	0.3956	0.0612	0.1060	0.0049	
EXPLOITS	0.5607	0.9053	0.6925	0.1095	
FUZZERS	0.4246	0.3494	0.3833	0.0376	
GENERIC	0.9998	0.9541	0.9764	6.309e-05	
NORMAL	0.9352	0.9098	0.9223	0.0517	
RECONNAISSANCE	0.7077	0.7875	0.7455	0.0146	
SHELLCODE	0.2747	0.5659	0.3698	0.0066	
WORMS	0.2	0.0714	0.1053	9.720e-05	

Tabela 19: Métricas na CNN-LSTM de classificação multi-classe – ataques originais

5.3. Classificação binária

Na presente secção, abordaremos a classificação binária, que tem como objetivo realizar a distinção entre amostras normais e amostras de ataque presentes na base de dados UNSW-NB15. Esta abordagem simplificada permite uma classificação mais direta e eficiente, fornecendo uma resposta clara sobre a presença ou ausência de atividades maliciosas nos dados analisados.

Com a primeira observação dos gráficos para a classificação binária da Figura 13: *Loss* e exatidão durante o treino dos classificadores binários, constata-se uma diminuição significativa nas *epochs* utilizadas para a aprendizagem entre as diferentes redes, comparativamente às classificações multi-classe anteriores. Observa-se uma esperada diminuição significativa nos valores de *loss* e aumento dos valores de exatidão, devido à diminuição das classes da variável alvo.

Comparando o comportamento durante o treino das diferentes redes, verifica-se a utilização de apenas 40 e 37 *epochs* para a FFNN e de DNN respetivamente. Entre estas duas redes verificam-se valores semelhantes de *loss*. A exatidão apresenta-se também semelhante, por volta dos valores de 0.91. Comparativamente às redes RNN e CNN-LSTM a exatidão das redes FFNN e DNN apresentou uma curva que não foi melhorando ao longo das *epochs*.

As redes RNN e CNN-LSTM obtiveram curvas que foram melhorando durante o treino com mais *epochs* que as redes anteriores, no entanto apresentando resultado bastante semelhantes.

Relativamente às matrizes de confusão da classificação binária destaca-se o desempenho da arquitetura FFNN onde obteve melhores valores para a identificação de amostras de ataque. No entanto é de referir que a FFNN obteve um valor elevado de normais erradamente identificadas como de ataque, enquanto a CNN-LSTM obteve novamente os valores.

Destaca-se novamente o desempenho do modelo CNN-LSTM com os melhores valores na identificação de amostras normais.

Os modelos DNN, RNN e CNN-LSTM apresentam valores bastante semelhantes de amostras de ataque corretamente identificadas.

Por fim destaca-se o modelo RNN com o pior resultado de amostras normais corretamente identificadas e o modelo DNN com o pior valor de ataques erradamente tomados como amostra normal.

Pela análise das tabelas das métricas nos diferentes modelos de classificação binária observamos que o desempenho dos modelos é bastante semelhante com uma exatidão entre

os 0.9202 e os 09301.

Os modelos apresentam uma alta precisão e um baixo *recall*/TPR relativo para as amostras normais, indicando que o modelo está mais inclinado a fazer previsões positivas para esta classe, no entanto perde algumas instâncias positivas no processo.

O F1-*score* para a classe normal é um pouco menor em comparação com a classe ataque, devido aos valores relativamente baixos de *recall*/TPR, indicando que os modelos têm dificuldade em equilibrar a precisão e o *recall*/TPR para a classe normal. Esta dificuldade pode ser explicada como desequilíbrio de classes presente nos dados de teste oficiais da base de daods UNSW-NB15, adotados na avaliação dos modelos. A FPR é relativamente baixa para a classe normal, indicando que os modelos têm uma boa capacidade de evitar classificar instâncias negativas (ataque) como positivas (normal).

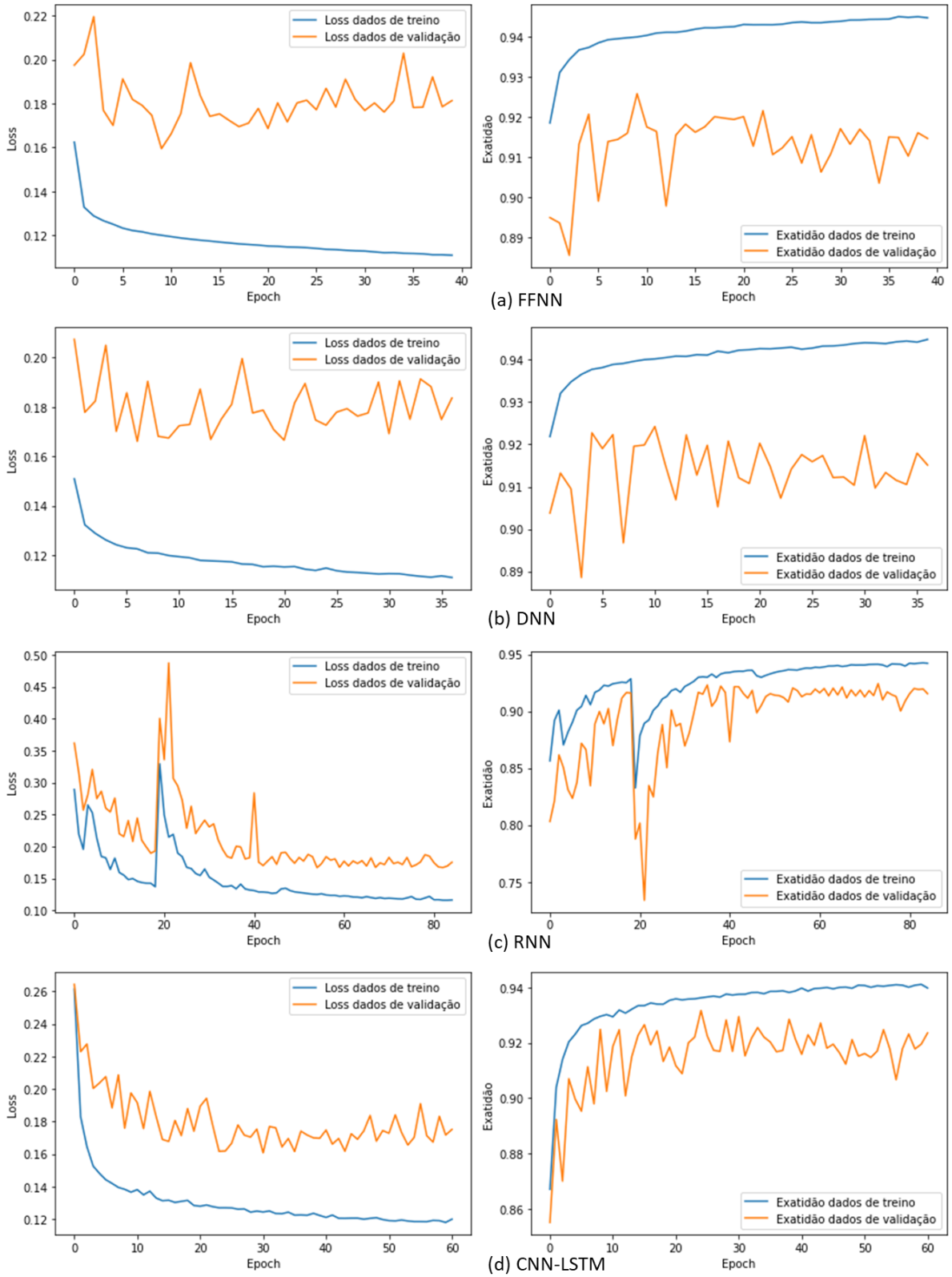

Figura 13: *Loss* e exatidão durante o treino dos classificadores binários

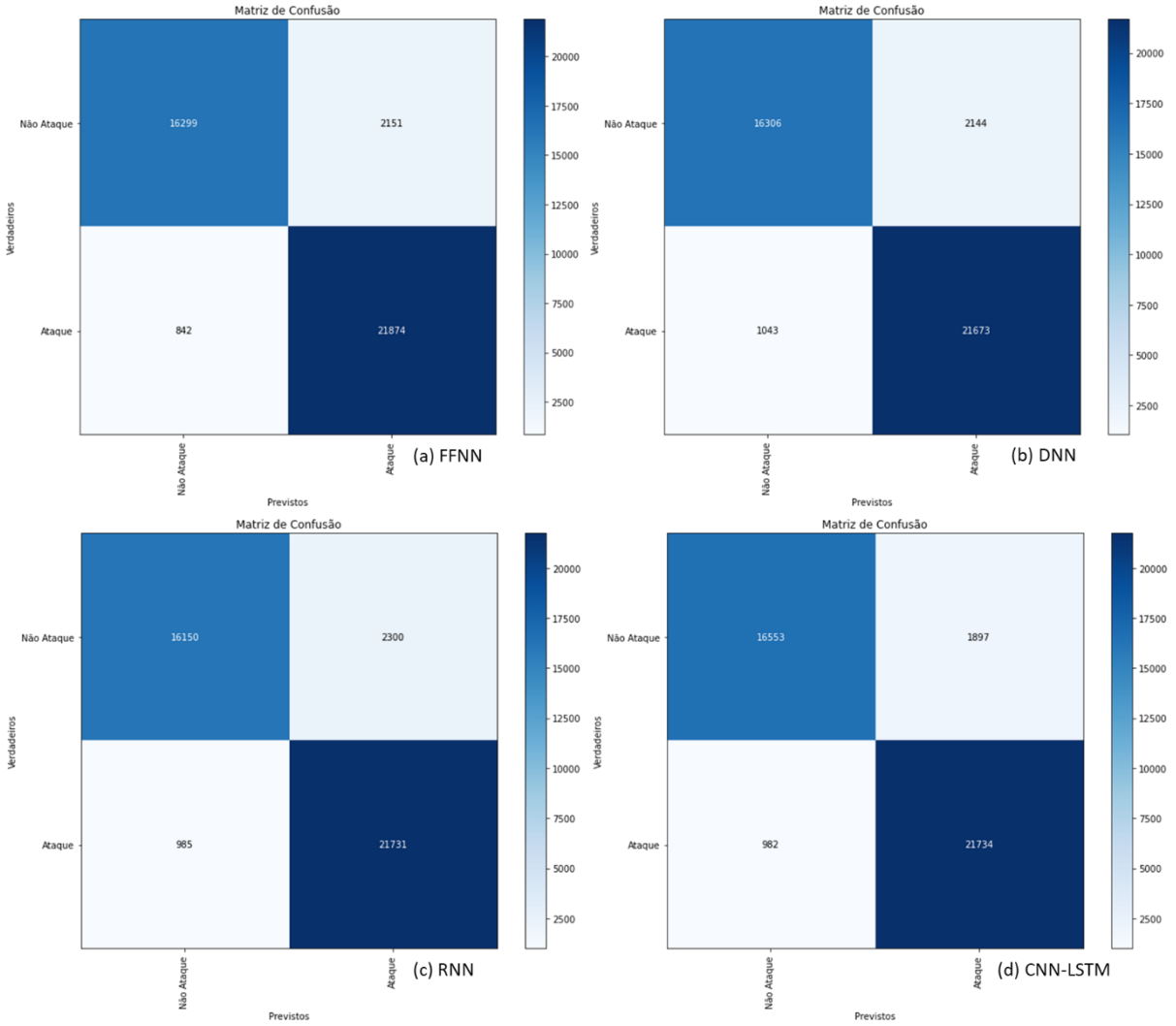

Figura 14: Matrizes de confusão das diferentes redes neuronais de classificação binária

TIPO DE ATAQUE	PRECISÃO	RECALL/TPR	F1-SCORE	FPR	EXATIDÃO
NORMAL	0.9509	0.8834	0.9159	0.0371	0.9273
ATAQUE	0.9105	0.9629	0.9360	0.1166	

Tabela 20: Métricas na FFNN de classificação binária

TIPO DE ATAQUE	PRECISÃO	RECALL/TPR	F1-SCORE	FPR	EXATIDÃO
NORMAL	0.9399	0.8838	0.9110	0.0459	0.9226
ATAQUE	0.9100	0.9541	0.9315	0.1162	

Tabela 21: Métricas na DNN de classificação binária

TIPO DE ATAQUE	PRECISÃO	RECALL/TPR	F1-SCORE	FPR	EXATIDÃO
NORMAL	0.9425	0.8753	0.9077	0.0434	0.9202
ATAQUE	0.9043	0.9566	0.9297	0.1247	

Tabela 22: Métricas na RNN de classificação binária

TIPO DE ATAQUE	PRECISÃO	RECALL/TPR	F1-SCORE	FPR	EXATIDÃO
NORMAL	0.9440	0.8972	0.9200	0.0432	0.9301
ATAQUE	0.9197	0.9568	0.9379	0.1028	

Tabela 23: Métricas na CNN-LSTM de classificação binária

Ao logo do capítulo exploraram-se os resultados de classificadores de ataques utilizando a base de dados UNSW-NB15 nas suas diferentes abordagens. Ao longo do processo, apresentaram-se e discutiram-se os resultados para três tipos de classificação: classificação multi-classe reduzida, classificação multi-classe para ataques originais e classificação binária.

Uma das principais vantagens da classificação multi-classe reduzida é a simplificação do problema, onde os ataques minoritários são removidos. Desta forma reduz-se a dimensionalidade do problema e facilita-se o treino e a interpretação dos resultados pelos modelos classificadores. No entanto, esta abordagem pode levar à perda de informações importantes, especialmente se o interesse for detetar o maior leque de ataques possível.

A classificação multi-classe com foco nos ataques originais oferece uma visão mais abrangente dos ataques. É uma abordagem útil em cenários onde a identificação do maior número de ataques é crucial. No entanto, a desvantagem é que pode aumentar a complexidade do problema, exigindo modelos mais robustos e treinos mais extensos. Estes modelos mais exigentes claramente não foram conseguidos. Devido ao desequilíbrio dos dados (menos de 2% de ataques minoritários) a deteção de ataques minoritários foi fraca. O equilíbrio de amostras entre todos os tipos de ataque seria uma solução possível para atingir valores mais satisfatórios.

A classificação binária simplifica o problema ainda mais, dividindo os dados em duas classes principais: ataque ou não ataque. Esta abordagem é benéfica quando o foco principal é distinguir atividades maliciosas de atividades normais. Tende também a ser computacionalmente mais eficiente e fácil de interpretar. No entanto, é desvantajosa se o objetivo for identificar o tipo de ataque específico da amostra em questão.

Considerando todos esses aspetos, é difícil determinar uma única "melhor" abordagem entre os modelos avaliados. Cada modelo tem os seus pontos fortes e fracos nas diferentes variantes de deteção.

A escolha do tipo de classificação depende do objetivo e das necessidades específicas do problema em questão. A classificação multi-classe reduzida é útil para uma visão geral simplificada, enquanto a classificação multi-classe - ataques originais oferece uma análise mais detalhada. Já a classificação binária é vantajosa quando o foco é apenas a identificação de atividades maliciosas. É importante considerar cuidadosamente os compromissos entre a

complexidade, a exatidão e a eficiência computacional ao selecionar a abordagem de classificação mais adequada para o cenário de aplicação.

Dentro do contexto do projeto Ciberrange da Escola Naval, a existência destas três abordagens abre o leque de escolha da capacidade de deteção do classificador que se pretende criar. Com a existência destas três abordagens permite-se até uma escala de complexidade de deteção, facilitando o desenvolvimento gradual do ciberrange.

Relativamente a outros resultados da literatura com abordagem semelhante que utilizaram técnicas de aprendizagem máquina comparáveis com as técnicas utilizadas na presente dissertação e o mesmo conjunto de teste selecionado e disponibilizado pelos autores da base de dados UNSW-NB15, apresenta-se a Tabela 24 com a comparação dos resultados da classificação multi-classe e a Tabela 25 com a comparação dos resultados da classificação binária.

REFERÊNCIA	MÉTODO	Nº. DE COLUNAS	EXATIDÃO
(CAVOJSKY ET AL., 2023)	DNN	41	0.7934
(VINAYAKUMAR ET AL., 2019)	DNN	42	0.6600
(MOUSTAFA & SLAY, 2016)	ANN	42	0.8134
(Y. YIN ET AL., 2023)	MLP (redução de ataques)	23	0.8424
PRESENTE DISSETAÇÃO	FFNN	23	0.8211
	FFNN (redução de ataques)	23	0.8358

Tabela 24: Comparação dos resultados da literatura na classificação multi-classe

REFERÊNCIA	MÉTODO	Nº. DE COLUNAS	EXATIDÃO
(KHAN ET AL., 2019)	Soft-max	10	0.8971
(WU ET AL., 2020)	CNN-RNN	42	0.8664
(VINAYAKUMAR ET AL., 2019)	RF	42	0.9030
(HANIF ET AL., 2019)	ANN	42	0.8400
(ALTUNAY & ALBAYRAK, 2023)	CNN-LSTM	42	0.9321
PRESENTE DISSETAÇÃO	CNN-LSTM	23	0.9301

Tabela 25: Comparação dos resultados da literatura na classificação binária

Ao logo do Capítulo 5: Validação dos resultados respondeu-se à questão da introdução: o que indicam, que limitações refletem e como podem ser comparados a projetos anteriores os resultados obtidos pelo classificador? Os modelos classificadores foram sujeitos a avaliações utilizando os métodos mais utilizados na literatura, incluindo uma comparação entre os resultados obtidos nesta dissertação e os estudos já existentes. Constatou-se que cada abordagem possui as suas próprias vantagens e desvantagens, dependendo do objetivo de aplicação específico. A classificação multi-classe reduzida demonstrou simplificação do problema, tornando o treino e a interpretação dos resultados pelos modelos classificadores mais acessíveis. No entanto, esta abordagem se torna menos favorável quando a deteção de uma variedade mais ampla de ataques é o foco principal.

A classificação multi-classe com foco nos ataques originais proporciona uma perspetiva mais abrangente das ameaças, mas, devido à sua maior complexidade, exige modelos mais robustos e treinos mais extensos, algo que não foi plenamente realizado nesta dissertação. Uma estratégia sugerida para melhorar os resultados seria o equilibrio das amostras entre os vários tipos de ataques.

A classificação binária, representando a forma mais simples do problema, revela-se computacionalmente mais eficiente, de fácil interpretação e produz resultados superiores. Contudo, essa abordagem é menos vantajosa quando o objetivo é identificar o tipo específico de ataque presente na amostra em questão.

Em relação à comparação com outros modelos descritos na literatura, os resultados da classificação multi-classe não apresentam contribuições significativas em relação ao que já foi alcançado em trabalhos anteriores. No entanto, a classificação binária demonstrou resultados competitivos em comparação com os estudos existentes, conseguindo ser comparável mesmo a arquiteturas mais complexas e processamentos mais detalhados.

Conclusões

Conclusões

A presente dissertação abordou a necessidade de desenvolver uma infraestrutura que permitisse aos indivíduos de uma organização aprender sobre ciberataques e como defender a organização, visando melhorar a cibersegurança organizacional. Durante o estudo, foi identificado o ciberrange como a infraestrutura mais adequada para esse propósito.

No entanto, a dissertação foi além da seleção da infraestrutura e focou-se na construção de um modelo classificador baseado em redes neurais para a deteção e análise de padrões e comportamentos cibercriminosos. Especialmente, utilizou-se o conjunto de dados UNSW-NB15 como uma base de treino e teste para o modelo. No entanto, como parte de trabalhos futuros, pretende-se melhorar os modelos utilizando dados gerados pelo ciberrange.

Para cumprir o objetivo proposto, a dissertação passou por várias etapas, mencionadas em forma de questões na introdução, cada uma contribuindo para a solução final.

Inicialmente, procurava-se saber quais as práticas utilizadas em classificadores semelhantes construídos em projetos anteriores, onde na Revisão da Literatura concluímos que existem diversas práticas usadas em projetos anteriores para a construção de classificadores. Começou-se por identificar que existem vários tipos de ataques e várias maneiras de os classificar, tendo sido adotada a classificação entre *Malware*, DoS, MITM, *Eavesdropping (Sniffing/Snooping)*, Ataque de acesso privilegiado e Ataques dia-zero. Analisou-se o desempenho das técnicas de aprendizagem máquina em projetos anteriores e conclui-se que as técnicas de aprendizagem máquina mais adequadas no contexto do projeto ciberrange da escola naval seriam a FFNN, DNN, LSTM e CNN-LSTM. Realizou-se um estudo das bases de dados *open-sorce* existentes, de entre as quais foi selecionada a base de dados UNSW-NB15 devido aos seus dados relativamente recentes, desafiantes e pré-selecionados pelos autores. Verificaram-se também quais as métricas mais usadas na literatura para avaliar os modelos classificadores criados, onde se concluiu que a F1-*score*, Exatidão, Precisão, *False Positive Rate* (FPR)/ *False Alarm Rate* e *True Positive Rate* (TPR) seriam as mais adequadas.

De seguida a intenção era perceber como identificar a melhor infraestrutura para a recolha de dados e ainda capaz de promover a preparação e proteção de uma organização relativamente à cibersegurança. Através da análise das necessidades de uma organização formar o pessoal sobre como identificar e responder a potenciais ameaças concluiu-se que o ciberrange seria a infraestrutura mais adequada para promover a preparação e proteção de

uma organização relativamente à cibersegurança e ainda ter a capacidade de recolher dados. No contexto da dissertação, foi apresentado o problema enfrentado pelas organizações, em particular a Marinha Portuguesa, enfatizando a importância contínua de aprimorar a cibersegurança organizacional. Por fim, foi ilustrado e descrito um esquema representativo da ideia do ciberrange, das suas capacidades e troca de informação interna.

A questão seguinte procurava quais os elementos-chave de uma infraestrutura que recolhe dados e apoia a aprendizagem colaborativa de preparação para a cibersegurança. Concluiu-se que um ciberrange para ser uma infraestrutura de formação dos elementos de uma organização e ainda efetuar uma recolha de dados relevante num ambiente que se reflita o mundo real, deve possuir os elementos-chave de realismo, ambiente isolado e controlado, deve ter a capacidade de simular a internet, capacidade de criar tráfego de utilizador e de rede, executar e simular ataques, simular a infraestrutura da organização, capacidade de colaboração e cooperação com outras plataformas e por fim ter ferramentas que permitam planeamento, execução, monitorização e análise da atividade no ciberrange.

Começaram a construir-se os modelos classificadores com o objetivo de responder à questão de como pode ser construído o classificador baseado em redes neuronais a partir de uma base de dados. Concluíu-se que um classificador pode ser construído com diferentes arquiteturas com diferentes parâmetros associados acompanhado com uma base de dados que sirva de treino e teste. Na presente dissertação adotou-se a base de dados UNSW-NB15 e as arquiteturas FFNN, DNN, LSTM e CNN-LSTM com os mesmos parâmetros de treino: otimizador *adam*, máximo de 100 *epochs* com paragem antecipada caso a *loss* não melhore nas últimas 30 *epochs* e *batch-size* de 256. Os cenários de classificação abrangeram tanto a classificação binária (ataque ou amostra normal) quanto a classificação multi-classe (identificação do tipo de ataque), utilizando tanto os ataques originais da base de dados quanto ataques reduzidos, de forma a abrir o leque de disponibilidade do que pode ser aplicado no projeto Ciberrange da Escola Naval em progresso.

Por fim, restava responder à questão o que indicam, que limitações refletem e como podem ser comparados a projetos anteriores os resultados obtidos pelo classificador. Os modelos classificadores foram avaliados de acordo com os métodos mais utilizados na literatura, incluindo uma comparação dos resultados obtidos na presente dissertação com os estudos existentes. Conclui-se que cada abordagem possuiu as suas vantagens e desvantagens face ao problema, mediante o interesse de aplicação. A classificação multi-classe reduzida revelou simplificação do problema, facilitando o treino e a interpretação dos resultados pelos modelos classificadores. Por outro lado, torna-se desvantajosa caso o objetivo seja detetar o

maior leque de ataques possível- A classificação multi-classe com foco nos ataques originais oferece uma visão mais abrangente dos ataques. Devido à sua maior complexidade, exige modelos mais robustos e treinos mais extensos, que não foram conseguidos na presente dissertação. Uma possível forma referida de alcançar melhores resultados seria o equilíbrio de amostras entre todos os tipos de ataque. A classificação binária representa a forma mais simples do problema sendo computacionalmente mais eficiente, fácil de interpretar e obtendo melhores resultados. No entanto, é desvantajosa se o objetivo for identificar o tipo de ataque específico da amostra em questão.

Relativamente à comparação com outros modelos construídos na literatura os resultados multi-classe não acrescentam muito ao que já foi realizado em trabalhos anteriores. No entanto, a classificação binária obteve resultados competitivos com a literatura existente, sendo até equiparável ao resultado de arquiteturas e processamentos mais complexos.

A dissertação segue a metodologia adotada de *design science research* como mencionado na secção 3.1. da Introdução e contribui de forma relevante para o campo de estudo, proporcionando informações relevantes e possíveis soluções práticas para melhorar a cibersegurança organizacional.

Assim sendo, a presente dissertação revela um estudo relevante na área, fornecendo uma abordagem inovadora e prática para fortalecer a cibersegurança nas organizações. Com o uso da infraestrutura ciberrange e a implementação dos modelos classificadores propostos, espera-se que as organizações possam aumentar sua resiliência contra ciberataques e proteger o seu ambiente digital de forma mais eficaz.

1. Trabalho futuro

Para trabalho futuro, como já foi referida, a inclusão de dados gerados pelo ciberrange no treino e teste do modelo classificador representa uma evolução significativa. Estes dados deverão conter informações sobre ataques reais e simulações de cenários de cibersegurança, o que proporcionará ao modelo uma compreensão mais abrangente e precisa dos padrões e comportamentos cibercriminosos. Além disso, o uso de dados do ciberrange permitirá ao modelo aprender com ameaças emergentes e táticas atualizadas, mantendo-se atualizado e adaptável a um ambiente em constante evolução. Estes dados refletirão cenários e comportamentos reais encontrados no ambiente digital, permitindo ao modelo aprender com situações autênticas e desenvolver respostas mais efetivas.

Ao aplicar este novo conjunto de dados ao modelo, espera-se alcançar melhores resultados em termos de deteção de ataques e classificação precisa dos tipos de ataques.

Outra vertente a ser considerada é a implementação de um ambiente de aprendizagem de cibersegurança no ciberrange. Esse ambiente forneceria aos indivíduos oportunidades práticas para adquirir habilidades e conhecimentos em cibersegurança, simulando cenários reais e fornecendo feedback imediato. Por meio de simulações interativas, exercícios práticos e desafios de segurança, os elementos da organização poderiam melhorar as suas habilidades defensivas e aprofundar sua compreensão dos métodos utilizados pelos cibercriminosos.

Referências bibliográficas

1998 DARPA Intrusion Detection Evaluation Dataset | MIT Lincoln Laboratory. (1998). https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (n.d.). *A Learning Algorithm for Boltzmann Machines**.

ADFA IDS Datasets | UNSW Research. (2013). https://research.unsw.edu.au/projects/adfa-ids-datasets

Adversarial Machine Learning (PDFDrive). (n.d.).

Akhtar, M. S., & Feng, T. (2022). Detection of Malware by Deep Learning as CNN-LSTM Machine Learning Techniques in Real Time. *Symmetry*, *14*(11). https://doi.org/10.3390/sym14112308

Alshaibi, A., Al-Ani, M., Al-Azzawi, A., Konev, A., & Shelupanov, A. (2022). The Comparison of Cybersecurity Datasets. In *Data* (Vol. 7, Issue 2). MDPI. https://doi.org/10.3390/data7020022

Al-Shareeda, M. A., Manickam, S., Laghari, S. A., & Jaisan, A. (2022). Replay-Attack Detection and Prevention Mechanism in Industry 4.0 Landscape for Secure SECS/GEM Communications. *Sustainability (Switzerland)*, *14*(23). https://doi.org/10.3390/su142315900

Altunay, H. C., & Albayrak, Z. (2023). A hybrid CNN + LSTMbased intrusion detection system for industrial IoT networks. *Engineering Science and Technology, an International Journal*, *38*. https://doi.org/10.1016/j.jestch.2022.101322

AndMal 2020 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2020). https://www.unb.ca/cic/datasets/andmal2020.html

Android Adware 2017| Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2017). https://www.unb.ca/cic/datasets/android-adware.html

Android Botnet 2015 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2015). https://www.unb.ca/cic/datasets/android-botnet.html

Android Malware 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2017). https://www.unb.ca/cic/datasets/andmal2017.html

Android Validation | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2014). https://www.unb.ca/cic/datasets/android-validation.html

Asuquo Ituen, U., & Mukeshkrishnan, M. B. (2013). Web Application Vulnerability Detection and Mitigation with Static Exploration. *International Journal of Computer Science Trends and Technology*, *3*. www.ijcstjournal.org

Bahassi, H., Edddermoug, N., Mansour, A., & Mohamed, A. (2022). Toward an exhaustive review on Machine Learning for Cybersecurity. *Procedia Computer Science*, *203*, 583–587. https://doi.org/10.1016/j.procs.2022.07.083

Bellare, M., Pointcheval, D., & Rogaway, P. (2000). Authenticated Key Exchange Secure Against Dictionary Attacks. In *Lecture Notes in Computer Science*. Springer-Verlag.

Berman, D. S., Buczak, A. L., Chavis, J. S., & Corbett, C. L. (2019). A survey of deep learning methods for cyber security. In *Information (Switzerland)* (Vol. 10, Issue 4). MDPI AG. https://doi.org/10.3390/info10040122

Bhensdadia, C. K. (2012). *Improved Decision Tree Induction Algorithm with Feature Selection, Cross Validation, Model Complexity and Reduced Error Pruning*. https://www.researchgate.net/publication/268422327

Botnet 2014 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2014). https://www.unb.ca/cic/datasets/botnet.html

Botnet and Ransomware Detection Datasets | ISOT research lab. (2010). https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/botnet-and-ransomware-detection-datasets/

Breiman, L. (2001). *Random Forests* (Vol. 45).

Burges, C. J. C. (n.d.). *A Tutorial on Support Vector Machines for Pattern Recognition*.

Caçador, F. (2022). *Devastador, ato terrorista ou um teste para algo mais perigoso? O dia em que o ataque à Vodafone deixou Portugal em alerta - Computadores - SAPO Tek*. https://tek.sapo.pt/noticias/computadores/artigos/devastador-ato-terrorista-ou-um-teste-para-algo-mais-perigoso-o-dia-em-que-o-ataque-a-vodafone-deixou-portugal-em-alerta

CAIDA Data - Completed Datasets - CAIDA. (2020). https://www.caida.org/catalog/datasets/completed-datasets/

Cavojsky, M., Bugar, G., & Levicky, D. (2023). Comparative Analysis of Feed-Forward and RNN Models for Intrusion Detection in Data Network Security with UNSW-NB15 Dataset. *2023 33rd International Conference Radioelektronika, RADIOELEKTRONIKA 2023*. https://doi.org/10.1109/RADIOELEKTRONIKA57919.2023.10109068

CIC-Bell-DNS 2021 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2021). https://www.unb.ca/cic/datasets/dns-2021.html

CIC-Evasive-PDFMal2022 | Datasets | Canadian Institute for Cybersecurity | UNB. (2022). https://www.unb.ca/cic/datasets/pdfmal-2022.html

Comodo Anti Malware Database Latest Version & Additions 2022. (2022). https://www.comodo.com/home/internet-security/updates/vdp/database.php

contagio. (2023). https://contagiodump.blogspot.com/

Contents 1. Why do we need Recurrent Neural Network? (n.d.).

correiagoncalves/classificador_UNSWNB15. (2023). https://github.com/correiagoncalves/classificador_UNSWNB15

Cortes, C., Vapnik, V., & Saitta, L. (1995). Support-Vector Networks Editor. In *Machine Leaming* (Vol. 20). Kluwer Academic Publishers.

Cunningham, P., & Delany, S. J. (2021). K-Nearest Neighbour Classifiers-A Tutorial. In *ACM Computing Surveys* (Vol. 54, Issue 6). Association for Computing Machinery. https://doi.org/10.1145/3459665

"Cyber Defense Exercise (CDX) 2009 Data" by Erik Dean, Thomas Cook et al. (2009). https://digitalcommons.usmalibrary.org/aci_datasets/1/

Damshenas, M., Dehghantanha, A., & Mahmoud, R. (n.d.). *A SURVEY ON MALWARE PROPAGATION, ANALYSIS, AND DETECTION*.

Darknet 2020 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2020). https://www.unb.ca/cic/datasets/darknet2020.html

Dasgupta, D., Akhtar, Z., & Sen, S. (2022). Machine learning in cybersecurity: a comprehensive survey. *Journal of Defense Modeling and Simulation*, *19*(1), 57–106. https://doi.org/10.1177/1548512920951275

DDoS 2019 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2019). https://www.unb.ca/cic/datasets/ddos-2019.html

Vazhakkat, S. (2022). *Difference between Threat and Attack - GeeksforGeeks*. https://www.geeksforgeeks.org/difference-between-threat-and-attack/

DoHBrw 2020 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2020). https://www.unb.ca/cic/datasets/dohbrw-2020.html

DoS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2017). https://www.unb.ca/cic/datasets/dos-dataset.html

DS2OS traffic traces | Kaggle. (2018). https://www.kaggle.com/datasets/francoisxa/ds2ostraffictraces

Duda, R. O., Hart, P. E., & Stork, D. G. (1995). *Pattern Classiication and Scene Analysis 2nd ed. Part 1: Pattern Classiication*.

Enriched Dataset | Datasets | Canadian Institute for Cybersecurity | UNB. (2021). https://www.unb.ca/cic/datasets/enricheddataset.html

FeedForward Neural Network. (n.d.).

Ferrag, M. A., Maglaras, L., Janicke, H., & Smith, R. (2019). *Deep Learning Techniques for Cyber Security Intrusion Detection : A Detailed Analysis*. https://registry.opendata.aws/cse-cic-ids2018/

Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2019). *Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study*.

Fischer, A., & Igel, C. (2012). An introduction to restricted Boltzmann machines. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7441 LNCS*, 14–36. https://doi.org/10.1007/978-3-642-33275-3_2

Fu, K., Cheng, D., Tu, Y., & Zhang, L. (2016). Credit card fraud detection using convolutional neural networks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9949 LNCS*, 483–490.

https://doi.org/10.1007/978-3-319-46675-0_53/COVER
GitHub - hgascon/adagio: Structural Analysis and Detection of Android Malware. (2013). https://github.com/hgascon/adagio
GitHub - rieck/malheur: A Tool for Automatic Analysis of Malware Behavior. (2011). https://github.com/rieck/malheur
GitHub - sontung/drebin-malwares: Malware detection using the Drebin dataset. (2014). https://github.com/sontung/drebin-malwares
Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (n.d.). *Generative Adversarial Nets*. http://www.github.com/goodfeli/adversarial
Goutam, R. K. (2015). Importance of Cyber Security. *International Journal of Computer Applications*, *111*(7), 975–8887.
Gu, Q., & Liu, P. (n.d.). *Denial of Service Attacks*.
Gümüsbas, D., Yıldırım, T., Genovese, A., Scotti, F., & Member, S. (2020). *A Comprehensive Survey of Databases and Deep Learning Methods for Cybersecurity and Intrusion Detection Systems*.
Han, J. (2015). *Data Mining: Concepts and Techniques-Chapter 6*. www.cs.uiuc.edu/~hanj
Hanif, S., Ilyas, T., & Zeeshan, M. (2019). Intrusion Detection in IoT Using Artificial Neural Networks on UNSW-15 Dataset. *HONET-ICT 2019 - IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT, IoT and AI*, 152–156. https://doi.org/10.1109/HONET.2019.8908122
Haris, M., & Sharif, U. (n.d.). *Web Attacks Analysis and Mitigation Techniques Information Security and Risk management View project Artificial Intelligence Against Cyber Attacks View project*. www.ijert.org
Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning Data Mining, Inference, and Prediction*.
Heritrix - Home Page. (2010). http://crawler.archive.org/index.html
Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). DESIGN SCIENCE IN INFORMATION SYSTEMS RESEARCH. In *Design Science in IS Research MIS Quarterly* (Vol. 28, Issue 1).
Hinton, G. (2010). *A Practical Guide to Training Restricted Boltzmann Machines*. http://learning.cs.toronto.edu
Hinton, G. E., & Osindero, S. (2006). *A Fast Learning Algorithm for Deep Belief Nets Yee-Whye Teh*.
Hinton, G. E., Osindero, S., & Teh, Y.-W. (n.d.). *A fast learning algorithm for deep belief nets **.
Hnamte, V., Nhung-Nguyen, H., Hussain, J., & Hwa-Kim, Y. (2023). A Novel Two-Stage Deep Learning Model for Network Intrusion Detection: LSTM-AE. *IEEE Access*, *11*, 37131–37148. https://doi.org/10.1109/ACCESS.2023.3266979
Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*.
Home Page - UMass Trace Repository. (n.d.). Retrieved May 20, 2023, from https://traces.cs.umass.edu/
HTTP DATASET CSIC 2010. (2010). https://www.tic.itefi.csic.es/dataset/
IBM. (2023). *IBM Security X-Force Threat Intelligence Index 2023*.
IDS 2012 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2012). https://www.unb.ca/cic/datasets/ids.html
IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2017). https://www.unb.ca/cic/datasets/ids-2017.html
IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2018). https://www.unb.ca/cic/datasets/ids-2018.html
IEEE 300-Bus System. (n.d.). Retrieved May 20, 2023, from https://electricgrids.engr.tamu.edu/electric-grid-test-cases/ieee-300-bus-system/
Insider Threat Test Dataset. (2016). https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099
Investigation on Android Malware 2019 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2019). https://www.unb.ca/cic/datasets/invesandmal2019.html
IoT Dataset 2023 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2023). https://www.unb.ca/cic/datasets/iotdataset-2023.html
Jiang, F., Fu, Y., Gupta, B. B., Liang, Y., Rho, S., Lou, F., Meng, F., & Tian, Z. (2020). Deep Learning Based Multi-Channel Intelligent Attack Detection for Data Security. *IEEE*

Transactions on Sustainable Computing, *5*(2), 204–212. https://doi.org/10.1109/TSUSC.2018.2793284

Kaggle: Your Machine Learning and Data Science Community. (2023). https://www.kaggle.com/

Karjalainen, M., & Kokkonen, T. (2020). Comprehensive Cyber Arena; the Next Generation Cyber Range. *Proceedings - 5th IEEE European Symposium on Security and Privacy Workshops, Euro S and PW 2020*, 11–16. https://doi.org/10.1109/EuroSPW51379.2020.00011

Kasongo, S. M., & Sun, Y. (2019). A deep learning method with filter based feature engineering for wireless intrusion detection system. *IEEE Access*, *7*, 38597–38607. https://doi.org/10.1109/ACCESS.2019.2905633

KDD Cup 1999 Data. (1999). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Khan, F. A., Gumaei, A., Derhab, A., & Hussain, A. (2019). TSDL: A Two-Stage Deep Learning Model for Efficient Network Intrusion Detection. *IEEE Access*, *7*, 30373–30385. https://doi.org/10.1109/ACCESS.2019.2899721

Kharon Malware Dataset. (2016). https://cidre.gitlabpages.inria.fr/malware/malware-website/dataset/index.html

Kim, J., Shin, N., Jo, S. Y., & Kim, S. H. (2017). Method of intrusion detection using deep neural network. *2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017*, 313–316. https://doi.org/10.1109/BIGCOMP.2017.7881684

Klein, M. W., Enkrich, C., Wegener, M., & Linden, S. (2006). Second-harmonic generation from magnetic metamaterials. *Science*, *313*(5786), 502–504. https://doi.org/10.1126/science.1129198

Koeune, F., & Standaert, F.-X. (n.d.). *A Tutorial on Physical Security and Side-Channel Attacks*. http://www.dice.ucl.ac.be/crypto/http://www.k2crypt.com/

Kumar, S., Pal, J., Giri, A., Raj, A., & Raj, R. (n.d.). *Intrusion Detection System using Random Forest Movie Success Prediction using Data Mining View project Crime Pattern Detection using Data Mining View project Intrusion Detection System using Random Forest*. https://www.researchgate.net/publication/332396674

LBNL Power Data | Berkeley Lab Cybersecurity R&D. (2016). https://secpriv.lbl.gov/project/powerdata/

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2323. https://doi.org/10.1109/5.726791

Li, G., Sharma, P., Pan, L., Rajasegarar, S., Karmakar, C., & Patterson, N. (2021). Deep learning algorithms for cyber security applications: A survey. In *Journal of Computer Security* (Vol. 29, Issue 5, pp. 447–471). IOS Press BV. https://doi.org/10.3233/JCS-200095

Lin, J., Dang, L., Rahouti, M., & Xiong, K. (n.d.). *ML Attack Models: Adversarial Attacks and Data Poisoning Attacks*.

Liu, S., Yang, N., Li, M., & Zhou, M. (n.d.). *A Recursive Recurrent Neural Network for Statistical Machine Translation*. Association for Computational Linguistics.

Lomuscio, A., & Maganti, L. (2017). *An approach to reachability analysis for feed-forward ReLU neural networks*. http://arxiv.org/abs/1706.07351

Lyu, Y., & Mishra, P. (2018). A Survey of Side-Channel Attacks on Caches and Countermeasures. *Journal of Hardware and Systems Security*, *2*(1), 33–50. https://doi.org/10.1007/s41635-017-0025-y

MalDroid 2020 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2020). https://www.unb.ca/cic/datasets/maldroid-2020.html

Malware Memory Analysis | Datasets | Canadian Institute for Cybersecurity | UNB. (2022). https://www.unb.ca/cic/datasets/malmem-2022.html

MAWI Working Group Traffic Archive. (2023). https://mawi.wide.ad.jp/mawi/

Mcculloch, W. S., Lrerr, W., & Pitts, H. (1943). *A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY*.

Microsoft Malware Classification Challenge (BIG 2015) | Kaggle. (2015). https://www.kaggle.com/c/malware-classification

Microsoft Security Intelligence Report VOLUME 23. (n.d.).

Mihaljević, M. J., Fossorier, M. P. C., & Imai, H. (2007). Birthday paradox based security analysis of certain broadcast encryption schemes. *IEICE Transactions on Fundamentals of Electronics,*

Communications and Computer Sciences, *E90-A*(6), 1248–1251. https://doi.org/10.1093/ietfec/e90-a.6.1248

Mijwil, M. M., Salem, I. E., & Ismaeel, M. M. (2023). The Significance of Machine Learning and Deep Learning Techniques in Cybersecurity: A Comprehensive Review. In *Iraqi Journal for Computer Science and Mathematics* (Vol. 4, Issue 1, pp. 87–101). College of Education, Al-Iraqia University. https://doi.org/10.52866/ijcsm.2023.01.01.008

ML-Based NIDS Datasets. (2022). https://staff.itee.uq.edu.au/marius/NIDS_datasets/

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (n.d.). *Playing Atari with Deep Reinforcement Learning*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Mosavi, A. (2018). *Deep Learning: A Review Visual Analytics View project Design Optimization of Electric Machines View project*. https://doi.org/10.20944/preprints201810.0218.v1

Moustafa, N., Creech, G., & Slay, J. (2017). *Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models* (pp. 127–156). https://doi.org/10.1007/978-3-319-59439-2_5

Moustafa, N., & Slay, J. (2015, December 7). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings*. https://doi.org/10.1109/MilCIS.2015.7348942

Moustafa, N., & Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal*, *25*(1–3), 18–31. https://doi.org/10.1080/19393555.2015.1125974

Moustafa, N., Slay, J., & Creech, G. (2019). Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks. *IEEE Transactions on Big Data*, *5*(4), 481–494. https://doi.org/10.1109/TBDATA.2017.2715166

Murdoch, S. J., & Danezis, G. (n.d.). *Low-Cost Traffic Analysis of Tor*. http://anon.inf.tu-dresden.de/

NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2009). https://www.unb.ca/cic/datasets/nsl.html

NumPy user guide — NumPy v1.25 Manual. (n.d.). Retrieved July 12, 2023, from https://numpy.org/doc/1.25/user/index.html#user

Park, M., Lee, H., Kim, Y., Kim, K., & Shin, D. (2022). Design and Implementation of Multi-Cyber Range for Cyber Training and Testing. *Applied Sciences (Switzerland)*, *12*(24). https://doi.org/10.3390/app122412546

Powers, D. M. W. (2007). *Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation*.

Radford, A., Metz, L., & Chintala, S. (2015). *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. http://arxiv.org/abs/1511.06434

Raymond, E. S. (2013). *How To Become A Hacker*. www.catb.org/~esr/faqs/hacker-howto.html

Raymond, J.-F. (2000). *Traffic Analysis: Protocols, Attacks, Design Issues and Open Problems*. http://www.freedom.net

Rish, I., & Rish, I. (n.d.). *An Empirical Study of the Naïve Bayes Classifier Predicting conversion to psychosis in clinical high risk patients using resting-state functional MRI features View project Clinical Machine Learning based on Cardiorespiratory models and simulation View project An empirical study of the naive Bayes classifier*. https://www.researchgate.net/publication/228845263

Rosenblatt, F. (1957). *The Perceptron, a Perceiving and Recognizing Automaton (Project Para)*.

Roumani, Y. (2021). Patching zero-day vulnerabilities: an empirical analysis. *Journal of Cybersecurity*, *7*(1). https://doi.org/10.1093/cybsec/tyab023

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning Representations by back-propagation errors. *Nature*, *323*, 533–536.

Rupprecht, D., Kohls, K., Holz, T., & Poepper, C. (2020, February 25). *IMP4GT: IMPersonation*

Attacks in 4G NeTworks. https://doi.org/10.14722/ndss.2020.24283
Saini, J. R. (n.d.). *A Comparative Analysis of Various Clustering Techniques used for Very Large Datasets*. https://www.researchgate.net/publication/281965200
Salakhutdinov, R., Mnih, A., & Hinton, G. (n.d.). *Restricted Boltzmann Machines for Collaborative Filtering*.
Sarcià, S. A., & Cantone, G. (2009). *Auto-associative Neural Networks to Improve the Accuracy of Estimation Models Software metrics View project Estimating the Number of Remaining Links in Traceability Recovery View project*. https://www.researchgate.net/publication/251811898
Sarhan, M., Layeghy, S., Moustafa, N., & Portmann, M. (2020). *NetFlow Datasets for Machine Learning-based Network Intrusion Detection Systems*. https://doi.org/10.1007/978-3-030-72802-1_9
Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). *Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization*. http://www.unb.ca/cic/datasets/IDS2017.html
Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., Chen, S., Liu, D., & Li, J. (2020). Performance comparison and current challenges of using machine learning techniques in cybersecurity. In *Energies* (Vol. 13, Issue 10). MDPI AG. https://doi.org/10.3390/en13102509
Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., & Xu, M. (2020). A Survey on Machine Learning Techniques for Cyber Security in the Last Decade. *IEEE Access*, *8*, 222310–222354. https://doi.org/10.1109/ACCESS.2020.3041951
SSH datasets - SimpleWiki. (2014). https://www.simpleweb.org/wiki/index.php/SSH_datasets
Takeda, H., Veerkamp, P., Tomiyama, T., & Yoshikawa, H. (1990). *Modeling Design Processes*.
Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., & Ghogho, M. (2016). Deep learning approach for Network Intrusion Detection in Software Defined Networking. *Proceedings - 2016 International Conference on Wireless Networks and Mobile Communications, WINCOM 2016: Green Communications and Networking*, 258–263. https://doi.org/10.1109/WINCOM.2016.7777224
Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). *A Detailed Analysis of the KDD CUP 99 Data Set*. IEEE.
TensorFlow Core. (n.d.). Retrieved July 12, 2023, from https://www.tensorflow.org/guide?hl=pt-br
The Bot-IoT Dataset | UNSW Research. (2018). https://research.unsw.edu.au/projects/bot-iot-dataset
The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background traffic. — Stratosphere IPS. (2013). https://www.stratosphereips.org/datasets-ctu13
The UNSW-NB15 Dataset | UNSW Research. (2021). https://research.unsw.edu.au/projects/unsw-nb15-dataset
The TON_IoT Datasets | UNSW Research. (2016). https://research.unsw.edu.au/projects/toniot-datasets
Tommy Morris - Industrial Control System (ICS) Cyber Attack Datasets. (2015). https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets
Tor 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2016). https://www.unb.ca/cic/datasets/tor.html
Traffic Data from Kyoto University's Honeypots. (2016). http://www.takakura.com/Kyoto_data/
UCI Machine Learning Repository: Spambase Data Set. (1999). https://archive.ics.uci.edu/ml/datasets/spambase
UGR'16 Dataset. (2016). https://nesg.ugr.es/nesg-ugr16/
Unsupervised feature extraction with Autoencoder MGI Mestrado em Gestão de Informação Master Program in Information Management. (n.d.).
Urias, V. E., Stout, W. M. S., Leeuwen, B. Van, & Lin, H. (n.d.). *Cyber Range Infrastructure Limitations and Needs of Tomorrow: A Position Paper*.
URL 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2016). https://www.unb.ca/cic/datasets/url-2016.html
USENIX Association., ACM SIGMOBILE., & ACM Digital Library. (2005). *Proceedings of the Workshop on End-to-End, Sense-and-Respons Systems, Applications, and Services : (EESR'05), June 5, 2005, Seattle, WA, USA*. USENIX Association.
User Guide — pandas 2.0.3 documentation. (n.d.). Retrieved July 12, 2023, from

https://pandas.pydata.org/docs/user_guide/index.html#user-guide
User guide and tutorial — seaborn 0.12.2 documentation. (n.d.). Retrieved July 12, 2023, from https://seaborn.pydata.org/tutorial.html
User guide: contents — scikit-learn 1.3.0 documentation. (n.d.). Retrieved July 12, 2023, from https://scikit-learn.org/stable/user_guide.html
Users guide — Matplotlib 3.7.2 documentation. (n.d.). Retrieved July 12, 2023, from https://matplotlib.org/stable/users/index.html
Vigliarolo, B. (n.d.). *MAN-IN-THE-MIDDLE ATTACKS: AN INSIDER'S GUIDE*.
Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access*, *7*, 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334
VirusShare.com. (n.d.). Retrieved May 20, 2023, from https://virusshare.com/
Vodafone Portugal. (2022). *Vodafone Portugal alvo de ciberataque - Vodafone Portugal*. https://www.vodafone.pt/press-releases/2022/2/vodafone-portugal-alvo-de-ciberataque.html
VPN 2016 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (2016). https://www.unb.ca/cic/datasets/vpn.html
Vykopal, J., Vizvary, M., Oslejsek, R., Celeda, P., & Tovarnak, D. (2017). Lessons learned from complex hands-on defence exercises in a cyber range. *Proceedings - Frontiers in Education Conference, FIE*, *2017-October*, 1–8. https://doi.org/10.1109/FIE.2017.8190713
Wager, S., Wang, S., & Liang, P. (2013). *Dropout Training as Adaptive Regularization*.
Wazid, M., Das, A. K., Chamola, V., & Park, Y. (2022). Uniting cyber security and machine learning: Advantages, challenges and future research. In *ICT Express* (Vol. 8, Issue 3, pp. 313–321). Korean Institute of Communication Sciences. https://doi.org/10.1016/j.icte.2022.04.007
WSN-DS | Kaggle. (2016). https://www.kaggle.com/datasets/bassamkasasbeh1/wsnds
Wu, P., Guo, H., & Moustafa, N. (2020). *Pelican: A Deep Residual Network for Network Intrusion Detection*. http://arxiv.org/abs/2001.08523
Yadav, R., Pathak, P., & Saraswat, S. (2020). Comparative Study of Datasets used in Cyber Security Intrusion Detection. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 302–312. https://doi.org/10.32628/cseit2063103
Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access*, *5*, 21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418
Yin, Y., Jang-Jaccard, J., Xu, W., Singh, A., Zhu, J., Sabrina, F., & Kwak, J. (2023). IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset. *Journal of Big Data*, *10*(1). https://doi.org/10.1186/s40537-023-00694-8

ANEXOS

ANEXOS

1. Anexo A – Tipos de ciberataques

Ataque *Malware*:

Um ataque de *malware* (Damshenas et al., n.d.; Wazid et al., 2022) é uma tentativa de infetar o sistema de computacional ou rede com software malicioso, chamado de *malware*.

O *malware* pode assumir várias formas, incluindo vírus, *worms*, cavalos de Troia, *ransomware*, spyware, *adware* e outros tipos de código malicioso. Podem ser realizados de diversos métodos como por e-mails de *phishing*, sites maliciosos, redes não seguras ou software ou dispositivos já infetados.

O *malware* pode afetar a vítima de várias maneiras, como por exemplo perda de dados, roubo de informações confidenciais, tempo de inatividade do sistema, perdas financeiras e danos à reputação.

Alguns exemplos de *malware* mais relevantes:

Vírus: O vírus é projetado para replicar-se e espalha-se para outros dispositivos ou sistemas. Os vírus normalmente anexam-se a ficheiros ou software legítimos e infetam-nos, fazendo com que o vírus se espalhe sempre que o ficheiro ou software infetado é aberto ou executado.

Worms: As *worms* têm o objetivo de se espalharem pelas redes de computadores explorando vulnerabilidades em sistemas operacionais, aplicativos ou protocolos de rede. Ao contrário dos vírus, as *worms* não precisam se anexar a um arquivo ou programa *host* para se propagarem, mas podem replicar-se e espalhar-se de forma autónoma.

Cavalos de Tróia: Também chamado de apenas *Trojan*, disfarça-se como um *software* legítimo para enganar os utilizadores a descarregá-lo e instalá-lo no seu computador ou dispositivo. Uma vez instalado, um *Trojan* pode executar uma variedade de ações maliciosas, como roubo de dados, danificar ficheiros, assumir o controlo do sistema ou instalar outros tipos de *malware*.

Ransomware: é um tipo de *malware* que criptografa os arquivos ou dados de uma vítima e exige pagamento em troca da chave para desbloquear os arquivos. Os ataques de *ransomware* podem ser devastadores, uma vez que podem resultar na perda de dados sensíveis ou insubstituíveis e podem causar perturbações significativas nas operações comerciais ou

atividades pessoais. Alguns tipos de *ransomware* também incluem ameaças adicionais, como ameaçar publicar dados confidenciais ou bloquear totalmente os utilizadores dos seus sistemas. Os ataques de *ransomware* podem ter como alvos indivíduos, empresas ou organizações e podem causar danos financeiros e reputacionais significativos.

***Denial-of-Service* (DoS)**:

Um ataque de negação de serviço (DoS, sigla inglesa) (Gu & Liu, n.d.; Wazid et al., 2022) é um tipo de ciberataque no qual um invasor sobrecarrega um site, rede ou serviço com solicitações ou outras formas de tráfego com o objetivo de impedir outros utilizadores de acederem ao destino atacado. O destino atacado fica incapaz de lidar com o tráfego legítimo, pois os seus recursos computacionais estão saturados, causando lentidão nos seus serviços. Esta ação pode ter repercussões graves como impedir os utilizadores legítimos de acederem a serviços críticos ou causar perdas financeiras para as empresas.

Os ataques DoS podem ser divididos em:

DoS numa Rede: a rede atacada é saturada com pacotes causando tráfego, podendo ser usados pacotes malformatados para interromper a comunicação.

DoS em aplicações: exploração de vulnerabilidades em aplicações ou serviços Web com o objetivo de causar falhas ou consumir recursos.

DoS distribuídos (DDoS): são usados vários dispositivos de forma a saturar o destino e descentralizando a fonte do ataque.

***Man-in-the-middle attack* (MiTM)**:

O ataque *Man-in-the-middle* (MITM) (Vigliarolo, n.d.; Wazid et al., 2022) é um ataque à rede no qual um invasor interceta a comunicação entre duas partes de maneira que lhe permite espiar, modificar ou inserir novas mensagens na conversa. O invasor pode chegar a este ponto representando-se como uma das partes sem conhecimento das partes envolvidas. Tronam-se ataques bastante perigosos, pois permitem que os invasores roubem várias informações confidenciais ou consigam realizar ações não autorizadas em nome da(s) vítima(s).

Ataque *Eavesdropping* (*Sniffing* ou *Snooping*):

O ataque *Eavesdropping* (Wazid et al., 2022) caracteriza-se por um ataque passivo. Consiste em ouvir secretamente uma conversa privada sem o conhecimento ou consentimento das entidades comunicantes envolvidas. Isso pode ser feito intencionalmente ou não e realizado através de canais de comunicação, como telefonemas, e-mails ou mensagens instantâneas.

Traffic analysis:

A análise de tráfego (Murdoch & Danezis, n.d.; J.-F. Raymond, 2000; Wazid et al., 2022) trata-se do processo de intercetar e analisar o tráfego de rede de forma a extrair informações sobre os padrões de comunicação e comportamentos dos utilizadores da rede, sendo assim um ataque passivo. Esta é informação pode ser posteriormente usada noutros ataques. É também utilizada para verificar se uma rede está a ser atacada ou ameaçada.

Ataque de computação de chave de sessão não autorizado:

Um ataque de computação de chave de sessão não autorizado (Bellare et al., 2000; Wazid et al., 2022) é um ataque criptográfico em que um invasor tenta calcular uma chave de sessão que é usada para comunicação segura entre duas partes, sem ter a autorização ou credenciais necessárias. A chave de sessão é uma chave temporária que é gerada durante uma sessão de comunicação segura e é usada para codificar e descodificar dados trocados entre as duas partes.

Um invasor pode iniciar um ataque de computação de chave de sessão não autorizado intercetando as mensagens trocadas entre as duas partes durante a sessão e tentando calcular a chave de sessão a partir dos dados intercetados. Se for bem-sucedido, o invasor pode usar a chave de sessão para descodificar e ler ou modificar os dados trocados durante a sessão e posteriormente fazer-se passar por uma das partes, dando origem a outros ataques.

O invasor pode chegar à chave de sessão através de outros ataques como roubo físico do dispositivo ou acesso privilegiado que irão ser falados de seguida.

Replay attack:

Um ataque *replay* (Al-Shareeda et al., 2022; Wazid et al., 2022) é um ataque à rede onde o invasor interceta e guarda pacotes de uma determinada rede com o objetivo de os retransmitir e assim tentar passar-se por um usuário legítimo ou obter acesso não autorizado a um sistema. Um exemplo deste ataque é o invasor intercetar pacotes numa comunicação como uma sessão de login entre o usuário e o servidor. O invasor ao guardar os pacotes pode mais tarde fazer-se passar pelo utilizador sem que o servidor se aperceba. Estas capacidades tornam os ataques de repetição muito perigosos, já que um invasor pode usar os dados intercetados para ignorar o processo de autenticação e obter acesso a informações ou sistemas confidenciais, por exemplo.

Ataque de representação:

Um ataque de representação (*Microsoft Security Intelligence Report VOLUME 23*, n.d.; Rupprecht et al., 2020; Wazid et al., 2022) é um ataque à rede onde o invasor deixa-se passar por um utilizador ou sistema legítimo na rede para obter acesso não autorizado a informações. Através deste ataque o invasor consegue credenciais como nomes de usuário e

palavras-passe ou utilizando engenharia social leva os outros utilizadores a fornecer informações confidenciais. A representação pode ser feita com uma falsificação de e-mail, através de phisihng ou até mesmo representação física.

Na falsificação de e-mail, por exemplo, o invasor envia um e-mail que parece ser de uma fonte legítima, como um banco ou uma empresa, na tentativa de induzir o destinatário a fornecer informações pessoais ou clicar num link malicioso.

Ataque *Scripting*:

Um ataque de *script* (Asuquo Ituen & Mukeshkrishnan, 2013; Haris & Sharif, n.d.; Wazid et al., 2022) é um tipo de ataque no qual um invasor usa scripts mal-intencionados para explorar vulnerabilidades num sistema ou aplicativo. Estes scripts podem ser escritos em diferentes linguagens de programação, como *JavaScript*, *PowerShell* ou *Python*, e podem ser executados automaticamente ou através de métodos de engenharia social para levar um utilizador a executá-los. Podem afetar bases de dados ou roubar informações confidenciais a utilizadores específicos que visitem uma página, por exemplo.

Ataque de acesso privilegiado:

Um ataque de acesso privilegiado (Haris & Sharif, n.d.; Wazid et al., 2022), tal como o nome indica, é um tipo de ataque em que um invasor obtém acesso a contas ou privilégios de usuário com níveis mais altos de permissão do que o normal. Isso significa que o invasor pode aceder, modificar ou até remover informações confidenciais ou sistemas críticos de uma organização.

Os ataques de acesso privilegiado podem ser realizados através de técnicas como engenharia social, phishing, exploração de vulnerabilidades ou roubo de credenciais. Os atacantes geralmente procuram contas com privilégios elevados, como contas de administrador, para obter acesso sem restrições aos sistemas e dados de uma organização.

Roubo físico de *smart devices*:

O roubo físico de um *smart device* (Lyu & Mishra, 2018; Wazid et al., 2022), como um smartphone ou um tablet, pode revelar-se bastante perigoso ao nível da cibersegurança, uma vez que pode conter dados sensíveis ou valiosos. Isso pode resultar numa perda de informações pessoais, dados financeiros ou informações comerciais confidenciais.

Power analysis attacks:

O dispositivo roubado pode ainda sofrer um *power analysis attack* (Koeune & Standaert, n.d.), um ataque envolve a análise do consumo de energia de um dispositivo para extrair informações confidenciais, como chaves criptograficas. Os ataques de análise de energia são frequentemente usados para contornar medidas de segurança em dispositivos inteligentes,

como protocolos de autenticação e criptografia.

Stolen verifier attack:

O dispositivo roubado pode sofrer ainda um roubo de verificador (Koeune & Standaert, n.d.), que é um tipo de ataque de autenticação em que um invasor obtém acesso a um verificador (por exemplo, uma senha com *hash* ou chave criptográfica) que é usado para autenticar um utilizador. O invasor pode usar esse verificador para se fazer passar pelo utilizador legítimo sem precisar saber a sua palavra-passe real.

Este ataque também pode ocorrer quando os verificadores são armazenados de forma insegura ou transmitidos em texto simples, permitindo que um invasor os intercete. Uma vez que o invasor tenha um verificador roubado, pode usá-lo para se passar pelo utilizador legítimo e obter acesso à sua conta ou sistema.

Ataque aniversário:

O paradoxo do aniversário (Mihaljević et al., 2007; Wazid et al., 2022) aborda o facto de num grupo de 23 pessoas, haver 50% de chance de que pelo menos duas pessoas partilhem o mesmo aniversário. A matemática por trás deste problema inspirou o ataque de aniversário, um conhecido ataque criptográfico, que usa esta estratégia probabilística para reduzir a dificuldade de quebrar uma função hash.

Um ataque de aniversário aproveita o facto de que há um número limitado de saídas de hash possíveis para uma determinada função de hash. À medida que o número de entradas hash com a mesma função aumenta, a probabilidade de duas mensagens produzirem a mesma saída de hash (ou seja, uma colisão) também aumenta. Na verdade, são necessárias apenas cerca de $2^{\frac{n}{2}}$ entradas, onde n é o comprimento da saída de hash, para ter 50% de chance de ocorrer uma colisão.

Em criptografia, uma função *hash* é uma função matemática que recebe uma entrada (ou "mensagem") e produz uma saída de tamanho fixo (ou "hash"). O objetivo de uma função *hash* é criar uma representação única e irreversível da mensagem que não pode ser facilmente revertida.

Um invasor pode usar um ataque de aniversário para encontrar uma colisão numa função de *hash*, gerando um grande número de mensagens aleatórias e calculando os seus *hashes*. Quando uma colisão é encontrada, o invasor pode usá-la para executar novos ataques, como ultrapassar mecanismos de autenticação.

Ataque dicionário:

O ataque de dicionário (Bellare et al., 2000; Wazid et al., 2022) é um tipo de ataque que

envolve chegar a uma palavra-passe ou frase secreta tentando sistematicamente todas as combinações possíveis de palavras num dicionário ou lista de palavras. O ataque baseia-se no facto de muitas pessoas usarem palavras ou frases comuns como palavras-passe, em vez de combinações aleatórias de caracteres.

No ataque, o invasor usa um programa que tenta cada palavra numa base de dados (que pode conter milhões de palavras) como possível palavra-passe. Normalmente, este programa pode também tentar variações das palavras, como adicionar números ou símbolos ao final da palavra, ou usar substituições comuns (como "1" para "i" ou "!" para "l"). O ataque continua até que a senha correta seja encontrada ou todas as combinações possíveis tenham sido testadas.

Ataques a modelos de aprendizagem máquina:

Os ataques a modelos de aprendizagem máquina (*Adversarial Machine Learning (PDFDrive)*, n.d.; Lin et al., n.d.; USENIX Association. et al., 2005; Wazid et al., 2022) caracterizam-se pela tentativa de agentes mal-intencionados de explorar vulnerabilidades na arquitetura, implementação ou dados usados por modelos de aprendizagem máquina para diminuir o seu desempenho, precisão ou segurança. Esses ataques podem ter uma série de objetivos, incluindo classificação incorreta, envenenamento de dados, roubo de modelos, evasão e injeção *backdoor*.

Alguns exemplos deste tipo de ataque são:

Ataques contraditórios: estes ataques envolvem a manipulação de dados de entrada para um modelo de aprendizagem máquina de forma que o modelo classifique incorretamente ou produza resultados incorretos. Podem ser realizados adicionando dados de entrada ou alterando dados de entrada de forma a altera o comportamento do modelo.

Ataques de envenenamento: neste ataque o invasor introduz dados mal-intencionados ou tendenciosos no conjunto de dados de treino de um modelo de aprendizagem máquina de forma a manipular o comportamento do modelo. Assim podem reduzir a precisão do modelo ou para criar um *backdoor* que permite que o invasor explore posteriormente o modelo para seu próprio proveito.

Ataques de roubo de modelo: neste ataque o invasor consulta o modelo e as suas saídas com o objetivo de, com engenharia reversa, replicar o modelo original. Com uma réplica do modelo que protege um sistema computacional, o invasor pode explorar as suas vulnerabilidades ou criar cópias não autorizadas do modelo original.

Ataques de evasão: neste ataque o invasor tenta escapar da deteção de um modelo de aprendizagem máquina criando dados de entrada que são semelhantes aos dados normais,

mas contêm alterações subtis que permitem ignorar as defesas do modelo.

Ataques dia-zero:

Um ataque de dia zero (Roumani, 2021) é um tipo de ataque que explora uma vulnerabilidade de software que não é conhecida pelo fornecedor de software ou pela comunidade de segurança. Este tipo de ataque pode ser particularmente perigoso porque pode ocorrer antes de um *patch* ou atualização de software estar disponível para resolver a vulnerabilidade. Por outras palavras, os atacantes estão a tirar partido de uma vulnerabilidade que ainda ninguém conhece, daí o termo "dia zero".

Os ataques de dia zero podem assumir muitas formas, incluindo infeções por *malware*, ataques de negação de serviço e roubo de informações.

2. Anexo B – Métodos de avaliação da solução

1) MATRIZ DE CONFUSÃO

Na matriz os dados reais são mostrados nas linhas, enquanto as previsões são mostradas nas colunas. As quatro entradas na matriz são:

- True Positive (TP): O número de casos realmente positivos que o modelo prevê corretamente como positivos.
- Falso Positivo (FP): O número de realmente negativos, mas que o modelo prevê incorretamente como positivo.
- True Negative (TN): O número de casos realmente negativos que o modelo prevê corretamente como negativos.
- Falso Negativo (FN): O número de casos realmente positivos, mas o modelo a prevê incorretamente como negativos.

Com estas entradas da matriz podem ser então calculadas diversas métricas, tais como:

2) PRECISÃO

Relaciona as amostras positivas corretamente classificadas para todas as amostras positivas classificadas no conjunto de dados (Eq. 1). Almeja-se uma precisão elevada, visto que esta é um indicativo de melhor desempenho do classificador. (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

$$\text{Precisão} = TP/(TP + FP) \quad (1)$$

3) *RECALL*/SENSIBILIDADE/TAXA DE POSITIVOS VERDADEIROS (TPR)

Trata-se de uma percentagem de amostras positivas classificadas corretamente em relação ao total de amostras positivas no conjunto de dados (Eq. 2). É preferível obter uma *Recall* maior, pois isto indica um desempenho superior do classificador. (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

$$\text{Recall} = TP/TP + FN \quad (2)$$

4) ESPECIFICIDADE/TAXA DE NEGATIVOS VERDADEIROS (TNR)

É uma proporção de amostras negativas ou maliciosas classificadas corretamente para o número total de amostras de maliciosas ou negativas no conjunto de dados (Eq. 3). Um alto nível de especificidade é desejado, pois demonstra um desempenho mais eficiente do classificador. (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

$$\text{Taxa Verdadeira Negativa} = TN/(TN + FP) \quad (3)$$

5) EXATIDÃO / *ACCURACY*

Relaciona as amostras corretamente classificadas para todas as amostras num conjunto de

dados (Eq. 4). A obtenção de uma precisão maior é vantajosa, já que isso denota um melhor desempenho do classificador. (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

$$\text{Exatidão} = (TP + TN)/(TN + FP + FN + TP) \quad (4)$$

6) TAXA DE ERRO

É a razão de amostras classificadas incorretamente para todas as amostras no conjunto de dados (Eq. 5). O desempenho do classificador é melhor quando se alcança uma taxa de erro mais baixa. (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

$$\text{Taxa de erro} = (FP + FN)/(TN + FP + FN + TP) \quad (5)$$

7) *FALL OUT/FALSE POSITIVE RATE* (FPR)

Trata-se da proporção de amostras maliciosas/negativas classificadas incorretamente para o número real total de amostras maliciosas/negativas no conjunto de dados (Eq. 6). É preferível obter uma FPR menor, pois indica um desempenho superior do classificador. (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

$$\text{Taxa de Falso Positivo} = FP/(FP + TN) \quad (6)$$

8) *MISS RATE/FALSE NEGATIVE RATE* (FNR)

É uma proporção de amostras benignas ou positivas classificadas incorretamente em relação ao número real total de amostras benignas ou positivas no conjunto de dados (Eq. 7). Um baixo nível de FNR é desejado, pois demonstra um desempenho mais eficiente do classificador. (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

$$\text{Taxa Falso Negativo} = FN/(FN + TP) \quad (7)$$

9) *FALSE DISCOVERY RATE* (FDR)

É a relação de amostras maliciosas/negativos classificadas incorretamente para o número total de amostras maliciosas/negativas classificadas no conjunto de dados (Eq. 8). A obtenção de uma FDR menor é vantajosa, já que isso denota um melhor desempenho do classificador. (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

$$\text{Taxa de descoberta falsa} = FP/(FP + TP) \quad (8)$$

10) *FALSE OMISSION RATE* (FOR)

É uma proporção de amostras benignas/positivas classificadas incorretamente em relação ao número real total de amostras benignas/positivas classificadas no conjunto de dados (Eq. 9). O desempenho do classificador é melhor quando se alcança menor valor de FOR. (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

$$\text{Taxa de Omissão Falsa} = FN/(FN + TN) \quad (9)$$

11) *NEGATIVE PREDICTED VALUE* (NPV)

É a proporção de itens corretamente classificados como não X para todos os itens

classificado como não X.

É a proporção de amostras maliciosas/negativas classificadas corretamente em relação ao número total de previsões classificadas amostras maliciosas/negativas (Berman et al., 2019).

$$NPV = TN/(TN+FN)) \quad (10)$$

11) F1-*SCORE*

Calculo que utiliza os valores de precisão e *recall* (Eq. 10). Esta medida é útil se se procurar um equilíbrio entre *recall* e precisão. É recomendável uma maior F1 *Score*, pois isso resulta num melhor desempenho do classificador. (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

$$\text{F1 Score} = 2.(\text{precisão} * \text{recall})/(\text{precisão} + \text{recall}) \quad (11)$$

12) G-*MEAN*

É calculado usando os valores verdadeiros previstos pelo classificador (Eq. 11). Caso o número de amostras negativas seja maior que o de amostras positivas, a precisão não demonstrará os resultados corretos para amostras positivas. G-Mean é útil nesta situação (Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020).

$$\text{G-média} = p\ (TP/(TP + FN)XTN/(TN + FP)) \quad (12)$$

13) CURVA DE CARACTERÍSTICAS OPERACIONAIS RECEBIDAS (ROC)

Trata-se de um gráfico que fornece um resumo de todos os desempenhos de limite, colocando no gráfico os valores de TPR (eixo y) e FPR (eixo x).(Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

14) ÁREA SOB A CURVA (AUC)

A área do gráfico ROC é denominada de AUC que varia de 0,5 a 1,0 valores. Um valor alto de AUC é desejado, pois demonstra um desempenho mais eficiente do classificador.(Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

3. Anexo C – Soluções de aprendizagem máquina

Aprendizagem profunda não supervisionada:

Este tipo de aprendizagem profunda (Li et al., 2021; Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020) é usada quando os dados de entrada não são rotulados, o que significa que o modelo não sabe a saída correta para cada entrada. O objetivo da aprendizagem não supervisionada é encontrar padrões ou estrutura nos dados, como agrupar pontos de dados semelhantes ou identificar anomalias. Alguns algoritmos comuns de aprendizagem profunda não supervisionada incluem autocodificadores, que aprendem a codificar os dados de entrada numa representação de dimensão inferior, e redes adversárias generativas (GANs), que aprendem a gerar novos dados semelhantes aos dados de entrada.

Aprendizagem profunda supervisionada:

Na aprendizagem profunda supervisionada (Li et al., 2021; Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020), os dados de entrada são rotulados, o que significa que o modelo sabe a saída correta para cada entrada. O objetivo da aprendizagem supervisionada é prever a saída correta para dados de entrada novos e invisíveis. Exemplos de algoritmos supervisionados de aprendizagem profunda incluem redes neurais convolucionais (CNNs), que são usualmente usadas para classificação de imagens, e redes neurais recorrentes (RNNs), que são frequentemente usadas para processamento de linguagem.

Aprendizagem profunda semi-supervisionada:

Este tipo de aprendizagem profunda (Li et al., 2021; Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020) combina dados rotulados e não rotulados para treinar o modelo. O objetivo da aprendizagem semi-supervisionada é usar os dados rotulados para melhorar a precisão do modelo nos dados não rotulados. A aprendizagem semi-supervisionada é frequentemente utilizada quando há poucos dados rotulados disponíveis, uma vez que pode ser difícil e caro rotular grandes conjuntos de dados. Exemplos de algoritmos de aprendizagem profunda semi-supervisionados incluem auto-treino, no qual o modelo é treinado nos dados rotulados e, em seguida, usado para rotular os dados não rotulados, e co-treino, no qual dois modelos diferentes são treinados em visões diferentes dos dados.

Aprendizagem profunda híbrida:

A aprendizagem profunda híbrida (Berman et al., 2019; Li et al., 2021; Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020) combina diferentes tipos de modelos de aprendizagem profunda para resolver uma tarefa específica. O objetivo desta solução é melhorar o desempenho do modelo para além do que pode ser alcançado usando um único tipo de

modelo de aprendizagem profunda. Um exemplo de aprendizagem profunda híbrida é o pré-treino de uma rede neural profunda usando aprendizagem não supervisionada antes de ajustá-la usando aprendizagem supervisionada. Outro exemplo é a combinação de CNNs com RNNs para tarefas como legenda de imagens, em que o modelo deve gerar uma descrição em linguagem natural de uma imagem.

4. Anexo D – Técnicas de aprendizagem máquina

Support Vector Machine (SVMs):

Support vector machines (SVMs) (Cortes et al., 1995) são uma técnica de aprendizagem supervisionada que pode ser usada para tarefas de classificação e regressão. São particularmente úteis quando se trata de conjuntos de dados multidimensionais, onde os dados podem ser muito complexos e difíceis de separar linearmente.

O objetivo do SVM é encontrar o hiperplano que melhor separa as classes do conjunto de dados. O hiperplano é definido como uma linha (ou plano) que divide o espaço em duas partes, onde cada parte corresponde a uma classe diferente. O SVM procura o hiperplano ideal, que é aquele que maximiza a distância entre as amostras de cada classe e o hiperplano. Esta distância é chamada de margem.

O SVM pode ser usado com dados não separáveis linearmente usando a função kernel. A função kernel mapeia os dados originais num espaço de maior dimensão, onde é mais provável que sejam linearmente separáveis.

Quando o SVM encontra o hiperplano ideal, o hiperplano pode ser usado para prever a classe de novos dados. Os novos dados são simplesmente mapeados para o espaço dimensional superior usando o mesmo kernel usado na fase de treino, e sua posição em relação ao hiperplano é usada para fazer a previsão.

As SVMs são uma poderosa técnica de aprendizagem máquina que pode ser usada numa variedade de tarefas de classificação e regressão. No entanto, o treino pode ser computacionalmente caro para conjuntos de dados muito grandes e complexos, sendo a escolha da função kernel e dos parâmetros de ajuste crítica para obter bons resultados. (Akhtar & Feng, 2022; Burges, n.d.; Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

Na cibersegurança esta técnica pode ser usada para classificar o tráfego de rede e detetar ciberataques. Os modelos podem ser treinados com recursos como dados de tráfego de rede ou *logs* do sistema.

Árvores de decisão:

Uma árvore de decisão (Hastie et al., 2009) é uma representação gráfica de um modelo de tomada de decisão que utiliza uma estrutura semelhante a uma árvore para representar vários resultados, consequências e decisões possíveis. É um algoritmo popular usado em aprendizagem máquina.

Numa árvore de decisão, cada nó interno representa um teste a um atributo, cada ramo representa o resultado desse teste e cada nó folha representa um rótulo de classe ou uma

decisão. A árvore é construída dividindo recursivamente o conjunto de dados em subconjuntos menores com base nos atributos mais significativos até que um critério de paragem seja atendido.

As árvores de decisão podem ser usadas para problemas de classificação e regressão. Na classificação, a árvore é construída para prever o rótulo de classe de uma determinada instância. Na regressão, a árvore é construída para prever o valor contínuo de uma variável de destino.

Uma das principais vantagens das árvores de decisão é a serem fáceis de compreender e visualizar. Têm também a capacidade de lidar com dados numéricos e categóricos, podendo também lidar com valores omissos. No entanto, podem ser sensíveis a pequenas alterações nos dados e podem facilmente sobreajustar ou não os dados de treino.(Bhensdadia, 2012; Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020; Wazid et al., 2022)

Na cibersegurança as árvores de decisão podem ser aplicadas a várias áreas como deteção de intrusão, análise de vulnerabilidade, deteção de *malware* e avaliação de riscos.

Florestas aleatórias:

As Florestas Aleatórias (Breiman, 2001) são um tipo de algoritmo de aprendizagem máquina que combina várias árvores de decisão independentes para executar tarefas de classificação e regressão. A ideia por trás das florestas aleatórias é que a combinação de várias árvores de decisão pode reduzir o efeito de *overfitting* e melhorar a precisão da previsão.

O processo de construção de uma floresta aleatória começa com a criação de várias árvores de decisão independentes. Cada árvore é construída usando uma amostra aleatória do conjunto de dados original e um subconjunto aleatório dos recursos disponíveis (variáveis). Isto reduz a correlação entre as árvores, aumentando a diversidade e reduzindo o efeito de overfitting.

Durante a fase de treino, cada árvore de decisão na floresta é construída usando um processo de divisão recursiva que seleciona o melhor recurso para dividir os dados em cada nó da árvore. A seleção do melhor recurso é baseada numa medida de impureza, como entropia ou Gini, que quantifica o quão bem um recurso divide as amostras de acordo com a classe alvo.

Depois de todas as árvores da floresta serem construídas, a classificação ou previsão é realizada usando uma votação maioritária sobre as previsões para cada árvore. Para problemas de classificação, a classe mais votada é selecionada como a classe prevista. Para problemas de regressão, a média das previsões de cada árvore é tomada como a previsão

final.

As Florestas Aleatórias têm várias vantagens sobre outros algoritmos de aprendizagem máquina. São robustos, fáceis de implementar, escaláveis e têm um bom desempenho num amplo número de tarefas. Além disso, são resistentes ao ajuste excessivo e podem lidar com conjuntos de dados multidimensionais.

No entanto, as florestas aleatórias também têm algumas desvantagens. Podem ser mais lentas do que outros algoritmos, como SVMs, em conjuntos de dados muito grandes. (Berman et al., 2019; Kumar et al., n.d.; Li et al., 2021; Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

As Florestas Aleatórias têm uma ampla gama de aplicações em cibersegurança, devido à sua capacidade de lidar com dados de alta dimensionalidade e de detetar anomalias e padrões em dados complexos. São utilizadas na deteção de *malware*, identificação de tráfego de rede anormal, deteção de intrusão, entre outros.

Naive Bayes:

Naive Bayes (Duda et al., 1995) é um algoritmo de aprendizagem supervisionado que é frequentemente usado para classificação de texto. O algoritmo é baseado no Teorema de Bayes assumindo independência condicional entre as características, tal como o nome indica "naive" (ingênuo) na tradução do inglês.

O algoritmo Naive Bayes é popular na classificação de texto, onde o objetivo é categorizar o texto numa classe específica, como por exemplo, "spam" ou "não-spam". Pode ser usado noutras tarefas de classificação, como deteção de sentimentos, categorização de notícias, deteção de spam, entre outros.

O funcionamento do Naive Bayes é baseado numa abordagem probabilística. Usa o Teorema de Bayes para calcular a probabilidade de um documento pertencer a uma classe específica, dadas as palavras que aparecem no documento. O algoritmo assume que cada palavra é independente das outras palavras e calcula a probabilidade de cada palavra aparecer a uma classe específica.

O processo de treino do Naive Bayes envolve a construção de um modelo estatístico a partir de um conjunto de dados de treino. O modelo é construído calculando a frequência de cada palavra em cada classe de documentos. De seguida, é calculada a probabilidade de cada classe de documento, dada a frequência de cada palavra em cada classe.

Durante o processo de teste, o Naive Bayes usa o modelo estatístico para classificar novos documentos de acordo com as classes existentes. Calcula a probabilidade de um documento pertencer a cada classe e seleciona a classe com a maior probabilidade.

O Naive Bayes tem várias vantagens em relação a outros algoritmos de classificação. É fácil de implementar, funciona bem com grandes conjuntos de dados e tanto o seu treino como o seu teste são rápidos. Além disso,pode lidar com muitas características diferentes e é robusto em relação a dados incompletos. (Rish & Rish, n.d.; Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

O algoritmo Naive Bayes tem várias aplicações em cibersegurança, especialmente em deteção de ameaças, ataques phishing, classificação de dados e filtragem de spam.

Clustering:

Clustering (Han, 2015) é um conjunto de técnicas de aprendizagem não supervisionada que tem como objetivo identificar grupos de objetos ou dados semelhantes dentro de um conjunto de dados. A ideia é separar os dados em grupos ou clusters, de tal forma que os objetos dentro do mesmo cluster sejam semelhantes entre si.

Existem diferentes algoritmos de *clustering*, que variam na sua abordagem para definir o número de clusters, a distância entre os objetos, e a estratégia de agrupamento. Algumas das técnicas mais comuns são:

K-means: é um algoritmo de *clustering* que divide os dados em k clusters, onde k é um número predefinido. Funciona iterativamente, escolhendo k centroides iniciais aleatórios e de seguida atribui-se objetos aos clusters de acordo com a distância euclidiana em relação ao centroide. O processo repete-se e continua até que a posição dos centroides não mude significativamente.(Saini, n.d.)

Hierárquico: é uma técnica de *clustering* que agrupa os objetos numa estrutura hierárquica de clusters. O algoritmo pode ser *bottom-up* ou *top-down*. No primeiro caso, cada objeto é inicialmente um cluster, e os clusters são combinados iterativamente com base na distância entre eles. No segundo caso, todos os objetos estão inicialmente num único cluster, que é dividido em subclusters.(Saini, n.d.)

DBSCAN: é um algoritmo de *clustering* que define clusters como regiões de alta densidade de objetos. Usa um parâmetro de distância para definir o raio em torno de cada objeto e um parâmetro de densidade para definir o número mínimo de objetos dentro do raio para considerar uma região como um cluster. Objetos que não pertencem a nenhum cluster são considerados como ruído.(Saini, n.d.)

As técnicas de *clustering* são amplamente utilizadas em várias áreas, como segmentação de mercado, deteção de anomalias, entre outras. No entanto, é importante lembrar que os resultados do *clustering* dependem da qualidade dos dados, do algoritmo escolhido e dos parâmetros de configuração, e que a interpretação dos resultados pode exigir conhecimento

especializado na área de aplicação.

Na cibersegurança pode ser usada para agrupar tráfego de rede semelhante ou *logs* do sistema. Estas capacidades podem ser uteis para detetar comportamentos anômalos ou identificar potenciais ameaças.

K-vizinhos

K-*nearest neighbors* (KNN) (Duda et al., 1995) é um algoritmo de aprendizagem máquina que é usado para tarefas de classificação e regressão.

O algoritmo KNN funciona definindo primeiro um valor para K, que representa o número de vizinhos mais próximos a considerar. De seguida, dada uma nova observação, o algoritmo identifica os K vizinhos mais próximos nos dados de treino com base numa medida de distância (por exemplo, distância euclidiana). A classe ou valor da nova observação é então determinada com base na classe ou valor médio mais comum dos K vizinhos mais próximos.

KNN pode ser um algoritmo simples e eficaz para tarefas de classificação e regressão, especialmente quando há muitos dados disponíveis. No entanto, pode ser sensível à escolha de K e à medida de distância usada, pode não ter um bom desempenho com dados multidimensionais ou com classes desequilibradas. Também pode ser computacionalmente caro para grandes conjuntos de dados. (Cunningham & Delany, 2021; Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

Na cibersegurança o KNN é útil na deteção de intrusão e classificação de *malware*.

5.4. Anexo D.1 – Técnicas de aprendizagem profunda

Rede Neural Artificial (ANNs):

As Redes Neurais Artificiais (ANNs) (Mcculloch et al., 1943) são um tipo de algoritmo de aprendizagem máquina que é inspirado na estrutura e função dos neurónios biológicos no cérebro humano. As ANNs consistem em nós interligados, ou neurónios artificiais, que podem processar e transmitir informação utilizando um conjunto de pesos que vai sendo aprendido.

Nas ANNs, os dados de entrada são alimentados na camada de entrada, que passa os dados através de uma série de camadas ocultas antes de produzir uma saída na camada de saída. Cada nó nas camadas ocultas executa um cálculo simples com base na sua entrada e um conjunto de pesos, que são atualizados durante o processo de treino usando um algoritmo de otimização, como descida de gradiente estocástico.

As ANNs têm sido aplicadas a uma ampla gama de problemas, incluindo reconhecimento de imagem e fala, processamento de linguagem natural e sistemas de recomendação. Também têm sido usados em cibersegurança para tarefas como deteção de intrusão e classificação de *malware*.

Um dos principais desafios das ANNs é o *overfitting*, onde o modelo se torna muito especializado para os dados de treino e tem um mau desempenho em dados novos e invisíveis. Para enfrentar este desafio, técnicas como regularização, paragem precoce e abandono escolar foram desenvolvidas.

De um modo geral, as ANNs provaram ser uma ferramenta poderosa e flexível para resolver problemas complexos numa variedade de domínios, incluindo a cibersegurança. No entanto, a escolha da arquitetura de rede neural apropriada e a estratégia de treino podem afetar significativamente o desempenho do modelo. (Berman et al., 2019; Mosavi, 2018; Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

Feedforward Neural Networks (FNNs):

Feedforward Neural Networks (FNNs) (Rosenblatt, 1957) são um tipo de rede neural artificial onde a informação flui numa direção, da camada de entrada através de uma ou mais camadas ocultas para a camada de saída, sem quaisquer ligações de feedback. Nas FNNs, cada camada é composta por um conjunto de nós ou neurónios, onde as conexões entre os neurónios são ponderadas.

As RNF são amplamente utilizadas em tarefas de aprendizagem supervisionada, tais como classificação e regressão. São particularmente eficazes a lidar com dados de entrada de alta

dimensão, como imagens ou sinais de som. Os FNNs também podem ser usados para tarefas de aprendizagem não supervisionadas, como extração de recursos e *clustering*.

O processo de treino de FNNs envolve ajustar os pesos das conexões entre os neurónios para minimizar uma função de custo, como erro quadrado médio ou entropia cruzada. Isto geralmente é feito usando um algoritmo de otimização, como descida de gradiente estocástico.

Um dos principais desafios das FNNs é o potencial para *overfitting*, onde o modelo se torna muito complexo e tem um mau desempenho em dados novos e invisíveis.(*FeedForward Neural Network*, n.d.)

Existem várias variantes de FNNs, incluindo Perceptrons Multicamadas (MLPs), Redes Neurais Convolucionais (CNNs) e Redes Neurais Recorrentes (RNNs). Cada variante tem uma arquitetura exclusiva e é adequada para diferentes tipos de dados e tarefas.

Redes Neurais Convolucionais (CNNs):

As Redes Neurais Convolucionais (CNNs) (LeCun et al., 1998) são um tipo de rede neural que são especificamente projetadas para processar e analisar imagens e outros dados multidimensionais. As CNNs são compostas por várias camadas, incluindo camadas convolucionais, camadas *pooling* e camadas totalmente conectadas.

A operação-chave numa camada convolucional é a operação de convolução. A operação de convolução numa CNN envolve pegar numa pequena janela ou filtro, normalmente 3x3 ou 5x5 e deslizá-lo sobre a imagem de entrada. Em cada posição do filtro, um produto de pontos é calculado entre o filtro e a parte correspondente da imagem.

Este produto de pontos produz um único valor escalar, que é então gravado no mapa de recursos de saída na mesma posição espacial do centro do filtro. Este processo é repetido para cada posição possível do filtro sobre a imagem de entrada, resultando num novo mapa de recursos que destaca a presença de padrões ou recursos específicos na imagem de entrada.

Um dos benefícios do uso de camadas convolucionais é que podem aprender a detetar padrões ou recursos locais, independentemente de sua localização na imagem. Isto torna as CNNs muito mais robustas para variações na posição, orientação e escala do objeto do que as técnicas tradicionais de visão computacional que dependem de recursos artesanais.

As camadas *pooling* são normalmente usadas para reduzir o tamanho dos mapas de recursos e aumentar a eficiência computacional da rede. Uma operação de agrupamento usa uma pequena secção do mapa de recursos e gera um único valor, normalmente o valor máximo ou médio nessa secção. Desta forma reduz a resolução espacial do mapa de recursos, mas preserva as informações mais relevantes.

As camadas totalmente conectadas no final de uma CNN pegam nos mapas de recursos nivelados e usam-nos para fazer uma previsão, como a classe de uma imagem.

As CNNs alcançaram desempenho de última geração numa vasta ampla gama de tarefas relacionadas à imagem, como classificação de imagem, deteção de objeto, entre outras. São também usadas noutros campos, como reconhecimento de fala e processamento de linguagem.

No entanto, o treino de uma CNN pode exigir uma grande quantidade de dados e recursos computacionais, sendo que a arquitetura e os parâmetros da rede precisam de ser cuidadosamente projetados e ajustados para alcançar um desempenho ideal. Além disso, interpretar o funcionamento interno de uma CNN e entender como chega às suas previsões pode ser um desafio.

No campo da cibersegurança as CNN podem ser treinadas em dados de tráfego de rede para identificar comportamentos anómalos ou detetar tipos específicos de ciberataques, como ataques DDoS ou verificações de portas. (Berman et al., 2019; Li et al., 2021; Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

Redes Neurais Recorrentes (RNNs):

Redes Neurais Recorrentes (RNNs) (Rumelhart et al., 1986) são um tipo de rede neural normalmente usado para processar dados sequenciais, como séries temporais ou processamento de linguagem natural. Ao contrário das redes neurais feedforward, que processam dados de entrada numa única passagem e não têm memória interna, as RNNs podem manter o estado durante vários *timesteps*.

Em cada *timestep*, a RNN toma um vetor de entrada e o seu estado interno do *timestep* anterior como entradas, e produz um vetor de saída e um novo estado interno como saídas. Assim permite-se que a RNN use informações de etapas anteriores para gerar a sua saída atual.

Na cibersegurança os modelos com RNN podem ser treinados em dados de log do sistema para identificar comportamentos anormais ou detetar tipos específicos de ciberataques, como injeção de SQL ou scripts entre sites.(Berman et al., 2019; *Contents 1. Why Do We Need Recurrent Neural Network?*, n.d.; Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

Redes de memória de longo prazo (LSTM):

A rede Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) é um tipo de RNN projetada para resolver o problema de gradiente de desaparecimento que pode acontecer em RNNs padrão. O problema do gradiente de desaparecimento surge quando os gradientes usados para atualizar os pesos da rede tornam-se muito pequenos, dificultando a

aprendizagem de dependências de longo prazo.

As redes LSTM usam uma estrutura interna mais complexa comparativamente às RNNs padrão, incluindo portas de entrada, esquecimento e saída que controlam o fluxo de informações através da rede. Desta forma permite-se que as redes LSTM se lembrem ou esqueçam seletivamente das informações por longos períodos, tornando-as adequadas para tarefas que exigem alteração de dependências de longo prazo. (Akhtar & Feng, 2022)

São normalmente utilizados para aplicações de cibersegurança, tais como deteção de intrusão e deteção de *malware*.

Restricted Boltzmann Machine (RBM):

Uma Máquina de Boltzmann Restrita (RBM) (G. Hinton, 2010) é um tipo de rede neural artificial generativa. Os RBMs são usados para aprendizagem não supervisionada, onde aprendem uma representação compactada dos dados de entrada sem a necessidade de dados de saída rotulados.

O RBM consiste em duas camadas: uma camada visível e uma camada oculta. A camada visível corresponde aos dados de entrada e a camada oculta corresponde aos recursos aprendidos. Cada neurónio na camada visível está conectado a cada neurónio na camada oculta, mas não há conexões entre os neurónios dentro da mesma camada.

O RBM aprende uma distribuição de probabilidade sobre os dados de entrada, minimizando uma função de energia que mede a diferença entre as previsões do modelo e os dados de entrada reais. Os pesos do RBM são ajustados usando uma técnica chamada divergência contrastiva, que envolve a amostragem da distribuição de probabilidade do modelo e a distribuição de dados para estimar o gradiente da função de energia.

Uma vez treinado, o RBM pode ser usado para gerar novos dados por amostragem a partir da distribuição de probabilidade aprendida. Os RBMs têm sido usados para uma variedade de tarefas, como reconhecimento de imagens, sistemas de recomendação e processamento de linguagem natural.

Uma limitação dos RBMs é que podem ser difíceis de treinar em dados multidimensionais, como imagens ou texto. No entanto, os RBMs têm sido usados como blocos de construção para modelos mais complexos, como Deep Belief Networks, que mostraram resultados promissores em várias tarefas. (Ackley et al., n.d.; Fischer & Igel, 2012; Salakhutdinov et al., n.d.)

As RBM têm sido usadas em cibersegurança para uma variedade de tarefas, como deteção de intrusão, deteção de anomalias e deteção de *malware*.(Berman et al., 2019; Li et al., 2021)

Deep Belief Networks (DBNs):

Deep Belief Networks (DBNs) (G. E. Hinton & Osindero, 2006) é um tipo de arquitetura de rede neural que consiste em várias camadas de máquinas de Boltzmann restritas (RBMs). Os DBNs são usados principalmente para aprendizagem não supervisionada, onde são treinados em dados não rotulados para aprender uma representação hierárquica dos dados de entrada.

Cada camada do modelo DBN consiste num RBM. Os RBMs são treinados usando uma variante de divergência contrastiva, que envolve a amostragem da distribuição de probabilidade do modelo e a distribuição de dados para estimar o gradiente dos parâmetros do modelo.

Depois de treinar os RBMs, as camadas do DBN são "desenroladas" para criar uma rede neural feedforward. A rede resultante pode ser ajustada usando aprendizagem supervisionada para executar uma tarefa específica, como classificação ou regressão.

Os modelos DBNs têm sido usados para uma variedade de tarefas, como reconhecimento de imagem e fala, processamento de linguagem natural e sistemas de recomendação. São particularmente úteis para modelar dados de alta dimensão com dependências complexas, pois podem aprender uma representação hierárquica que captura padrões locais e globais nos dados.

No entanto, o treino de DBNs pode ser computacionalmente intensivo e requer um ajuste cuidadoso da taxa de aprendizagem entre outros parâmetros. Os recentes avanços na aprendizagem profunda, como as redes neurais convolucionais e recorrentes, substituíram em grande parte os DBNs para muitas tarefas, mas continuam a ser uma ferramenta importante para a aprendizagem não supervisionada e a compreensão dos princípios da aprendizagem profunda.(G. E. Hinton et al., n.d.; Klein et al., 2006)

Os modelos DBNs foram aplicados à cibersegurança em tarefas como a deteção de intrusões e a deteção de *malware*. (Berman et al., 2019; Li et al., 2021; Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

Autocodificadores:

Autocodificadores (Sarcià & Cantone, 2009; *Unsupervised Feature Extraction with Autoencoder MGI Mestrado Em Gestão de Informação Master Program in Information Management*, n.d.) são um tipo de arquitetura de rede neural que é usado para aprendizagem não supervisionada. São projetados para aprender uma representação compactada, ou codificação, de dados de entrada e, em seguida, reconstruir os dados originais a partir dessa codificação. Noutras outras palavras, os codificadores automáticos aprendem a codificar dados num espaço de dimensão inferior e, em seguida, descodificá-los de volta à sua forma

original.

Os autocodificadores consistem em duas partes principais: uma rede codificadora e uma rede de descodificadores. A rede codificadora recebe os dados de entrada e produz uma representação compactada deles, muitas vezes chamada de *bottleneck* ou representação latente. De seguida, a rede descodificadora usa essa representação de *bottleneck* e reconstrói os dados de entrada originais.

O processo de treino de um *autoencoder* envolve minimizar o erro de reconstrução entre os dados de entrada originais e sua saída reconstruída. Desta forma força a rede a aprender uma representação compactada dos dados de entrada que é útil para reconstruí-los.

Uma das principais aplicações dos autocodificadores é a redução da dimensionalidade, onde são usados para aprender uma representação de baixa dimensão de dados de alta dimensão, como imagens ou texto. Os autocodificadores também podem ser usados para compressão de dados, deteção de anomalias e remoção de ruído de imagens, entre outros.

Podem ser utilizados para aplicações na cibersegurança, tais como deteção de anomalias e deteção de intrusão. (Berman et al., 2019; Li et al., 2021; Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020)

Generative Adversarial Networks (GANs):

Generative Adversarial Networks (GANs) (Goodfellow et al., n.d.; Radford et al., 2015) é um tipo de arquitetura de rede neural usada para modelagem generativa. As GANs consistem em duas redes: uma rede geradora e uma rede discriminadora.

A rede geradora recebe ruído aleatório como entrada e produz dados sintéticos que imitam a distribuição dos dados de treino. A rede discriminadora recebe dados reais e sintéticos como entrada e prevê se os dados são reais ou sintéticos.

Durante o treino, a rede geradora é treinada para produzir dados sintéticos que são cada vez mais difíceis para o discriminador distinguir dos dados reais. Ao mesmo tempo, a rede discriminadora é treinada para se tornar melhor na distinção entre dados reais e sintéticos.

As duas redes são treinadas simultaneamente num cenário de jogo, onde o gerador está constantemente a tentar produzir dados sintéticos que são indistinguíveis dos dados reais, enquanto o discriminador está a tentar classificar corretamente os dados como reais ou sintéticos. Isto conduz a uma forma de treino contraditório, em que as duas redes competem entre si para melhorar o seu desempenho.

Os GANs têm sido usados para uma variedade de tarefas, como gerar imagens realistas, sintetizar fala e música e gerar texto em linguagem natural. Uma vantagem dos GANs em relação a outros modelos generativos é que podem produzir amostras altamente realistas e

diversificadas que capturam a distribuição subjacente dos dados de treino.

No entanto, as GANs podem ser difíceis de treinar e propensas à instabilidade. Os resultados às vezes podem ser imprevisíveis e difíceis de controlar.

Podem ser usados para aplicações de cibersegurança, como a geração de dados sintéticos de tráfego de rede para testar sistemas de deteção de intrusão.(Berman et al., 2019; Li et al., 2021)

Deep Reinforcement Learning (DRL):

O Deep Reinforcement Learning (DRL) (Rumelhart et al., 1986) é um método de aprendizagem profunda que combina algoritmos de aprendizagem por reforço com redes neurais profundas para permitir que os modelos aprendam com suas interações com um ambiente.

Na aprendizagem por reforço, um modelo interage com um ambiente agindo e recebendo recompensas ou penalizações. O objetivo do modelo é aprender uma política, que é um mapeamento dos estados para as ações, que maximiza a recompensa acumulada ao longo do tempo. No DRL, redes neurais profundas são usadas para representar a política ou função de valor do modelo, permitindo que ele aprenda com entradas de alta dimensão, como imagens, áudio ou texto.

O DRL foi aplicado com sucesso a uma vasta gama de problemas, tais como jogos, robótica e condução autónoma.

Um dos principais desafios da DRL é equilibrar a exploração do meio ambiente com a exploração dos conhecimentos atuais do modelo. As redes neurais profundas podem facilmente sobreajustar-se aos dados de treino, levando a uma má generalização e desempenho em novas tarefas. Para enfrentar este desafio, várias técnicas como repetição de experiência, estratégias de exploração e regularização foram desenvolvidas.

De um modo geral, a DRL tem-se revelado muito promissora na resolução de problemas complexos e reais, mas ainda existem muitos desafios e oportunidades para mais investigação e desenvolvimento.(Mnih et al., n.d., 2015)

DRL tem sido aplicado à cibersegurança na formação de moldelos para detetar e responder a ciberataques em tempo real.(Shaukat, Luo, Varadharajan, Hameed, & Xu, 2020; Wazid et al., 2022)

Rede neural recursiva:

As Redes Neurais Recursivas são um tipo de arquitetura de rede neural projetada para processar dados estruturados, como árvores ou gráficos. Nas redes neuronais recursivas, cada nó representa uma subestrutura nos dados de entrada, e a rede processa os dados

recursivamente combinando as representações dos nós filho para calcular a representação do nó pai.

As redes neuronais recursivas são particularmente úteis para tarefas como processamento de linguagem natural, onde frases e parágrafos podem ser representados como árvores ou gráficos. Ao processar a entrada recursivamente, as redes neuronais recursivas podem capturar dependências de longo alcance e relações semânticas entre palavras ou frases.

Existem vários tipos de redes neuronais recursivas onde os diferentes modelos diferem na sua arquitetura e métodos de treino, mas todos partilham o processamento recursivo de dados estruturados.(Liu et al., n.d.)

Um desafio com as redes neuronais recursivas é que a estrutura recursiva da rede pode resultar em alta complexidade computacional e requisitos de memória, particularmente para grandes conjuntos de dados.(Berman et al., 2019)