INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL




SISTEMA DE DETEÇÃO DE TRANSAÇÕES FRAUDULENTAS NO E-COMMERCE ATRAVÉS DE MACHINE LEARNING

PEDRO FRANCISCO DE BORGES CASTRO DE RODRIGUES SOARES
Outubro de 2023

SISTEMA DE DETEÇÃO DE TRANSAÇÕES FRAUDULENTAS NO E-COMMERCE ATRAVÉS DE MACHINE LEARNING

Pedro Francisco de Borges Castro de Rodrigues Soares

Aluno nº: 1160714

Dissertação para obtenção do Grau de

Mestre em Engenharia de Inteligência Artificial

Orientador: Doutor António Constantino Lopes Martins, Professor Adjunto do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Júri

Presidente: Doutor Carlos Fernando da Silva Ramos, Professor Coordenador Principal do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Vogais: Doutora Isabel Cecília Correia da Silva Praça Gomes Pereira, Professora Coordenadora do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Doutor António Constantino Lopes Martins, Professor Adjunto do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Porto, setembro de 2023

Resumo

O crescimento exponencial do comércio eletrónico trouxe inúmeras vantagens e oportunidades ao facilitar o estilo de vida dos seres humanos. No entanto, deu também origem a um grave problema: a fraude online. Com o propósito de colmatar este problema, este trabalho aborda a necessidade de desenvolver sistemas de deteção de fraude complexos no âmbito do comércio eletrónico.

Após uma revisão abrangente da literatura, foram identificadas e implementadas técnicas que contribuíram para a melhoria dos projetos existentes, permitindo uma análise comparativa mais precisa. Neste contexto, os algoritmos de RF, LR, SVM, KNN, DT, LSTM e CNN, por serem os mais adequados a sistemas de classificação pela sua versatilidade e capacidade de aprender padrões complexos nos dados, foram aplicados a três conjuntos de dados distintos.

Para avaliar rigorosamente os modelos propostos, o conjunto de dados foi dividido em 70% de dados para treino e os restantes 30% para teste. Cada um dos conjuntos de dados apresenta características específicas, de forma a avaliar o impacto de técnicas de *oversampling* e *undersampling.* Os algoritmos foram aplicados também aos mesmos conjuntos com os dados normalizados, para inferir quais os modelos que beneficiam desta normalização.

Os resultados demonstraram que os modelos RF e CNN apresentaram um desempenho superior em comparação com os restantes algoritmos testados. Estes algoritmos foram posteriormente otimizados com a exploração dos hiper-parâmetros respetivos, o que permitiu melhorar o desempenho do modelo e, por sua vez, alcançar resultados de maior qualidade.

A utilização de inteligência artificial na deteção de fraude no comércio eletrónico é fundamental para proteger os interesses tanto das empresas como dos consumidores. Este trabalho teve como foco principal contribuir para o avanço dos sistemas de deteção de transações fraudulentas ao fornecer informações sobre pontos positivos e negativos de vários algoritmos de *machine learning* no contexto do problema em questão.

Palavras-chave: Inteligência artificial, Comércio Eletrónico, Fraude Online, Deteção de Fraudes, Machine Learning, Deep Learning, Sistemas de Classificação

Abstract

The exponential growth of e-commerce has brought numerous advantages and opportunities by facilitating the lifestyle of human beings. However, it has also given rise to a serious problem: online fraud. With the purpose of solving this problem, this work addresses the imperative need to develop complex fraud detection systems within the scope of electronic commerce.

After a systematic review of the literature, different techniques were identified and implemented that contributed to the improvement of existing projects, allowing for a more accurate comparative analysis. In this context, the RF, LR, SVM, KNN, DT, LSTM and CNN algorithms, as they are the most suitable for classification systems due to their versatility and ability to learn complex patterns in data, were applied to three distinct datasets.

To rigorously evaluate the proposed models, the dataset was divided into 70% training data and the remaining 30% to testing data. Each of the datasets consists in specific characteristics, in order to evaluate the impact of oversampling and undersampling techniques. The algorithms were also applied to the same datasets with normalized data, to infer which models benefit from this normalization.

The results demonstrated that the RF and CNN algorithms presented superior performance compared to the remaining algorithms tested. These algorithms were subsequently optimized by exploring the respective hyper-parameters, which allowed improving the model's performance and, in turn, achieving higher quality results.

The use of artificial intelligence to detect fraud in e-commerce is essential to protect the interests of both companies and consumers. This work's main focus was to contribute to the advancement of fraudulent purchase detection systems by providing information about the positive and negative points of various machine learning algorithms in the context of the problem in question.

Keywords: Artificial Intelligence, E-Commerce, Online Fraud, Fraud Detection, Machine Learning, Deep Learning, Classification Systems

Agradecimentos

Antes de mais quero agradecer ao meu orientador, Dr. Prof. António Constantino pelo valioso apoio e orientação demonstrado durante todo o processo de elaboração do presente documento. Os conselhos e recomendações que proporcionou foram fundamentais para o desenvolvimento deste trabalho. Além disso, desejo estender os meus agradecimentos a todos os professores com quem tive a honra de trabalhar ao longo deste mestrado. Cada um deles contribuiu significativamente para a elaboração e conclusão desta tese, ao enriquecer o meu conhecimento durante todo o mestrado.

À minha família, pai, mãe, irmãs e avós, quero expressar a minha mais profunda gratidão por serem os meus pilares desde o início e me apoiarem incondicionalmente em todas as fases da minha vida. Não posso também deixar de agradecer aos meus amigos e à minha namorada, que estiveram sempre do meu lado, oferecendo-me um enorme apoio e momentos de descontração nas situações mais desafiantes. As palavras de incentivo e carinho foram fundamentais para manter a minha motivação e determinação durante este período.

Estou eternamente grato por todo o amor e apoio que recebi durante a minha jornada académica.

Índice

1 Introdução 1

1.1 Contextualização 1

1.2 Descrição do problema 2

1.3 Objetivos 5

1.4 Motivação 6

1.5 Resultados esperados 6

1.6 Metodologia de investigação 7

1.7 Estrutura do documento 8

2 Estado de Arte e Formalização Teórica 11

2.1 Metodologia de pesquisa 11
- 2.1.1 Questões de pesquisa 12
- 2.1.2 Bases de conhecimento 12
- 2.1.3 Termos de pesquisa 13
- 2.1.4 Critérios de inclusão e exclusão 13
- 2.1.5 Extração de dados 14

2.2 E-commerce 16

2.3 *Machine Learning* 17
- 2.3.1 *Machine learning* em deteção de fraude 19

2.4 Sistemas de deteção de fraude 25
- 2.4.1 Sistemas de deteção de fraude com base em padrões comportamentais 29
- 2.4.2 Sistema de deteção de fraude e pré-processamento com SMOTE 34
- *2.4.3* Sistema de deteção de fraude utilizando *deep-learning* 37

2.5 Sumário 40

3 Experimentação 43

3.1 Conjunto de dados 43
- 3.1.1 Ética e proteção de dados 46

3.2 Análise exploratória de dados 47
- 3.2.1 Tratamento de dados 53

3.3 Sumário 58

4 Implementação e Avaliação 60

4.1 Algoritmos e métricas de performance 60
- 4.1.1 Logistic Regression 63
- 4.1.2 Random Forest 64
- 4.1.3 Support vector machine 66
- 4.1.4 K-Nearest neighbours 67
- 4.1.5 Decision Tree 68

4.1.6 Long Short-Term Memory 69
4.1.7 Convolutional Neural Networks 71
4.2 Avaliação dos algoritmos 72
4.3 Modelo proposto – algoritmos com hiper-parâmetros otimizados 73
4.3.1 *Random Forest* com hiper-parâmetros otimizados 75
4.3.2 *Convolutional Neural Networks* com hiper-parâmetros otimizados 77
4.4 Discussão e comparação de resultados com a literatura 78
4.5 Sumário 83
5 Conclusões finais 84
5.1 Limitações 86
5.2 Trabalho futuro 87

Lista de Figuras

Figura 1 - Fraude de triangulação baseado em (Yen, 2021) 4
Figura 2 - Grupos de fraude baseado em (Rodrigues et al., 2022) 5
Figura 3 – Diagrama PRISMA..... 15
Figura 4 – Variações mensais de faturação do retalho online durante a pandemia, baseado em (Szász et al., 2022)..... 17
Figura 5 – Fluxograma do mecanismo implementado em (Mathew et al., 2022)..... 26
Figura 6 – Gráfico de comparação da performance dos algoritmos antes do pré-processamento em (Abhirami et al., 2021) 28
Figura 7 - Gráfico de comparação da performance dos algoritmos após pré-processamento em (Abhirami et al., 2021) 29
Figura 8 – Comparação da exatidão em (Raja et al., 2021) 31
Figura 9 – Comparação da revocação em (Raja et al., 2021)..... 31
Figura 10 – Modelo de deteção de fraude em (Li, 2022)..... 32
Figura 11 – Exatidão do modelo de JiaoLong Li, (2022), em amostras de treino e teste 33
Figura 12 – Exatidão do modelo proposto nas amostras de teste em (Li, 2022) numa amostra de 1000 registos..... 33
Figura 13 – Comparação da exatidão do modelo proposto em (Li, 2022) com outras técnicas de *machine learning*..... 34
Figura 14 – Rácio do conjunto de dados antes do *oversampling* em (Saputra & Suharjito, 2019) 35
Figura 15 – Rácio do conjunto de dados depois do *oversampling* em (Saputra & Suharjito, 2019) 35
Figura 16 – Resultados da exatidão em (Saputra & Suharjito, 2019) 36
Figura 17 - Resultados da revocação em (Saputra & Suharjito, 2019) 36
Figura 18 - Resultados da pontuação f-1 em (Saputra & Suharjito, 2019) 37
Figura 19 – Pontuação f-1 obtida no SCD, ECD e TCD em (T. Nguyen et al., 2020) 39
Figura 20 – Resultados das técnicas de *oversampling* e *undersampling* aplicadas ao conjunto ECD em (T. Nguyen et al., 2020) 39
Figura 21 – SMOTE *vs* Distribuição normal no conjunto de teste ECD em (T. Nguyen et al., 2020) 40
Figura 22 - Contagem de fraudes..... 47
Figura 23 – Rácio de valores nulos..... 48
Figura 24 – "*ProductCD*" e respetiva percentagem de fraude..... 49
Figura 25 – "*P_email_domain*" e respetiva percentagem de fraude..... 50
Figura 26 - "*R_email_domain*" e respetiva percentagem de fraude 50
Figura 27 - "*DeviceType*" e respetiva percentagem de fraude 51
Figura 28 – "DeviceInfo" e respetiva percentagem de fraude..... 52
Figura 29 – "*Card4*" e respetiva percentagem de fraude 53
Figura 30 - "*Card6*" e respetiva percentagem de fraude 53
Figura 31 – Distribuição dos dados após a aplicação da técnica *SMOTE*..... 56

Figura 32 - Distribuição dos dados após a aplicação da técnica *RandomUnderSampler*..... 57

Lista de Tabelas

Tabela 1 – Questões de pesquisa 12
Tabela 2 – Repositórios científicos 12
Tabela 3 – Termos de pesquisa 13
Tabela 4 – Critérios de inclusão 13
Tabela 5 – Critérios de exclusão 14
Tabela 6 – *Query string* 14
Tabela 7 – Tabela de comparação entre os diferentes métodos de *machine learning*, com base em (Bhatt & Meniya, 2022), (Ghosh et al., 2020), (T. T. Nguyen et al., 2019) , (Wang et al., 2020) e (Esenogho et al., 2022) 25
Tabela 8 – Comparação da performance dos algoritmos em (Mathew et al., 2022) 26
Tabela 9 – Comparação da performance dos algoritmos antes do pré-processamento em (Abhirami et al., 2021) 28
Tabela 10 - Comparação da performance dos algoritmos após pré-processamento em (Abhirami et al., 2021) 28
Tabela 11 – Características da tabela de transações 45
Tabela 12 – Características da tabela de identidade 45
Tabela 13 – Matriz de confusão 61
Tabela 14 – Resultados do LR nos conjuntos de dados não normalizados 63
Tabela 15 - Resultados do LR nos conjuntos de dados normalizados 64
Tabela 16 – Resultados do RF nos conjuntos de dados não normalizados 65
Tabela 17 – Resultados do RF nos conjuntos de dados normalizados 65
Tabela 18 – Resultados do SVM nos conjuntos de dados não normalizados 66
Tabela 19 – Resultados do SVM nos conjuntos de dados normalizados 66
Tabela 20 – Resultados do KNN nos conjuntos não normalizados 67
Tabela 21 – Resultados do KNN nos conjuntos normalizados 68
Tabela 22 – Resultados do DT nos conjuntos não normalizados 69
Tabela 23 – Resultados do DT nos conjuntos normalizados 69
Tabela 24 – Resultados do LSTM nos conjuntos não normalizados 70
Tabela 25 - Resultados do LSTM nos conjuntos normalizados 70
Tabela 26 – Resultados do CNN nos conjuntos não normalizados 71
Tabela 27 - Resultados do CNN nos conjuntos normalizados 72
Tabela 28 – Hiper-parâmetros do RF que obtiveram melhores resultados 76
Tabela 29 – Resultados do RF com hiper-parâmetros otimizados 76
Tabela 30 - Hiper-parâmetros da CNN que obtiveram melhores resultados 77
Tabela 31 – Resultado da CNN com hiper-parâmetros otimizados 77
Tabela 32 - Resultados obtidos vs Resultados dos artigos estudados 80

Acrónimos

Lista de Acrónimos

BDM	*Big Data Mining*
CNN	*Convolutional Neural Network*
DSR	*Design Science Research*
DT	*Decision Tree*
HP	Hiper parâmetros
IA	Inteligência artificial
ISEP	Instituto Superior de Engenharia do Porto
KNN	*K-Nearest Neighbours*
LR	*Logistic Regression*
LSTM	*Long Short-Term Memories*
MEIA	Mestrado de Engenharia da Inteligência Artificial
NN	*Neural Network*
RF	*Random Forest*
RGPD	Regulamento Geral de Proteção de Dados
SMOTE	*Synthetic Minority Oversampling Technique*
SVM	*Support Vector Machines*

1 Introdução

No presente capítulo será descrito uma contextualização do tema em questão assim como os principais problemas identificados dentro da área que o projeto se insere. Serão também referidos os objetivos principais para solucionar o problema em questão, os motivos para o qual o tema em questão foi escolhido e os possíveis contributos para a área de inteligência artificial. Será também descrito a metodologia de investigação que foi usada e uma breve descrição da estrutura do documento em causa.

1.1 Contextualização

O comércio online tem sido um mercado que se encontra em bastante desenvolvimento, devido aos constantes avanços de tecnologias, quer de informação, quer de informática. Como consequência, há uma maior utilização de plataformas online, incluindo, claro, plataformas de comércio eletrónico, mais conhecido como e-commerce (Szász et al., 2022). Este mercado, que já era bastante utilizado numa altura anterior à pandemia, sofreu um crescimento ainda maior durante a este período. Cada vez mais, as pessoas optam por fazer compras online, quer no ramo da moda, de eletrodomésticos, produtos tecnológicos para lazer, entre outros (Szász et al., 2022).

As plataformas de *e-commerce* oferecem inúmeras vantagens, entre elas: o processo de compra é mais rápido, os custos são menores, os clientes conseguem rapidamente comparar os preços entre as diferentes entidades, o que torna este mercado flexível para os clientes e as respostas aos pedidos do mercado são mais rápidas (Y. Liu et al., 2021). Estas situações levam a um aumento exponencial da procura por compras online, incluindo produtos e necessidades diárias como alimentação e saúde.

Um mercado tão conhecido e utilizado como o *e-commerce* atrai muitas pessoas suspeitas de cometerem fraude devido ao elevado número de transações que se realizam todos os dias (Rodrigues et al., 2022). As fraudes causam grandes prejuízos financeiros e prejudicam a reputação das plataformas do comércio eletrónico. Portanto, embora o uso de transações

online tenha tornado a vida mais conveniente e produtiva, também abriu a porta a uma variedade de ameaças, como as transações fraudulentas. Como consequência, a deteção de fraudes em transações no *e-commerce* tornou-se uma questão importante de ser focada (Zhou et al., 2019).

A deteção de fraude dentro do comércio eletrónico tornou-se uma área de investigação bastante relevante nos últimos anos onde abordagens de *machine-learning* e *data mining* são utilizados para combater este problema que, apesar de bastantes esforços em detetar fraude de cartões de crédito, a luta contra estes ciberataques ainda está no início devido a algumas dificuldades (Carta et al., 2019). Por questões de segurança, bancos e empresas financeiras proíbem a divulgação de dados confidenciais, o que limita assim a sua investigação pois restringem a quantidade de informação relevada para o exterior (Rtayli & Enneya, 2020).

1.2 Descrição do problema

Um dos principais problemas do comércio eletrónico é a constante tentativa de fraude e, por isso, existem vários tipos de fraude nomeadamente: fraude de transações de dinheiro através do cartão de crédito, fraude amigável, fraude de aquisição de conta, fraude de reembolso e fraude de triangulação (Varga, 2022).

Uma fraude no comércio eletrónico engloba qualquer tipo de ação maliciosa projetada para explorar lojas online. Os ataques mais comuns estão relacionados com transações fraudulentas, feitas com números de cartão de crédito roubados (Yen, 2021). No entanto, e apesar de ser uma das fraudes mais comuns, não é a única, e todos os tipos de fraudes acontecem, infelizmente, muitas vezes. Para uma melhor contextualização, os tipos de fraudes podem ser definidos como (Varga, 2022):

- **Fraude de transações através do cartão de crédito:** este tipo de fraude ocorre, como o nome indica, quando alguém faz uma compra não autorizada com informações de cartão de crédito que não são dessa pessoa. Estes números do cartão de crédito roubados são normalmente obtidos através de ataques de *phishing* que entram em contacto com as pessoas por telefone ou e-mail e tentam convencê-las a fornecer as informações do cartão de crédito. Muitas das vezes as pessoas até oferecem recompensas de maneira a obter este tipo de informação, ou fazem-se mesmo passar por entidades populares, empresas conhecidas e confiáveis, de maneira a obterem o que pretendem. Para além disso, os cartões de crédito também podem ser obtidos através de *hacks* onde o hacker consegue obter informações sobre os cartões de crédito dos clientes (Bradley, 2022).

- **Fraude amigável:** este tipo de fraude ocorre quando um cliente faz, conscientemente, uma compra, mas pede o reembolso ao banco, na tentativa de obter o produto de forma gratuita. No entanto, nem todas as fraudes de transação são causadas por

contrabandistas profissionais. Às vezes, um reembolso é iniciado por alguém cujo cartão foi roubado.

- **Fraude de aquisição de conta:** este tipo de fraude ocorre quando um fraudador obtém acesso a uma conta que pertence a outra pessoa. A maioria das pessoas utiliza palavras-passes de baixa segurança e a mesma para vários sites diferentes, o que permite aos contrabandistas *hackear* estas contas com um pouco de informação pessoal encontrada na internet. O *phishing* é também uma causa comum de aquisição de contas. Este processo envolve o envio de e-mails ou mensagens de texto falsas, que parecem ser de uma empresa confiável, como um banco ou um site de comércio eletrónico, onde solicitam informações confidenciais, como palavras-passes ou números de cartão de crédito (Marley, 2019).

- **Fraude de reembolso:** neste tipo de fraudes, uma pessoa compra um produto com um cartão de crédito roubado e devolve o produto, de maneira a ser reembolsado, mas neste caso pede que o reembolso seja devolvido para um cartão ou conta diferente. Normalmente, as empresas têm também uma política de reembolso bastante generosa o que aumenta ainda mais a fraude de reembolso (Varga, 2022).

- **Fraude de triangulação:** este tipo de fraude envolve três intervenientes: o fraudador que possui uma loja falsa, um cliente legítimo e uma loja legítima online. Neste caso, o cliente faz uma compra numa loja online, que por sua vez é falsa e faz parte do esquema do fraudador. Depois de receberem o pedido, o fraudador compra o mesmo produto, mas, neste caso, numa loja legítima usando o número de cartão de crédito da pessoa que está a ser roubada e enviam esse mesmo produto para a sua morada pois assim, o cliente recebe o produto que comprou. O comprador, ao perceber que o pagamento extra, irá pedir o reembolso ao qual a empresa legítima irá iniciar o processo de devolução (Varga, 2022). No entanto, o primeiro pagamento que foi feito, na loja falsa do fraudador, nunca será devolvido e o fraudador manterá o dinheiro. Mesmo que seja pedido o reembolso, o fraudador não irá responder. Na Figura 1 está representado um esquema sobre este tipo de fraude (Yen, 2021).

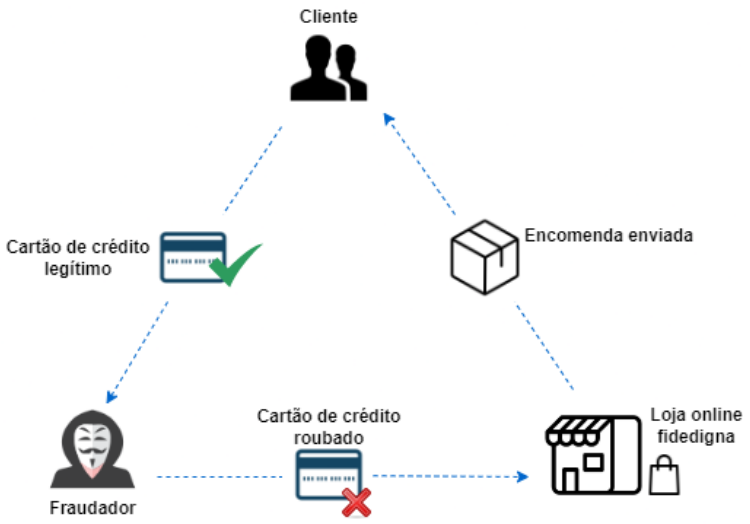

Figura 1 - Fraude de triangulação baseado em (Yen, 2021)

Estes tipos de fraude podem ser classificados também em quatro grupos distintos (Rodrigues et al., 2022):

I. Cartão de crédito
II. Transações online
III. Transações financeiras
IV. Falsos vendedores

O primeiro grupo, cartão de crédito, é o campo mais estudado e explorado em transações fraudulentas pois tornou-se o meio de pagamento mais utilizado em todo o mundo e, claro, no comércio eletrónico, abrindo uma enorme possibilidade de lucro para contrabandistas caso a segurança da loja online não seja abordada de uma forma correta e prevenida (Rodrigues et al., 2022). Este meio de pagamento permite também aos contrabandistas explorar todas as entidades envolvidas na transação, o que torna um problema não apenas para o comércio eletrónico, como também para instituições financeiras.

O segundo grupo remete-nos para transações online. Os cartões de crédito estão, de certa maneira, inseridos neste grupo, pois uma transação online pode ser efetuada através de um cartão de crédito, a diferença é que neste tipo incluem-se todas as transações online independentemente do meio de pagamento (Rodrigues et al., 2022).

Já o terceiro grupo são as, transações financeiras, englobam não só transações de compras, como também transações financeiras fraudulentas em forma de pagamento, transferência e/ou levantamento. Neste grupo, a possibilidade de fraude é maior pois existem diferentes tipos de transações e cada tipo permite uma abordagem diferente por parte do fraudador (Rodrigues et al., 2022).

O quarto grupo são os falsos vendedores (*fake sellers*). Este grupo concentra-se nos contrabandistas que se fazem passar por lojas fidedignas e autênticas, que tentam vender

produtos falsos ou que nem sequer existem, a clientes (Rodrigues et al., 2022). Na Figura 2 está representado um esquema dos grupos de fraude, assim como os tipos de transações.

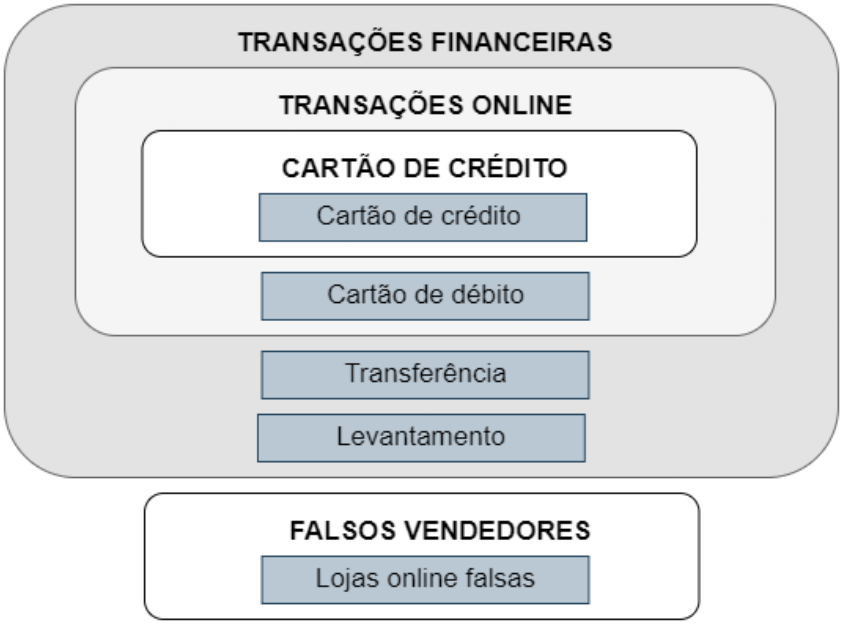

Figura 2 - Grupos de fraude baseado em (Rodrigues et al., 2022)

O problema de fraude é, então, uma preocupação importante para os lojistas e os consumidores que este projeto visa combater.

1.3 Objetivos

O principal objetivo do presente trabalho prende-se com a criação de um sistema que, através de modelos e técnicas de *machine learning*, detetará transações fraudulentas.

Assim, a questão de investigação que orientou o trabalho foi a seguinte: A definição de um modelo de deteção de fraudes através de algoritmos de *machine learning* pode potenciar a deteção de transações fraudulentas *online*?

Esta questão permitirá inferir se o modelo pretendido poderá proteger as empresas e clientes de atividades fraudulentas no comércio eletrónico. Para concretizar o objetivo geral foram definidos os seguintes objetivos específicos:

- **O1:** Analisar e estudar o estado de arte atual de sistemas de deteção de fraudes no comércio online;
- **O2:** Que modelos, métodos e/ou algoritmos são necessários implementar num sistema de forma a este obter resultados de maior rigor e precisão possível;

- **O3:** Extrair e aprimorar um conjunto de dados com informação necessária para a deteção de transações fraudulentas;
- **O4:** Aplicar vários modelos e técnicas para a deteção de transações fraudulentas;
- **O5:** Avaliar diferentes conjuntos de dados;
- **O6:** Avaliar a performance e precisão dos diferentes modelos e técnicas testados;
- **O7:** Aferir qual o melhor modelo para a deteção de transações fraudulentas.

1.4 Motivação

A motivação para este trabalho veio principalmente de um compromisso pessoal com a pesquisa e desenvolvimento de um sistema de deteção de compras fraudulentas, que visa a obtenção de conhecimento e experiência tanto na área financeira, como na área do comércio online e claro, na área de inteligência artificial e *machine-learning* especificamente. Foi um projeto autoproposto e aceite pelos responsáveis da disciplina e do Mestrado, no Instituto Superior de Engenharia do Porto (ISEP).

O interesse em adquirir conhecimento e desenvolver um projeto sobre cyber-segurança e o mercado financeiro foram um ponto crucial na tomada de decisão sobre o tema do projeto. Conjugando o mencionado com o comércio online, uma área que me cativa também bastante interesse por estar conectado comigo diariamente, deu origem ao tema em questão. O *e-commerce* tem se tornado cada vez mais popular nos últimos anos e é uma tendência que vai continuar em peso no futuro pois permite que as pessoas comprem produtos e serviços de forma rápida e conveniente, sem precisar de sair de casa. É também uma ótima opção para empresas pois permite que alcancem um público mais amplo e aumentem as suas vendas, reduzindo os custos e aumentando a eficiência.

Relativamente às áreas em questão, a motivação recai sobre contribuir para os futuros avanços tecnológicos na área do comércio online, utilizando inteligência artificial e auxiliar as plataformas do *e-commerce* sobre a importância da segurança da sua loja online. Estas plataformas devem aplicar as melhores políticas de segurança às suas infraestruturas de forma a proteger os dados dos utilizadores, como palavras-passes, números de cartões de crédito e morada, devem evitar ataques cibernéticos como *phishing*, *malware* ou outras ameaças de maneira a garantir a integridade e fiabilidade dos dados dos utilizados e fortalecer a confiança dos mesmos. Isto motiva a área da engenharia da inteligência artificial no sentido em que permite a criação de algoritmos cada vez mais precisos e eficientes para detetar fraudes.

1.5 Resultados esperados

A segurança online é um aspeto crucial da engenharia de computação e da tecnologia da informação. A internet é cada vez mais um bem essencial nas nossas vidas que usamos para comunicar, fazer compras, trabalhar e aprender. No entanto, como a internet é um ambiente virtual, é suscetível a ciberataques, como vírus, *malware*, *phishing*, ataques de negação de

serviço e outros problemas de segurança. Assim, proteger os dados e informações confidenciais dos utilizadores é obrigatório e extremamente necessário de forma a garantir a privacidade e confidencialidade dos mesmos (Zhou et al., 2019).

Sendo a fraude um problema crescente que pode prejudicar tanto as empresas como os consumidores finais, os sistemas de deteção de fraude em compras online são imperativos. As fraudes em compras online podem incluir o uso de cartões de crédito roubados ou falsificados, o uso de informações pessoais falsas ou o acesso não autorizado a contas bancárias. Os sistemas de deteção de fraude em compras online ajudam a proteger as empresas de perdas financeiras decorrentes, de transações fraudulentas e a proteger os consumidores finais de perdas financeiras e danos à reputação. Além disso, os sistemas de deteção de fraude em compras online ajudam a garantir a fiabilidade e a segurança das transações online, o que pode promover a confiança dos consumidores nas compras online e aumentar o volume de negócios das empresas (Szász et al., 2022). A segurança pode passar por verificar os endereços de cobrança e endereços de entrega fornecidos pelo consumidor com os registos do cartão de crédito ou do banco para verificar se são consistentes, verificar se o cartão de crédito é válido e se não está relacionado com tentativas de fraude anteriores, monitorizar o comportamento do utilizador durante a compra como, por exemplo, o tipo de dispositivo utilizado e o local de acesso, de maneira a identificar compras suspeitas, verificar informações pessoais fornecidas pelo consumidor como nome, endereço, data de nascimento, e-mail e garantir que são válidos e consistentes, entre outras (Carta et al., 2019).

Em resumo, os sistemas de deteção de fraude em compras online são importantes porque ajudam a proteger as empresas e os consumidores de fraudes em compras online e a garantir a fiabilidade e a segurança das transações online.

1.6 Metodologia de investigação

Uma metodologia de investigação representa um conjunto de procedimentos e técnicas utilizados para realizar uma pesquisa científica onde se inclui a definição do problema, a formulação de hipóteses e análise de dados. Considera-se que uma das principais vantagens de uma metodologia de investigação é orientar o autor na condução da pesquisa e garantir que os resultados obtidos sejam válidos e confiáveis (Warfield, 2010).

Este documento assenta numa metodologia de investigação denominada *Design Science Research* (DSR) que é adequada para criar soluções práticas para problemas do mundo real (Cheong et al., 2013). A metodologia DSR promove uma abordagem mais prática para resolver os problemas (Akker et al., 2006) e permite que a solução seja continuamente aprimorada, pois o design é incremental. Permite assim incluir alterações e melhorias a partir de evidências identificadas durante diferentes fases do período experimental (Molina et al., 2007).

Segundo (Peffers et al., 2007), o processo de interação da metodologia DSR pode ser definida por uma sequência de seis atividades, entre elas:

1. Identificação do problema e motivação;
2. Definição de objetivos da solução;
3. *Design* e desenvolvimento;
4. Demonstração;
5. Avaliação;
6. Comunicação.

Este processo foi adaptado ao projeto em questão, que deu origem assim a:

1. **Identificação do problema e motivação:** Nesta fase foi feita uma análise dos sistemas de deteção de fraude relativamente aos problemas e principais contributos do mesmo. Entre eles, combater o problema excessivo de fraudes no comércio eletrónico através de técnicas de aprendizagem automática de forma a diminuir a percentagem de transações fraudulentas;

2. **Definição de objetivos da solução:** Nesta fase foram definidos os objetivos da solução pretendida. O principal objetivo prende-se com desenvolver um sistema de deteção de fraude no comércio online através de algoritmos de *machine learning*;

3. **Design e desenvolvimento:** Nesta fase foi elaborado o *design* da solução e todo o processo de análise ao conjunto de dados para a criação de um modelo de deteções de fraude;

4. **Demonstração e avaliação:** Neste passo foi demonstrada a eficácia e eficiência, através de métricas de performance, do modelo proposto;

5. **Comunicação:** Esta fase consiste na publicação do projeto desenvolvido em bases de conhecimento para futuros estudos e desenvolvimentos sobre a área em questão.

1.7 Estrutura do documento

Este último subcapítulo do capítulo da Introdução pretende expor brevemente a estrutura e conteúdo do documento. Esta estrutura dividir-se-á em cinco capítulos principais: introdução, estado da arte e formalização teórica, análise e tratamento do conjunto de dados, implementação/avaliação e conclusões.

No primeiro capítulo, Introdução, são contextualizados o problema e objetivos do projeto dentro da área em que se insere, assim como as motivações para a elaboração deste projeto e os contributos para a área de engenharia da inteligência artificial. É também descrita a metodologia de investigação utilizada na elaboração do presente documento e a maneira como está estruturado. No segundo capítulo está presente o estado de arte atual de sistemas de

fraude no comércio eletrónico, que inclui trabalhos relacionados com o tema em questão e as principais contribuições e conclusões mencionadas pelos autores. Para além disso, é feita também uma formalização teórica sobre as áreas que o projeto se insere. É também mencionado a metodologia utilizada na pesquisa bibliográfica, assim como toda a extração de dados envolvente.

Na terceira parte do documento deparamo-nos com a experimentação dos dados. Neste capítulo é feito uma análise do conjunto de dados que será utilizado para a construção do sistema de deteção de compras fraudulentas. É ainda descrito o tratamento e pré-processamento de dados envolvente, assim como o processo de validação e testagem do mesmo.

No quarto capítulo, implementação e avaliação, são descritos todas as técnicas, métodos e algoritmos de *machine-learning*, incluindo *deep learning,* utilizados para o desenvolvimento do sistema de deteção de fraudes. Por fim, é feita uma avaliação dos resultados obtidos e uma comparação entres os diferentes modelos utilizados a fim de concluir qual o modelo com melhor performance.

No último capítulo são apresentadas a conclusões do estudo efetuado, onde é feito o balanço final do trabalho, realçando os aspetos principais do trabalho desenvolvido, os seus contributos e limitações e propostas de trabalho futuro.

2 Estado de Arte e Formalização Teórica

Neste capítulo será feita uma análise do estado da arte e uma formalização teórica do tema em questão. No estado da arte será feito uma revisão sistemática das principais pesquisas e trabalhos já realizados sobre os sistemas de deteção de fraude no e-commerce, com o objetivo de identificar as conclusões alcançadas e os desafios e limitações ainda pendentes. É descrita a metodologia utilizada e todo o processo de extração de dados envolvente. É também feita uma breve formalização teórica do *e-commerce* e *machine learning*, visto serem as principais áreas envolventes do trabalho em questão. No final do capítulo, é feito um breve sumário, como termo de comparação entre os trabalhos relacionados selecionados.

2.1 Metodologia de pesquisa

A metodologia de pesquisa utilizada na elaboração deste capítulo foi o PRISMA, do inglês *Preffered Reporting Items for Systematic Reviews and Meta-Analyses.* Esta metodologia, publicada em 2009 (Page et al., 2021), foi projetada para ajudar investigadores a relatar de forma transparente porque fizeram a revisão em questão, como fizeram e o que descobriram, assim como facilitar a comparação entre diferentes revisões sistemáticas. A metodologia PRISMA inclui também um diagrama, conhecido como diagrama PRISMA, Figura 3, que ilustra o processo de pesquisa e seleção de estudos para a pesquisa (Page et al., 2021).

O objetivo da pesquisa em questão recai sobre coletar todos os artigos científicos confiáveis e influentes que investigaram os sistemas de deteção de fraude no comércio eletrónico nos últimos anos. Em particular, explorar as técnicas e metodologias já desenvolvidas nos projetos selecionados são o principal objetivo desta pesquisa. Para atingir esse objetivo, foram definidas três questões de pesquisa.

2.1.1 Questões de pesquisa

Para os propósitos desta revisão sistemática, definiu-se a principal questão de pesquisa como: "Como podemos detetar e prevenir transações fraudulentas no e-*commerce*?". Para responder adequadamente a esta questão, definiu-se três questões de pesquisa, que podem ser encontradas na Tabela 1, em baixo representada.

QP1	Que métodos e tecnologias são utilizados no desenvolvimento de um sistema de deteção de fraude?
QP2	Quais são os comportamentos mais recorrentes numa compra/transação fraudulenta?
QP3	Como se devem as plataformas de *e-commerce* prevenir para combater as pessoas suspeitas de cometerem fraude?

Tabela 1 – Questões de pesquisa

A primeira questão foca-se em determinar quais são os métodos, tecnologias ou algoritmos que já foram implementados e utilizados na construção de um sistema de deteção de fraude. Esta questão irá ajudar a perceber em que nível estes sistemas usam esses métodos e como estes são implementados. A segunda questão recai sobre avaliar se existe algum comportamento regular e frequente que possa ser identificado de maneira a prever e combater as tentativas de fraude. Por último, a terceira questão foi elaborada de maneira a identificar técnicas e maneiras que as plataformas do comércio eletrónico podem implementar para prevenir as tentativas de fraude.

2.1.2 Bases de conhecimento

Para uma revisão sistemática e respetiva pesquisa científica, os dados extraídos devem ser credíveis, autênticos, seguros e disponíveis para futuras atualizações e partilha de dados. O objetivo é usar a mesma *query string* em todas as bases de conhecimentos selecionadas. Na Tabela 2, estão definidos os cinco repositórios científicos que foram utilizadas para este estudo.

BC1	eBook University Press Collection (EBSCOhost)
BC2	Academic Search Complete
BC3	IEEE Xplore Digital Library
BC4	Science Direct
BC5	Business Source Complete

Tabela 2 – Repositórios científicos

2.1.3 Termos de pesquisa

O próximo passo foi definir um conjunto de termos de pesquisa, os quais, a respeito de um domínio específico, selecionamos diferentes palavras-chaves relacionadas com esse domínio. Isto permite estabelecer uma relação entre os diferentes domínios que o estudo envolve e a pesquisar os melhores estudos científicos já realizados para responder às nossas questões de pesquisa da melhor maneira possível. Foi especificado três domínios principais e para cada um deles um conjunto de palavras-chaves, como pode ser visto na Tabela 3.

Fraude	"Fraud" **OR** "Scam"
Sistemas de deteção	"Detection systems" **OR** "Monitoring systems" **OR** "Alert Systems" **OR** "Tracking systems"
E-commerce	"E-commerce" **OR** "Electronic commerce" **OR** "Online store" **OR** "Online commerce" **OR** "Market Store"

Tabela 3 – Termos de pesquisa

2.1.4 Critérios de inclusão e exclusão

O processo de seleção de estudos e extração de dados inclui a definição de critérios de inclusão e exclusão para descartar resultados indesejáveis. Após essa filtragem, obtemos os resultados mais relevantes e pertinentes para a pesquisa. Os critérios de inclusão e exclusão estão definidos nas Tabela 4 e Tabela 5, respetivamente.

CI1	A fonte foca-se em sistemas de deteção de fraude no *e-commerce*
CI2	A fonte explora diferentes técnicas de *machine learning* aplicadas na área de deteção de fraude
CI3	A fonte identifica vantagens e limitações relacionados com os sistemas de deteção de fraude no *e-commerce* e as técnicas de *machine learning* envolvidas
CI4	A fonte analisa diferentes tipos de sistemas de deteção de fraude no *e-commerce*
CI5	A fonte descreve uma contribuição significativa para os campos de estudo

Tabela 4 – Critérios de inclusão

CE1	A fonte relacionada com sistemas de deteção de fraude tem mais de 5 anos
CE2	A fonte não se encontra em inglês
CE3	A fonte não esclarece como é que as técnicas de machine learning detetam possíveis tentativas de fraude
CE4	A fonte não apresenta estudos relacionados com o comércio online
CE5	Fontes duplicadas

Tabela 5 – Critérios de exclusão

2.1.5 Extração de dados

Os dados foram extraídos pelas bases de conhecimento mencionadas no ponto 2.1.2, através da query string representada na Tabela 6, com as respetivas palavras-chaves para cada domínio. É importante que as fontes relacionadas com os sistemas de deteção de fraude sejam inferiores a 5 anos pois é uma área que se encontra sempre em constante inovação e não se pretende encontrar artigos científicos que possam estar desatualizados. Além disso, é importante que os artigos estejam escritos em inglês e aplicados à área do comércio eletrónico, pois os sistemas de deteção de fraude podem ser aplicados a outras áreas. A Figura 3 ilustra o processo de tomada de decisão utilizado na extração de dados.

Query string	(“Fraud” **OR** “Scam”) **AND** (“Detection systems” **OR** “Monitoring systems” **OR** “Alert Systems” **OR** “Tracking systems”) **AND** (“E-commerce” **OR** “Electronic commerce” **OR** “Online store” **OR** “Online commerce” **OR** “Market Store”)

Tabela 6 – *Query string*

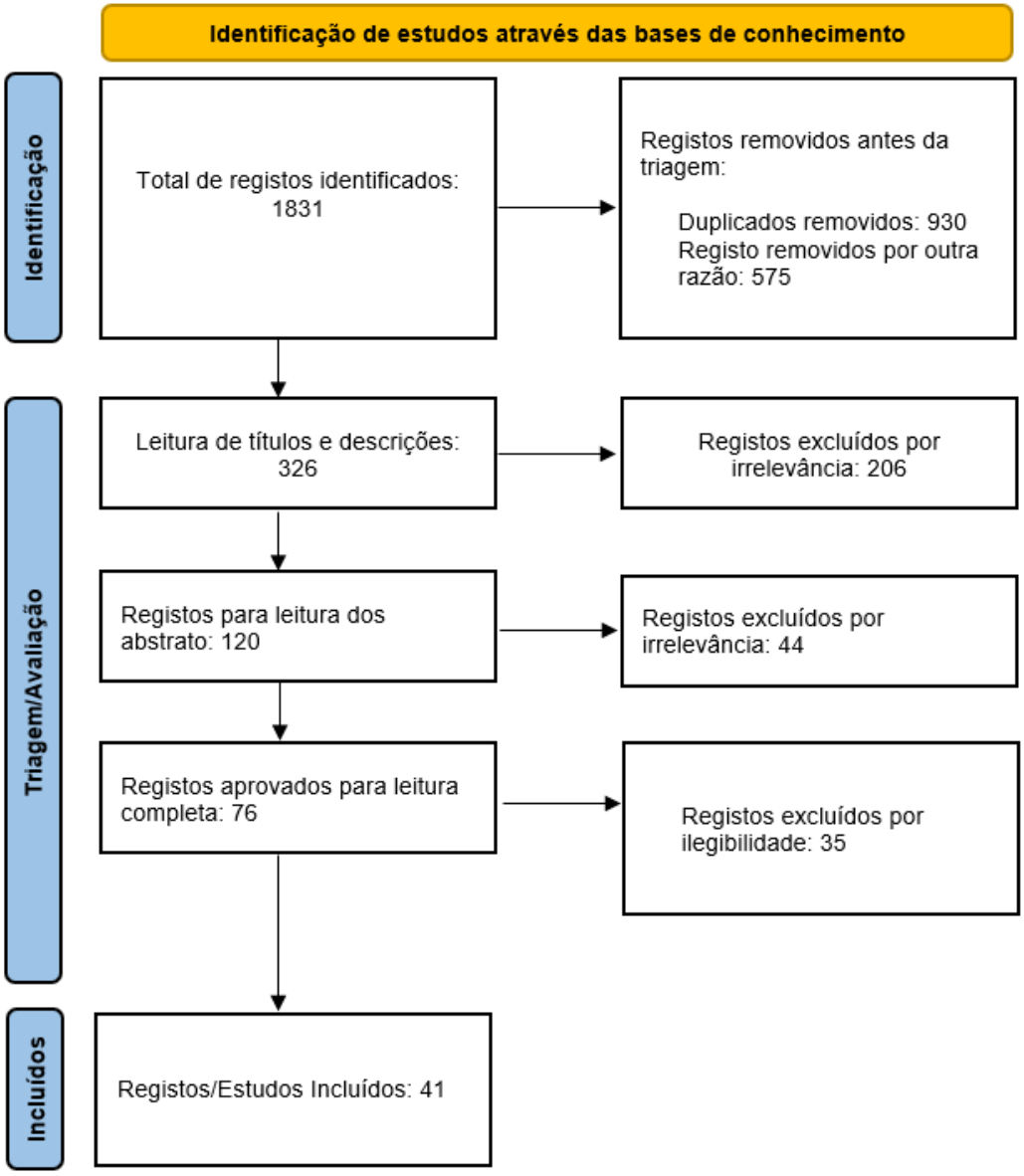

Figura 3 – Diagrama PRISMA

Da primeira pesquisa, resultou um total de 1831 registos (Figura 3). O primeiro passo do processo passou por remover os duplicados, neste caso 930 registos eram duplicados por, por exemplo, estarem presentes nas mesmas bases de conhecimento, o que diminuiu assim os registos para 326. Estes 326 registos passaram então para a fase seguinte. A primeira fase é a triagem/avaliação, onde apenas com a leitura dos títulos de cada artigo é possível perceber se o registo será relevante ou não para o tema em questão. Desta revisão rápida foram excluídos 206 artigos, passando assim para um total de 120 registos, que foram aprovados para a leitura do abstrato. Após a leitura do abstrato foi possível concluir que 44 dos 120 artigos aprovados não eram relevantes para as questões de pesquisa enumeradas, sobrando assim 76 registos

aprovados para leitura completa. Destes 76 artigos, 35 foram excluídos por serem inelegíveis, resultando assim em 41 resultados obtidos através da revisão sistemática efetuada.

Os 41 artigos auxiliaram tanto no capítulo do estado de arte, relativo aos trabalhos relacionados sobre o tema em questão, como também facilitaram e contribuíram para a elaboração da formalização teórica visto se encontrarem dentro das áreas relevantes para o trabalho atual entre elas o *e-commerce*, *machine learning* e *deep learning*. Estes artigos serão analisados e estudados nos capítulos seguintes.

2.2 E-commerce

O comércio eletrónico, *eletronic commerce* em inglês, ou simplesmente, *e-commerce*, é o processo de venda e compra de produtos ou serviços através da internet (Pahadi et al., 2022). Plataformas como a *Amazon*, *eBay* ou *Alibaba* são empresas notoriamente famosas dentro desta área que permitem a compra e venda de produtos online. A *Amazon* vende uma ampla variedade de produtos, incluindo livros, eletrodomésticos, roupas, brinquedos, alimentos, entre outros, enquanto que, o *eBay*, é uma plataforma de leilão online que permite que as pessoas vendam e comprem produtos novos ou usados. Já o *Alibaba* é uma empresa que oferece uma plataforma para empresas comprarem e venderem produtos em grandes quantidades diretamente dos fabricantes na China (Kulshrestha & Saini, 2020).

Este mercado expandiu-se significativamente com a pandemia causada pela propagação do coronavírus, SARS-CoV-2, onde as empresas e os consumidores foram forçados a se adaptarem às mudanças drásticas trazidas por este vírus. Por exemplo Sheth, (2020), argumenta que a pandemia teve vários efeitos poderosos e imediatos no comportamento do consumidor: estes substituíram velhos hábitos por novos, como a mudança para as compras online, permitindo assim "a loja voltar para casa". Em coerência, Jiang e Styos, (2021), propuseram que as pressões individuais durante o confinamento forçaram os consumidores a criar "uma nova realidade de compras no mundo do retalho" que envolveu maior utilização do mundo digital e o aumento de compras online.

De maneira a investigar como as mudanças no comportamento dos clientes e nas regulamentações governamentais impulsionaram a evolução do retalho online durante a pandemia, os investigadores Szász, Bálint, Csíki, Nagy, Rácz, Csala e Harris, (2022), criaram um gráfico (Figura 4), que demonstra a evolução mensal das vendas no retalho online durante a pandemia, entre fevereiro de 2020 e janeiro de 2022 em países europeus.

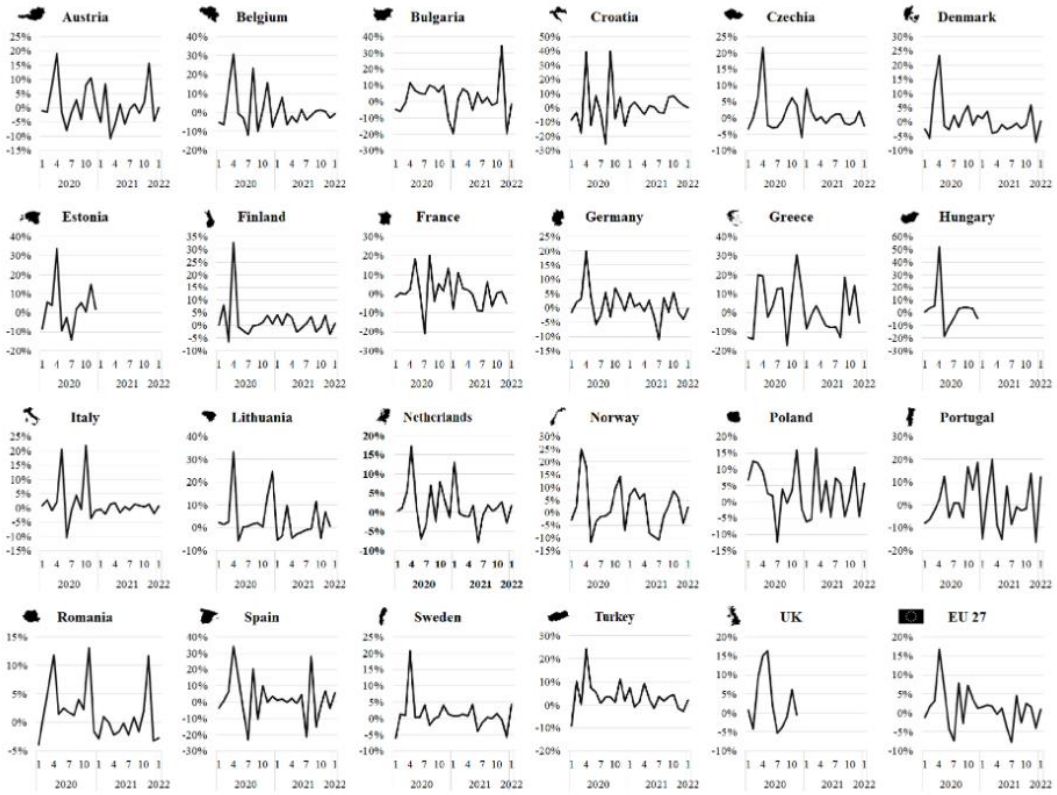

Figura 4 – Variações mensais de faturação do retalho online durante a pandemia, baseado em (Szász et al., 2022)

Foi utilizada uma base de dados do Eurostat, uma organização estatística da Comissão Europeia que produz dados estatísticos para a União Europeia, por ser considerada, pelos autores, a mais adequada. Fornece dados do retalho online para vinte e três países europeus, cobrindo assim todas as principais economias da Europa. A título de síntese, em apenas alguns meses, durante a pandemia, o retalho online cresceu tão exponencialmente, que levaria anos a atingir o mesmo nível em circunstâncias normais (Szász et al., 2022). Muitas empresas superaram o problema ao rapidamente apostarem no mundo online e, por sua vez, os clientes adaptaram-se a esta nova realidade e continuarão a efetuar compras online, em virtude da sua conveniência, simplicidade e rapidez.

2.3 *Machine Learning*

A aprendizagem automática ou aprendizagem da máquina, do inglês *machine learning*, é uma área da ciência da computação que se concentra em desenvolver sistemas que são capazes de aprender e de se adaptar com base em experiências (Abhirami et al., 2021). Os sistemas de *machine learning* são treinados com conjuntos, ao invés de serem programadas com instruções especificas, e utilizam algoritmos de aprendizagem para tomar decisões autonomamente. Isto

permite que estes sistemas sejam capazes de realizar tarefas complexas como o reconhecimento de imagem ou linguagem natural (Abhirami et al., 2021).

Machine learning refere-se a uma ampla quantidade de algoritmos e abordagens para a classificação, regressão, agrupamento e deteção de anomalias. É um sistema de inteligência artificial que permite que um sistema ou computador aprenda e decida por si mesmo (Khatri et al., 2020). Existem três tipos principais de *machine learning*, entre elas:

1. **Aprendizagem supervisionada (*Supervised learning*):** é o tipo mais comum de *machine learning* onde os dados de treino são rotulados, isto é, os dados são acompanhados por uma resposta onde o algoritmo gera uma função que mapeia as entradas e as saídas pretendidas. Uma das tarefas mais comuns desta aprendizagem recai sobre o problema de classificação, onde o objetivo é induzir com que o computador aprenda um sistema de classificação que o utilizador cria. É também uma técnica muito utilizada para treinar *neural networks* e *decision trees*, pois ambas estas técnicas são bastante dependentes das informações fornecidas pelas classificações pré-determinadas. No caso das NN, a classificação é usada para determinar o erro da rede e, em seguida, realizar o respetivo ajuste. Já nas *decision tree*, as classificações são utilizadas para determinar quais os atributos que fornecem mais informação (Taiwo Oladipupo Ayodele, 2010);

2. **Aprendizagem não supervisionada (*Unsupervised learning*):** é um tipo de *machine learning* em que os dados de treino não são rotulados, invés disso o sistema descobre padrões e relações nos dados por conta própria. O objetivo da aprendizagem não supervisionada é incentivar o sistema ou computador a educar-se a fazer algo que ainda não sabe nem tem informações precedentes que o ajude. Existem duas abordagens bastante comuns para este tipo de aprendizagem sendo que a primeira é ensinar o agente a não dar categorizações explícitas, mas sim usar um tipo de sistema de recompensa que indique o sucesso da operação. Este é um tipo de treino que encaixa no problema de decisão, onde o objetivo não é produzir uma classificação, mas sim tomar decisões que maximizem as recompensas. O segundo tipo de aprendizagem não supervisionada é o *clustering,* onde o objetivo não passa por maximizar a função, mas sim por agrupar os dados em clusters, tendo por base as semelhanças e as diferenças dos dados (Taiwo Oladipupo Ayodele, 2010). A redução da dimensionalidade (*dimensionality reduction*) é também uma técnica de aprendizagem não supervisionada onde o objetivo é reduzir o número de recursos (*features*) ou dimensão dos dados, mantendo o máximo de informação possível;

3. **Aprendizagem por reforço (*Reinforcement learning*):** tal como o nome indica, é um tipo de aprendizagem em que um agente, animal ou máquina, é treinado para realizar ações específicas em resposta de estímulos específicos de maneira a maximizar uma recompensa. O agente aprende através da tentativa e erro,

tentando diferentes abordagens e ações e observando as consequências de cada uma delas. Por fim, as ações que resultam em recompensas maiores são reforçadas enquanto que aquelas que resultam em recompensas menores são descontinuadas. É um tipo de aprendizagem bastante utlizado na área da inteligência artificial, mas também em campos como a psicologia e a economia. Tal como (Taiwo Oladipupo Ayodele, 2010) refere, o algoritmo aprende uma política de como agir dada uma observação do mundo e toda a ação tem algum impacto no ambiente, que fornece feedback de maneira a orientar a aprendizagem do algoritmo;

2.3.1 *Machine learning* em deteção de fraude

Os algoritmos de *machine learning* são utilizados sempre que é necessário prever ou detetar algo. Assim, estes algoritmos podem ser utilizados em sistemas de deteção de fraude de várias maneiras e um exemplo disso é treinar o modelo para detetar padrões anormais ou suspeitos em transações financeiras (Zhou et al., 2019). Isso pode ser feito com base em dados de transações fidedignas e fraudulentas, onde o modelo aprende a distinguir o que é fidedigno e o que é suspeito de fraude com base em características como o valor de transação, a localização geográfica, o horário e outras informações relevantes. Depois de treinado, o modelo pode ser usado para classificar novas transações como fidedignas ou suspeitas de fraude. Outra maneira de utilizar o *machine learning* em sistemas de deteção de fraude é treinar um modelo para prever o risco de fraude em novas transações (Mathew et al., 2022). Nesse caso, o modelo pode ser treinado com base em dados de transações passadas, onde o risco de fraude foi previamente identificado. O modelo aprende a detetar padrões e características que estão associados a um risco elevado de fraude, e pode, então, avaliar o risco de fraude em novas transações (Zhou et al., 2019).

O *machine learning* pode também ser utilizado em conjunto com outras técnicas de deteção de fraude, como a análise de regras e a deteção com base em comportamentos, de maneira a aumentar a eficácia do sistema de deteção de fraude (Abhirami et al., 2021).

A deteção de fraude através de machine learning é possível devido ao poder dos algoritmos de *machine learning*. Estes algoritmos permitem que os dados sejam tratados e processados a uma velocidade nunca alcançável por seres humanos (Mathew et al., 2022). Algumas técnicas de *machine learning* que podem ser usadas para a deteção de fraude são as seguintes:

- **Regressão logística (*Logistic Regression*)**

A regressão logística, ou *logistic regression* em inglês, é uma técnica de análise de dados que é usada para prever a probabilidade de ocorrência de um determinado evento. O resultado esperado é uma variável binária que pode assumir apenas dois valores, como um “sim” e “não”,

ou “verdadeiro” e “falso”, ou até mesmo “0” e “1”. O objetivo do LR é prever a probabilidade de um evento ocorrer com base numa série de variáveis independentes. De maneira a obter estes resultados, a LR usa uma equação matemática que tem em conta os valores das variáveis independentes e produz um resultado que é transformado numa probabilidade (Negi et al., 2022). A partir desta probabilidade, é possível determinar se um evento é mais ou menos provável de acontecer. É assim um método de classificação supervisionado que retorna a probabilidade de um conjunto de variáveis binárias a partir de uma variável independente (Mathew et al., 2022).

Este modelo pode ser usado tanto para classificação como para regressão, apesar de a classificação ser a utilização mais típica. Embora os conceitos de “regressão” e “classificação” sejam distintos e incompatíveis, o foco do LR está na palavra “logística” que se refere à função logística que realiza a operação de classificação no algoritmo (Zou et al., 2019). Como consequência, obtém-se um binário que representa uma das classes. Também através da imposição de uma regra de decisão, um modelo de regressão de previsão de probabilidade pode ser utilizado como parte de um classificador que com ajuda de variáveis dependentes é possível prever o resultado. O princípio do LR e como funciona é explicado na equação (1), representa em baixo, onde a0, a1 a an são os coeficientes, x1, x2 a xn são as variáveis independentes e o resultado da probabilidade é p (Negi et al., 2022):

$$p = \frac{1}{1 + e^{-(a0 + a1x1 + a2x2 + \cdots + anxn)}} \quad (1)$$

- **Floresta aleatória (*Random Forest*)**

A floresta aleatória, ou *random forest* em inglês, é um tipo de modelo de *machine learning* com base em *decision trees*. Esta técnica consiste num grande número de DT treinadas de forma independente e aplicadas de forma paralela. Cada árvore na floresta é treinada usando uma amostra aleatória dos dados de treino e um subconjunto aleatório de recursos para tomar as decisões onde, depois disso, as previsões de cada árvore são combinadas de maneira a obter uma previsão final. É um modelo com enorme capacidade de lidar com problemas de classificação e regressão, tal como o LR (Rodrigues et al., 2022). É assim um mecanismo de aprendizagem supervisionada que recolhe previsões, em vez de se basear numa única DT, obtêm a maioria das previsões e prevê o resultado final. Podemos assim inferir que, quanto mais árvores tivermos, mais exatos e precisos serão os resultados (Gracia et al., 2021).

(Mathew et al., 2022) afirma que o RF está entre as tecnologias utilizadas unicamente para melhorar ainda mais a sua prosperidade em termos de exatidão nos algoritmos de *machine learning*.

- **Máquina de vetores de suporte (*Support Vector Machine*)**

A técnica máquina de vetores de suporte (SVM), ou *support vector machine* em inglês, é um algoritmo de aprendizagem supervisionada usada para a classificação e regressão, que analisa os dados encontrando o hiperplano que maximiza a margem entre as duas classes. Para conjuntos de dados que são separáveis linearmente, uma máquina de vetores de suporte tem a capacidade para conseguir uma boa classificação encontrando o hiperplano ótimo (Anowar & Sadaoui, 2020). Já em conjunto de dados não lineares, uma SVM pode mapear o espaço de características original para algum espaço de características de dimensão superior de maneira a resolver o problema de classificação (Zhou et al., 2019).

A ideia principal de uma SVM é então desenvolver um hiperplano como um plano de decisão que maximiza a separação entre o modo positivo e o modo negativo, no entanto pode ser um algoritmo lento para treinar em conjunto de dados de grande dimensão e pode ser sensível ao ruído nos dados de treino (Mathew et al., 2022).

- **K-Vizinhos próximos (*K-Nearest neighbours*)**

O K-Vizinhos próximos, ou *K-Nearest neighbours* em inglês, é um algoritmo de aprendizagem supervisionada utilizado em tarefas de classificação e regressão. É um algoritmo que tem como objetivo calcular a distância entre todos os registos e considerar de forma justa todos os parâmetros interligados de maneira a fornecer classificações eficientes, encontrando os vizinhos mais próximos de cada registo no conjunto de dados. Para identificar os k-vizinhos mais próximos de cada registo, cada classificação é dada limites a cada classe onde cada classificação é considerada tendo em conta a maior semelhança e proximidade entre os seus vizinhos. O número *k* representa a quantidade de vizinhos mais próximos necessários de cada registo (Almohaimeed & Gampa, 2019).

É um algoritmo simples e fácil de implementar, mas pode ser lento para classificar exemplos novos em conjunto de dados de grande dimensão (Almohaimeed & Gampa, 2019).

- **Árvore de decisão (*Decision Tree*)**

A árvore de decisão, ou *decision tree* em inglês, é um método de aprendizagem supervisionada utilizada em problemas de classificação. É formada por um conjunto de nós de decisão que criam um caminho lógico para chegar a uma determinada conclusão. Ou seja, cada nó de decisão representa uma decisão que precisa de ser tomada com base num conjunto de características ou atributos, enquanto que as folhas representam as conclusões finais da DT (Mustaqim et al., 2021). Com base na probabilidade de acontecimentos que já ocorreram, o

objetivo de uma DT é prever e selecionar a melhor solução possível, comparando todas as soluções que são avaliadas com o cálculo da probabilidade e o gráfico em árvore (Zhou et al., 2019).

- **Redes Neuronais Convolucionais (*Convolutional Neural Network*)**

As redes neuronais convolucionais (CNN), ou *Convolutional Neural Network* em inglês, são um tipo especializado de arquitetura *feed-forward* de redes neuronais profundas. A ideia principal de uma CNN é a aplicação de filtros convolucionais em diferentes partes dos dados de entrada, permitindo a extração de caraterísticas relevantes de forma hierárquica. Esses filtros, também conhecidos como *kernels*, são pequenas janelas que percorrem a entrada e são multiplicados ponto a ponto com valores de entrada em cada localização(T. T. Nguyen et al., 2019).

As CNN tornaram-se algoritmos de última geração para visão computacional, processamento de linguagem natural e problemas de reconhecimento de padrões. Para além disso, são altamente eficazes na extração automática de características relevantes dos dados de entrada, permitindo que as redes neuronais aprendam representações hierárquicas e obtenham um desempenho superior em várias tarefas de análise de imagem (Ghosh et al., 2020).

- **Memória de longo prazo (*Long Short-Term Memory*)**

A rede neural de memória a longo prazo (LSTM), ou *Long Short-Term Memory* em inglês, é um tipo especial de rede neural recorrente, do inglês *Recurrent Neural Network* (RNN), que alcançou excelente desempenho na aprendizagem de dependências a longo prazo e evita o problema do desaparecimento de gradiente (Wang et al., 2020).

Este algoritmo consiste numa célula de memória que guarda informações anteriores e serve como memória para escrever, ler e apagar informações (Venna et al., 2019). Cada célula de memória possui três componentes principais: um portão de entrada (*input gate*), um portão de esquecimento (*forget gate*) e um portão de saída (*output gate*). O portão de entrada controla que as informações de entrada serão guardadas na célula de memória, o portão de esquecimento é responsável por decidir que informações antigas da célula de memória devem ser descartadas e, por último, o portão de saída determina quais as informações da célula de memória atual devem ser usadas como saída da célula de memória. Estes três portões trabalham em conjunto para controlar o fluxo de informações na célula de memória ao longo do tempo. (Esenogho et al., 2022).

Os modelos e algoritmos supramencionados permite com que os sistemas de deteção de fraude sejam rápidos, escaláveis, eficientes e precisos no que diz respeito a deteção de fraude (Mathew et al., 2022):

- **Rapidez**: Quando se trata de detetar fraudes os resultados necessitam de ser rápidos. Com técnicas de *machine learning* é possível detetar qualquer atividade fraudulenta em milissegundos. No caso de detetar qualquer atividade de fraude, o sistema bloqueia automaticamente esse pagamento específico.

- **Escalabilidade**: No mundo atual, todas as empresas pretendem aumentar o seu volume de negócio e transações e, como consequência, mais dados terão de ser processados e guardados. Isso será benéfico, pois quanto mais dados, melhor será o resultado obtido pelos modelos de *machine learning*.

- **Eficiência**: O custo das tarefas de classificação ou previsão seria bastante elevando se fosse executado por um ser humano, enquanto que através de técnicas de *machine learning* o custo é apenas o funcionamento dos servidores, o que é bastante inferior. As técnicas de *machine learning* permitem fazer o trabalho monótono de análise de informação num curto espaço de tempo. Ao contrário dos humanos, as máquinas são capazes de realizar tarefas repetitivas e aborrecidas 24 horas por dia e só precisam de comunicar decisões a um humano se forem específicas.

- **Precisão**: As técnicas de machine learning podem ser muito eficazes quando comparado com os seres humanos na procura de padrões indefinidos e pouco comuns que podem levar a compras fraudulentas no futuro. Os modelos de machine learning são bastante rápidos e precisos em detetar este tipo de comportamentos. Isto é benéfico para detetar clientes contrabandistas, mesmo que não tenha sido efetuada nenhuma compra ou devolução.

De maneira a sintetizar e sumariar os métodos de *machine learning* em cima mencionados foi elaborada a Tabela 7, na qual estão descritas as vantagens e limitações das técnicas mencionadas (Bhatt & Meniya, 2022).

Método	Vantagens	Limitações
Logistic regression	Fácil de implementar, fácil de interpretar, bastante eficiente para treinar e rápido para classificar registos desconhecidos.	A principal limitação é a linearidade entre a variável dependente e independente.
Random Forest	Diminui o problema de overfitting e ajuda a melhorar precisão. É versátil para cada problema quer de classificação quer de regressão. Automatiza os valores ausentes e desconhecidos no conjunto de dados.	A principal limitação é que um intervalo grande de árvores tornará o algoritmo lento e menos eficaz para certos períodos. São algoritmos rápidos para treinar, mas lentos para formas previsões depois de treinados.
Support Vector Machines	É mais simples em espaços de alta dimensão. É eficaz nos casos em que o número de dimensões é maior que o número de amostras. É eficiente em termos de memória.	A principal limitação é não ser apropriado a conjunto de dados de grande dimensão. Não funciona tão bem quando existe ruído adicional.
K-nearest neighbours	É bastante simples de implementar uma vez que necessita apenas de dois parâmetros, um valor de *k* e a distância percorrida.	A principal desvantagem é que com grandes quantidades de dados a previsão pode ser lenta. Precisa de muita memória para armazenar todos os dados de treino e no caso de armazenar todos os dados de treino, pode ser computacionalmente caro.
Decision Tree	Os outputs são simples de interpretar sem a necessidade de conhecimento estatístico. São simples de preparar e requer menos limpeza de dados.	A principal limitação é ser instável. Ou seja, uma pequena alteração no conhecimento resultará numa grande mudança na estrutura da DT.
Convolutional Neural Network	Capazes de aprender automaticamente as características relevantes dos dados de entrada e de lidar com variações de	Necessidade de dados rotulados em grande quantidade senão não é eficaz. Pode também exigir muito computacionalmente

	posição e escala nos dados de entrada.	para treinar e executar modelos de CNNs.
Long Short-Term Memory	Capturam dependências a longo prazo. Lidam com diferentes tipos de sequências, incluindo séries temporais, dados textuais e outras formas de dados sequenciais. Permitem um controlo mais preciso do fluxo de informações nas células de memória.	São bastante complexas em termos de arquitetura e, por isso, exigem mais recursos computacionais. Requerem um conjunto de dados de treino grandes o suficiente para capturar padrões significativos.

Tabela 7 – Tabela de comparação entre os diferentes métodos de *machine learning*, com base em (Bhatt & Meniya, 2022), (Ghosh et al., 2020), (T. T. Nguyen et al., 2019) , (Wang et al., 2020) e (Esenogho et al., 2022)

2.4 Sistemas de deteção de fraude

Mathew, Nithya, Vishwanatha, Shetty, Priya e Kavya, (2022), propõem uma metodologia para classificação de transações fraudulentas utilizando algoritmos de machine learning para a deteção de fraudes em conjunto de dados com cartões de crédito. Os autores acreditam que é muito importante os setores financeiros, bancos e empresários minimizem as perdas que podem ocorrer devidos aos contrabandistas, através da implementação de várias metodologias. Utilizaram algoritmos de aprendizagem supervisionada como a *logistic regression* (LR), *random forest* (RF), *Naive-Bayes classifier* (NBC), *Support Vector Machines* (SVM), k-*nearest neighbours* (KNN) e *decision tree* (DT) para avaliarem o desempenho na deteção de fraudes financeiras visto serem as técnicas mais eficientes (Chen & Lai, 2021; Lim et al., 2021).

Com base na taxa de exatidão, foram feitas previsões sobre qual técnica seria mais benéfica de implementar de forma a reduzir o número de tentativas de fraude e puderam perceber que, em alguns casos, usar o recurso “tempo” permite obter uma taxa de precisão melhor. Os algoritmos de LR, RF e DT obtiveram os melhores resultados em comparação com as outras técnicas e o classificador do LR teve melhor desempenho com conjunto de dados sem dados de amostra do que em dois conjuntos de dados com dados de amostras assim como um modelo híbrido de *machine learning* ajuda a obter resultados mais precisos do que a implementação de modelos individuais (Nithya & Ilango, 2020a, 2020b).

O conjunto de dados utilizado contem 284807 transações efetuadas por titulares de cartões de crédito europeus onde 492 transações são fraudulentas. A Figura 5 representa o fluxo dos algoritmos aplicados para identificar as transações fraudulentas.

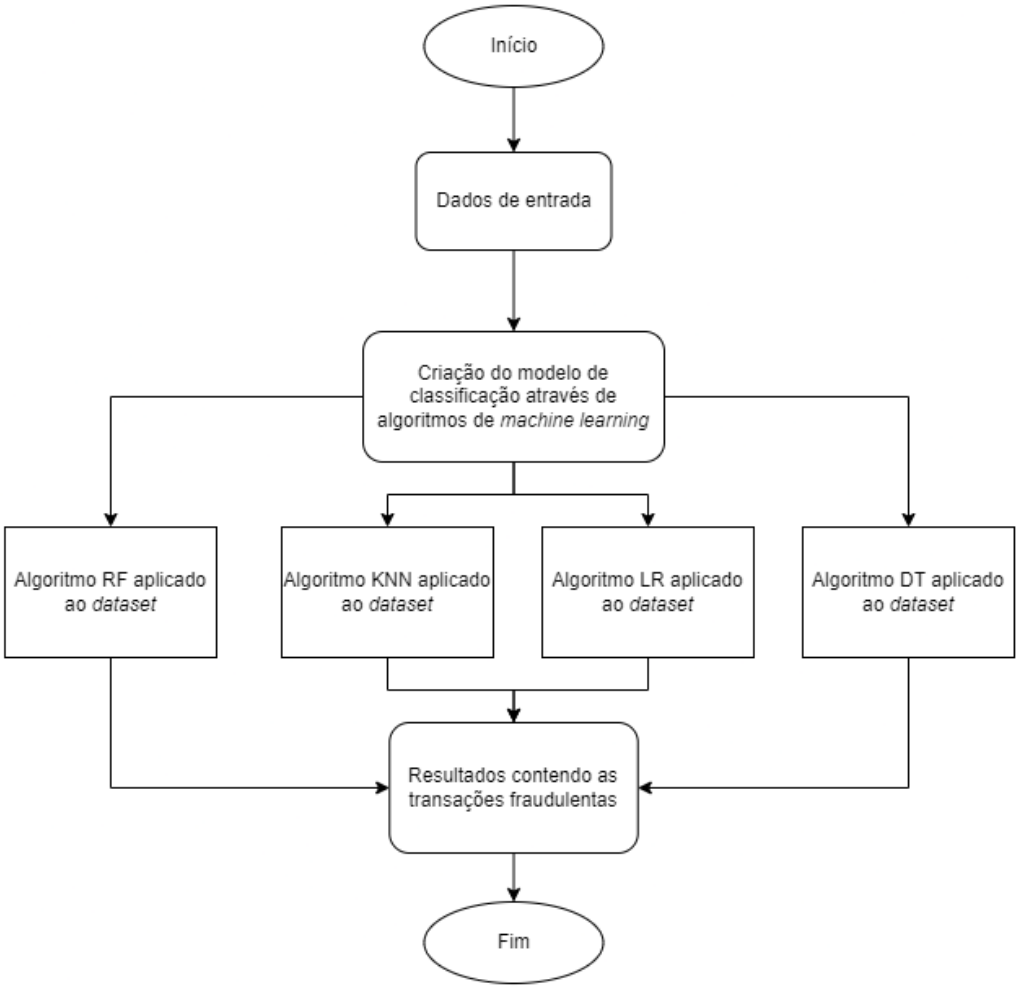

Figura 5 – Fluxograma do mecanismo implementado em (Mathew et al., 2022)

A Tabela 8 apresenta os resultados obtidos pelos autores no cálculo da exatidão (*accuracy*), precisão (*precision*), revocação (*recall*) e pontuação f-1 (*f-1 score*) dos vários algoritmos aplicados.

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
DT	0.99	0.73	0.82	0.77
RF	0.99	0.94	0.82	0.87
KNN	0.94	0.95	0.95	0.95
LR	0.99	0.86	0.57	0.68

Tabela 8 – Comparação da performance dos algoritmos em (Mathew et al., 2022)

O método RF demonstrou maior exatidão com uma pontuação f-1 bastante eficiente. O RF e LR também demonstraram bons resultados em termos de exatidão. Já o KNN, como se pode verificar, obteve o resultado mais baixo quanto a esta métrica. Em termos de precisão, o RF e o KNN foram os modelos mais precisos e quanto a revocação e pontuação f-1, o KNN foi o modelo com a pontuação mais elevada. A termo de conclusão, Mathew e os restantes autores, (2022), afirmam que o método do RF e DT obtiveram, no geral, melhores resultados.

Mathew e os restantes autores, (2022), afirmam também que os únicos inconvenientes encontrados durante a implementação foi ter de usar um conjunto de dados de grande escala de maneira a melhorar a precisão do estudo, pois quando implementados com um conjunto de dados mais pequeno, os resultados não eram tão exatos nem fiáveis.

Por sua vez, Abhirami, Pani, Manohar e Kumar (2021) realizaram um estudo comparativo entre vários algoritmos de aprendizagem automática com o objetivo de identificar as melhores técnicas para a deteção de transações fraudulentas. Os autores acreditam que ao utilizar uma abordagem de *machine learning* é a forma mais eficaz de detetar e prever de forma eficaz transações fraudulentas, sem utilizar instruções específicas. A fim de produzir uma previsão, um algoritmo de *machine learning* cria um modelo matemático com base em amostras de dados.

Tal como Abhirami e os restantes autores (2021) mencionam, é difícil detetar e prevenir fraudes durante transações *online*, mas a implementação de algoritmos de *machine learning* como o *random forest*, *decision tree* e a abordagem *XGBOOST* facilitam este processo. O RF supera a DT e as abordagens *XGBOOST* quando a precisão é comparada (Jain et al., 2020). No entanto, nesta situação, a técnica de DT é mais precisa e exata e fornece melhores resultados em comparação com o modelo de RF. O objetivo da análise de dados é encontrar padrões e utilizá-los para tomar melhores decisões em várias situações que possam ocorrer (Dhankhad et al., 2018).

O conjunto de dados utilizado para o estudo realizado por Abhirami e os restantes autores, (2021), contém 284807 transações, onde 492 são fraudes. A diferença das transações é identificada por um parâmetro, denominado "*target"*, onde 1 indica que a transação é fraudulenta e 0 indica que a transação é legítima. Além disso, o sistema foi implementado no *Anaconda Jupyter Notebook*, uma aplicação de código aberto que permite criar e editar documentos, na linguagem *Python*. Após o treino, validação e pré-processamento do conjunto de dados, Abhirami e os restantes autores, (2021) recorreram a cinco modelos com o objetivo de obter o modelo de melhor desempenho que, por sua vez, será o modelo ideal para a previsão. As métricas de performance utilizadas para avaliação foram a precisão, exatidão e revocação. Por fim, as utilizaram as técnicas de classificação de DT, LR, KNN, SVM e RF. Estes modelos foram aplicados ao conjunto de dados em cima mencionado antes e depois do pré-processamento e os resultados obtidos encontram-se representados nas Tabela 9 e Tabela 10 e Figura 6 e Figura 7.

	Métricas de performance			
	Exatidão	Precisão	Revocação	Tempo de execução (seg)
DT	0.99939	0.99929	0.99932	0.04981
KNN	0.99825	0.99840	0.99840	0.08127
LR	0.99915	0.99891	0.99860	0.85735
SVM	0.999816	0.99665	0.99832	0.03110
RF	0.99926	0.99927	0.99931	0.32989

Tabela 9 – Comparação da performance dos algoritmos antes do pré-processamento em (Abhirami et al., 2021)

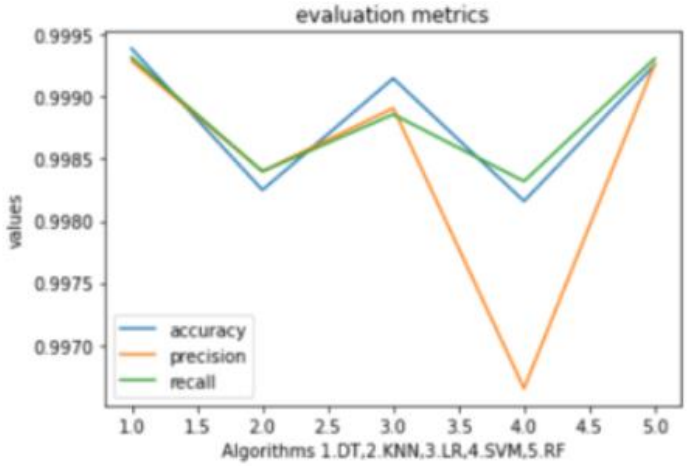

Figura 6 – Gráfico de comparação da performance dos algoritmos antes do pré-processamento em (Abhirami et al., 2021)

	Métricas de performance			
	Exatidão	Precisão	Revocação	Tempo de execução (seg)
DT	0.92385	0.89583	0.94505	0.04981
KNN	0.94416	0.96551	0.92307	0.08127
LR	0.96446	0.98837	0.95604	0.85736
SVM	0.93908	0.97647	0.91208	0.03110
RF	0.92893	0.95505	0.93406	0.32989

Tabela 10 - Comparação da performance dos algoritmos após pré-processamento em (Abhirami et al., 2021)

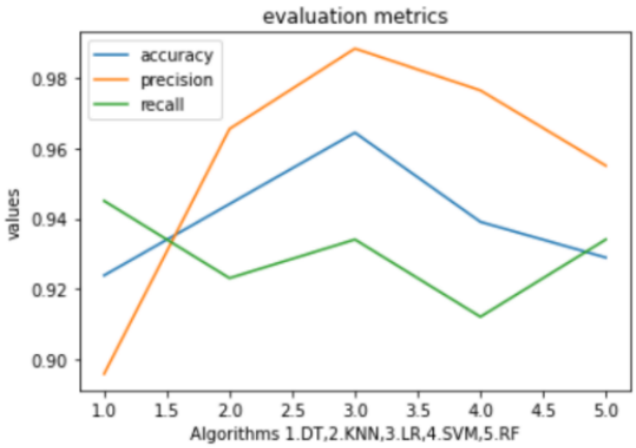

Figura 7 - Gráfico de comparação da performance dos algoritmos após pré-processamento em (Abhirami et al., 2021)

O pré-processamento e subamostragem (*undersampling*) foram necessários para resolver o problema de não ser possível inferir nenhuma conclusão com os resultados antes do pré-processamento, visto os valores serem todos muito próximos. Foi removida a coluna "tempo" do conjunto de dados e à coluna "quantidade" foi feita uma normalização. Através dos resultados obtidos, concluem que o método LR obteve melhores resultados e, por sua vez, melhor performance e, em contrapartida, o método DT obteve os piores resultados e teve então a pior performance dos cinco modelos aplicados.

Em termos de limitações, um conjunto de dados com maior quantidade de dados traria resultados ainda mais precisos e eficazes. Além disso, o conjunto de dados escolhido já se encontrava pré-processado, pelo que Abhirami, Pani, Manohar e Kumar, (2021) foram obrigados a fazer um novo pré-processamento dos dados, de maneira a inferir qualquer tipo de conclusões.

2.4.1 Sistemas de deteção de fraude com base em padrões comportamentais

Raja, Raman e Ushakiruthika, (2021) propuseram um sistema dividido em quatro fases para detetar atividades fraudulentas. As deteções de atividades fraudulentas podem ser classificadas em duas categorias: deteção de anomalias e deteção com base em classificadores. As anomalias são detetadas ao determinar a variação entre a transação corrente e o perfil do utilizador (Brzeziński, 2010). Qualquer transação que seja inconsistente com as transações regulares do utilizador é considerado uma anomalia. Já a deteção com base em classificadores, é utilizada a aprendizagem supervisionada para ensinar os classificadores apropriados a recolher as transações que envolvam tanto transações fidedignas como transações fraudulentas (Q. Liu et al., 2018).

A abordagem sugerida extrai o padrão comportamental dos dados recolhidos dos utilizadores e identifica cada um através de um método de grupamento. Assim, a abordagem molda o padrão de comportamento transacional de cada utilizador através de padrões de comportamento organizados e está dividida em quatro fases (Raja et al., 2021):

1. Criação de um PIN para um esquema de *feedback* seguro: após a solicitação e envio do cartão de crédito a um titular, este tem de criar um PIN que é considerado estático e é necessário durante o processo de feedback e conclusão de uma transação, se esta for detetada como fraudulenta. Este PIN, juntamente com um PIN criado por um servidor enviado para o utilizador por notificação, que os autores propuseram, é usado para processos de verificação durante a deteção de fraude;

2. Criação de padrões comportamentais: foram criados padrões comportamentais de vários utilizadores como por exemplo a hora que a transação foi efetuada, a quantia, local e a categoria da compra;

3. Classificação das transações como fraudulentas ou genuínas: através da técnica do RF, todas as transações foram classificadas como fraude ou genuínas. Através do RF, agruparam as transações atuais com as passadas, com base em classificadores. As condições de agrupamento destes classificadores foram com base na categoria do produto (restauração, hospital, lazer, etc.), localização, montante da transação e tempo da transação. Após o processo de classificação, se algum dos classificadores classificar a transação como fraude, o sistema vai para o nível seguinte de autenticação, que é o mecanismo de *feedback*, explicado de seguida;

4. Atualização do perfil comportamental dos utilizadores através do mecanismo de *feedback*: este mecanismo que utiliza uma combinação dos dois PIN's para autenticar o utilizador, atualiza os detalhes da transação atual nos registos e atualiza o perfil de comportamento do utilizador. Assim que o titular insere o PIN e se tudo estiver correto, a transação acontece. Se as credenciais estiverem incorretas, a transação é parada.

Para testarem a abordagem proposta e a respetiva eficácia, (Raja et al., 2021) utilizaram as métricas de catalogação dupla como o Verdadeiro-Positivo (TP), Falso-Positivo (FP), Falso-Negativo (FN) e Verdadeiro-Negativo (TN), onde os TP representam o número de transações ilegais previstas corretamente como ilegais enquanto o FP representa o número de transações legais previstas incorretamente. Por outro lado, os FN representam o número de transações ilegais que foram previstas como legais enquanto que os FP representam o número de transações legais previstas como ilegais. Os autores calcularam a exatidão e revocação do modelo através de três métodos diferentes: *random forest* com o mecanismo de feedback (RF

+ FB), cadeia de Markov (MC) e RF. Os resultados obtidos encontram-se representados nas Figura 8 e Figura 9.

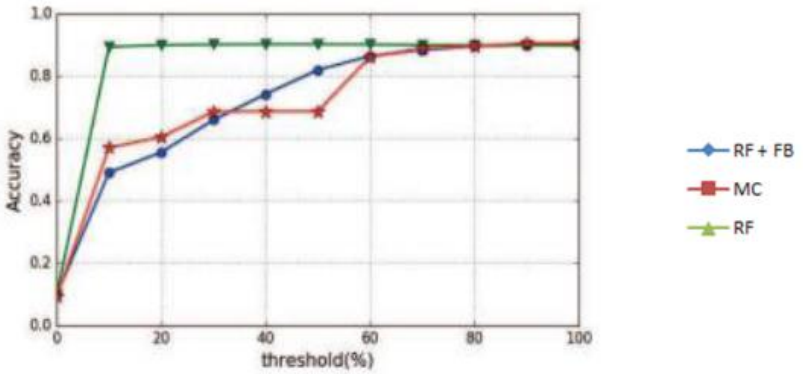

Figura 8 – Comparação da exatidão em (Raja et al., 2021)

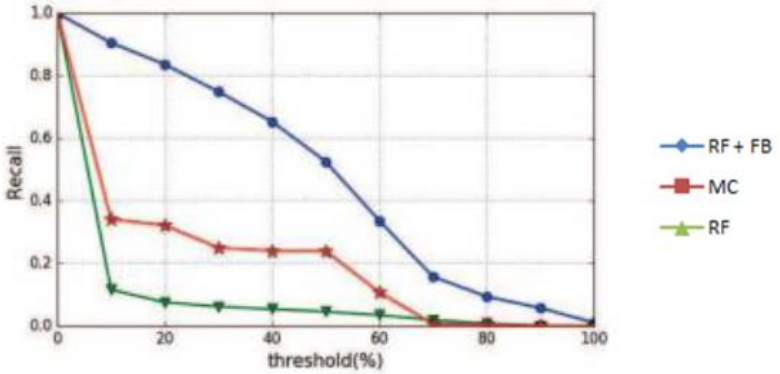

Figura 9 – Comparação da revocação em (Raja et al., 2021)

Através dos gráficos, Raja e os restantes autores, (2021) concluem que a exatidão do modelo RF é melhor do que os outros dois esquemas, enquanto que em termos de revocação é pior. Já o modelo RF + FB combinados obtém bons resultados. Relativamente a limitações, os autores consideram que podem ser adicionadas mais restrições ao algoritmo para obter melhores resultados e melhor exatidão em trabalhos futuros.

2.4.1.1 *Big Data Mining*

O estudo realizado por JiaoLong Li, (2022), tem também o objetivo de identificar fraudes no e-commerce e diminuir o risco financeiro de empresas através de *Big Data Mining* (BDM). BDM permite extrair dados desconhecidos ou "ocultos" de inúmeros dados incompletos, ruidosos ou aleatórios que são potencialmente úteis (Triguero et al., 2019; Yan, 2020) . Para isso, o autor desenvolveu um modelo de deteção de fraude no e-commerce baseado na fusão de tecnologias de informação, que envolve inteligência artificial e *data mining*, e comparou-o

com modelos de *machine learning* já existentes como o SVM e LR. O modelo proposto por JiaoLong Li, (2022) (Figura 10), coleta inicialmente as informações do utilizador, desde transações efetuadas a comportamentos habituais do mesmo e envia para o módulo de análise. Esta informação é depois enviada para o motor de correspondência dentro do módulo de deteção de fraude que, através de técnicas de *data mining*, cria os resultados das correspondências. De seguida, o resultado é enviado para o módulo de análise de fraude. Finalmente os resultados das correspondências de saída formam comportamentos de fraude específicos que serão incluídos na plataforma de anti-fraude de forma a prevenir e prever esses comportamentos.

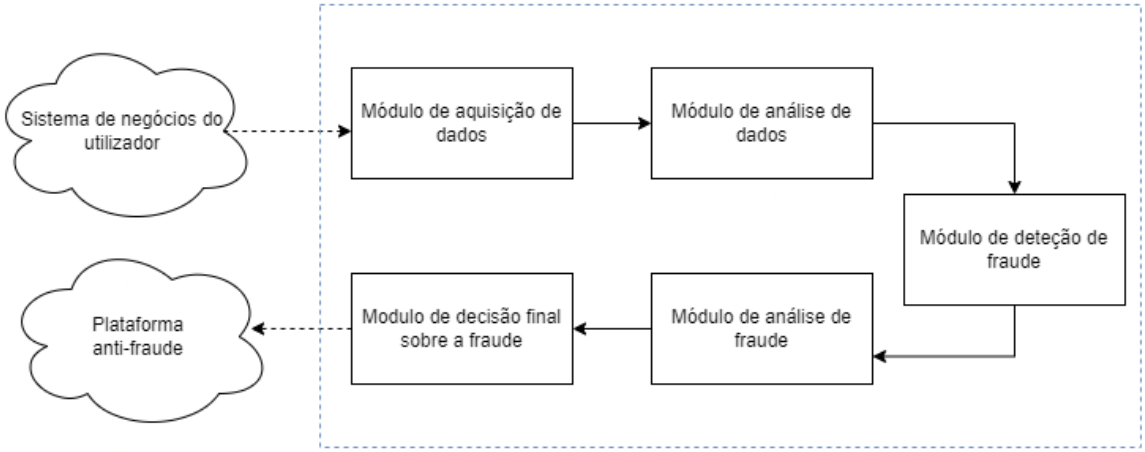

Figura 10 – Modelo de deteção de fraude em (Li, 2022)

De forma a testar o modelo proposto, JiaoLong Li, (2022), utilizou um conjunto de dados com 30000 exemplos de comportamentos fraudulentos e não fraudulentos. A Figura 11 analisa a exatidão do modelo em amostras de testes e de treino. Foi possível concluir que com o aumento da quantidade de dados, a exatidão também aumenta. A exatidão da amostra de teste, 84,1%, é também ligeiramente superior à de treino, 75,2%.

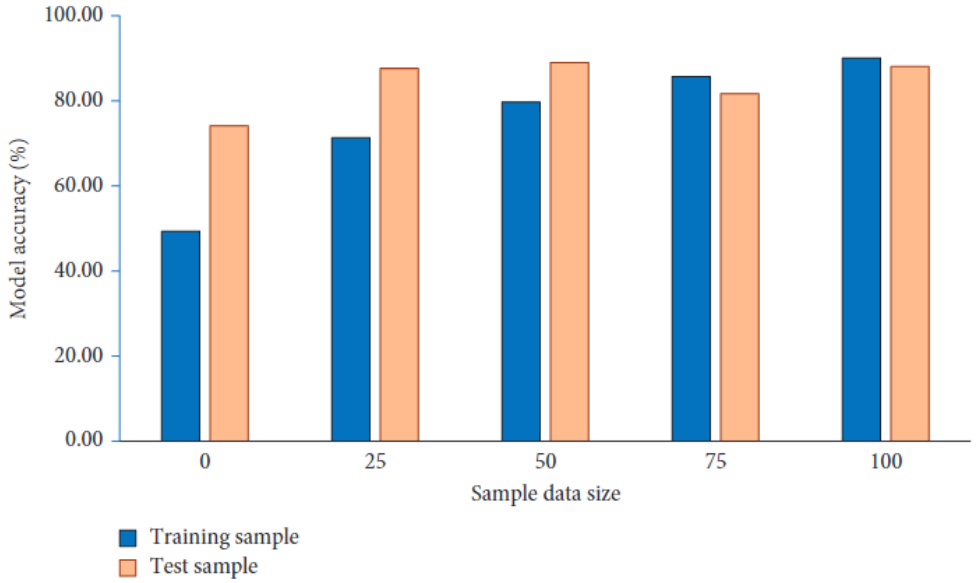

Figura 11 – Exatidão do modelo de JiaoLong Li, (2022), em amostras de treino e teste

JiaoLong Li, (2022), avaliou ainda a exatidão nas amostras de teste de um conjunto de dados com 1000 registos (Figura 12). Mais uma vez, com o aumento da quantidade de dados, a exatidão do modelo apresenta uma média de 89.41%. A Figura 13 analisa ainda o efeito de classificação de amostras de fraude no e-commerce em diferentes métodos de *machine learning*. É possível inferir que o modelo proposto pelo autor apresenta uma exatidão superior aos modelos de SVM e LR e pode assim processar e calcular possíveis comportamentos fraudulentos no e-commerce e obter maior precisão.

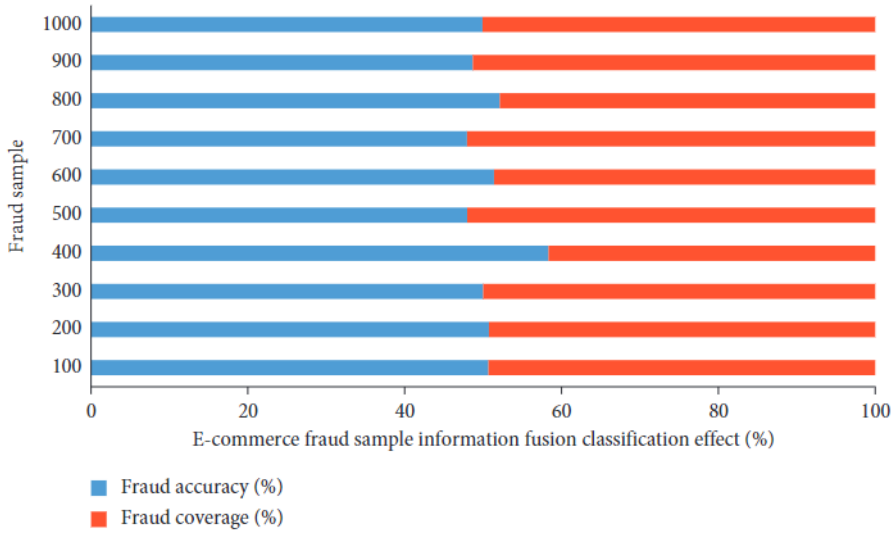

Figura 12 – Exatidão do modelo proposto nas amostras de teste em (Li, 2022) numa amostra de 1000 registos

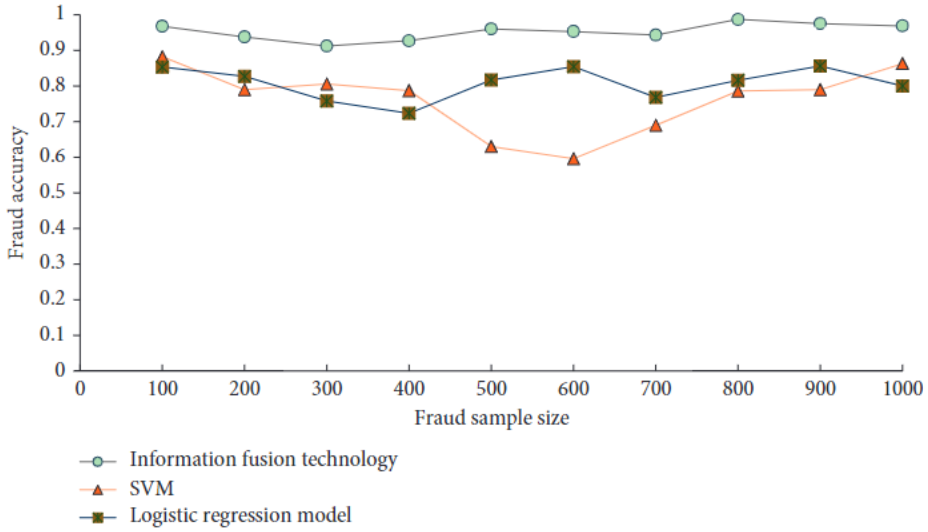

Figura 13 – Comparação da exatidão do modelo proposto em (Li, 2022) com outras técnicas de *machine learning*

Em suma, os resultados comprovam que o modelo proposto por JiaoLong Li, (2022), baseado na fusão de tecnologias de informação e que envolve inteligência artificial e *data mining,* apresenta uma precisão superior a métodos como o SVM e LR. A tecnologia de BDM permite também que a análise e classificação de comportamentos fraudulentas seja mais eficaz. Em termos de limitações, outros modelos de machine learning como o RF, DT e KNN podiam também ser implementados testados de forma a aumentar a taxa de comparações.

2.4.2 Sistema de deteção de fraude e pré-processamento com SMOTE

Com maior ênfase no tratamento de dados e a importância de o conjunto de dados ser balanceado, Saputra e Suharjito, (2019), elaboraram um estudo com o objetivo de comparar diferentes técnicas de *machine learning* e perceber a interferência da técnica de sobresamdição minoritária sintética (SMOTE). O SMOTE é uma técnica de *oversampling* que permite aumentar o número de casos no conjunto de dados de forma equilibrada pois aumenta o número de classes positivas até estes serem iguais às classes negativas (Sharma et al., 2018). O algoritmo encontra o k vizinho mais próximo para uma classe positiva e constrói um conjunto de cópias de casos minoritários existentes, combinando aleatoriamente as suas características e aumentado assim o número de casos minoritários (Kim et al., 2016).

O conjunto de dados utilizado por Saputra e Suharjito, (2019), foi retirado do *Kaggle* e contém 151112 registos dos quais 14151 correspondem a fraudes, o que torna assim o conjunto de dados desequilibrado. De forma a obterem um equilíbrio, Saputra e Suharjito, (2019), começaram por selecionar as *features* a serem usadas e procederam então ao pré-processamento através do PCA. A análise de componentes principais (PCA) é um procedimento

usado na compactação de dados, e também muito utilizado na extração de *features* em conjunto de dados de grande dimensão (Sadaghiyanfam & Kuntalp, 2018). Após o pré-processamento, o conjunto de dados foi balanceado através do SMOTE que permitiu estabelecer o equilíbrio (Figura 14 e Figura 15). Os algoritmos de RF, DT, NN e NBC foram utilizados para avaliar o desempenho na deteção de fraudes com base nas métricas de exatidão, precisão, revocação, *g-mean* e pontuação f-1.

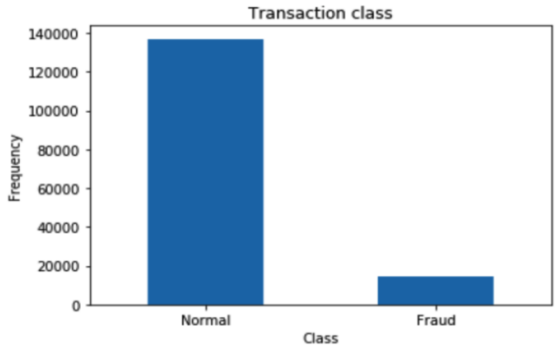

Figura 14 – Rácio do conjunto de dados antes do *oversampling* em (Saputra & Suharjito, 2019)

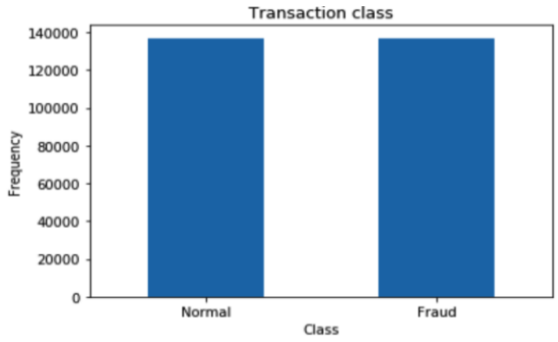

Figura 15 – Rácio do conjunto de dados depois do *oversampling* em (Saputra & Suharjito, 2019)

Alguns exemplos dos resultados obtidos por Saputra e Suharjito, (2019), encontram-se representados nas Figura 16, Figura 17 e Figura 18.

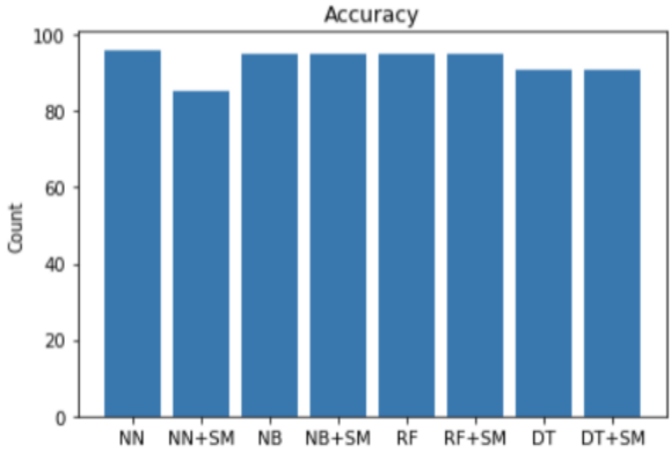

Figura 16 – Resultados da exatidão em (Saputra & Suharjito, 2019)

Com 96%, o algoritmo com maior exatidão foi o NN. Excluindo o caso de NN + SMOTE, todos obtiveram uma exatidão elevada pelo que não foi possível obter mais conclusões quanto à exatidão.

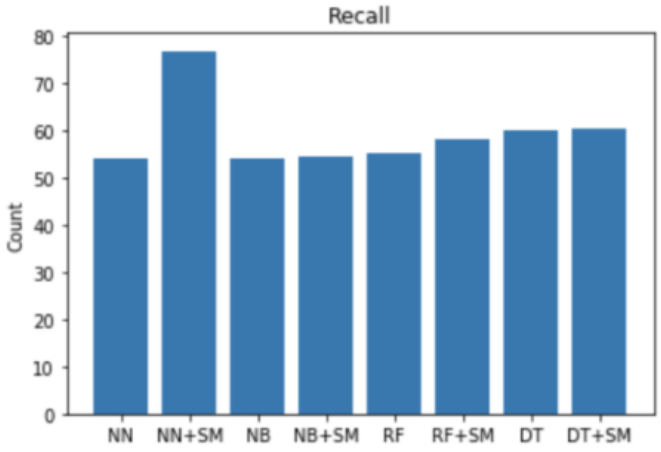

Figura 17 - Resultados da revocação em (Saputra & Suharjito, 2019)

O algoritmo do NN + SMOTE obteve um revocação mais elevado e este aumenta sempre que o algoritmo de *machine learning* é utilizado com o SMOTE. Com o NN permitiu aumentar a revocação de 54% para 76.7% e o RF de 55% para 58%.

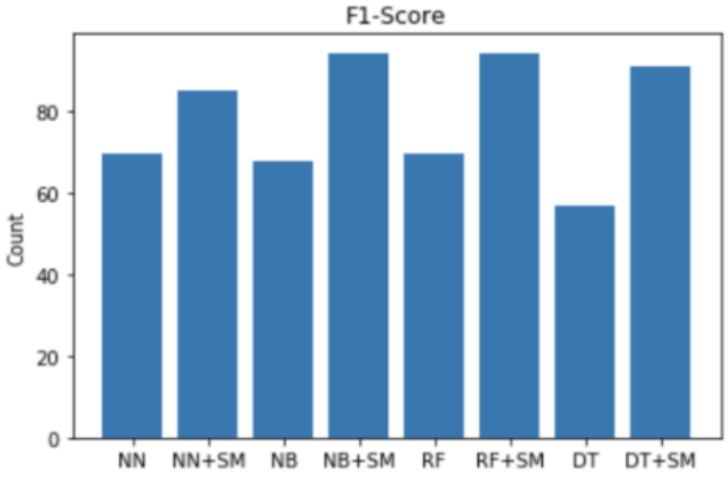

Figura 18 - Resultados da pontuação f-1 em (Saputra & Suharjito, 2019)

Com 94.5%, o modelo NB + SMOTE obteve a pontuação de f-1 mais elevada onde novamente a técnica SMOTE permitiu o aumento deste valor, ao elevar a percentagem de 67.9% para 94.5%. O SMOTE permitiu que todos os algoritmos obtivessem melhores resultados quanto à métrica f-1.

Em termo de conclusão, Saputra e Suharjito, (2019), inferem que a implementação do SMOTE é eficaz ao permitir lidar com o problema do desequilíbrio em conjunto de dados e assim aumentar o desempenho dos modelos de classificação. Quanto a limitações, os autores referem que deveriam ter implementado técnicas de *deep-learning* de forma a investigarem mais profundamente a interferência do SMOTE.

2.4.3 Sistema de deteção de fraude utilizando *deep-learning*

Utilizando uma abordagem ligeiramente diferente, Nguyen, Tahir, Abdelrazek e Babar (2020) afirmam que várias técnicas de *machine learning* têm sido usadas para detetar fraudes com cartões de crédito, mas que, até aos dias de hoje, nenhum sistema de deteção de fraudes conseguiu oferecer alta eficiência. Para isso, e de forma a combater o crescimento exponencial de fraudes, os autores afirmam que o desenvolvimento recente de *deep learning* é bastante promissor e vantajoso para resolver diversos problemas em áreas distintas.

Nguyen e os restantes autores, (2020), defendem que os algoritmos tradicionais de machine learning, como SVM, DR e LR foram extensivamente propostos para a deteção de fraudes em transações financeiras e que estes algoritmos não são muito adequados para grandes conjuntos de dados. A utilização de métodos de *deep-learning* como CNN (*Convolutional Neural Network*) e LSTM (*Long Term-Short Memory*) ainda são muito limitados e usados em problemas de classificação de imagens e processamento de linguagem natural, devido à sua capacidade de lidar com grandes conjuntos de dados. Os autores decidiram então explorar o desempenho destes dois métodos de *deep-learning* quando são aplicados a um

sistema de deteção de fraude, e como o desempenho do modelo é afetado em resposta ao pré processamento dos dados.

As CNN são um mecanismo de *deep learning* que estão associados a dados espaciais e que, à semelhança de ANN (*Artificial Neural Network*) possuem uma camada oculta, além de camadas de convolução especiais com um número diferente de canais em cada camada (T. T. Nguyen et al., 2019). Além disso, as CNN executam automaticamente a redução de recursos, o que a torna menos propensa a *overfitting* e a não requerer um pré-processamento de dados muito pesado. No estudo feito por Nguyen e os restantes autores, (2020), são utilizadas redes 2DCNN e 1DCNN para classificar os casos como fraude e não fraude. Redes 2DCNN e 1DCNN são arquiteturas de CNN que são projetadas para processar dados de diferentes dimensões, duas e uma, respetivamente. Além das CNN, os autores aplicaram também o algoritmo de LSTM para avaliar e comparar os dois modelos de *deep-learning* (T. Nguyen et al., 2020).

Nguyen e os restantes autores, (2020), utilizaram três conjuntos de dados diferentes para avaliar o desempenho do modelo, entre eles: *European Card Data* (ECD), *Small Card Data* (SCD) e *Tall Card Data* (TCD). Estes conjuntos apresentam um número de instâncias de fraude muito menor em comparação com as transações legítimas e, por isso, encontram-se bastante desequilibrados. O primeiro conjunto de dados, ECD, contém 284 807 transações com 31 *features* das quais 492 são fraude. O SCD contém apenas 3075 transações com 12 *features* e das quais 448 são fraude. Por fim, o TCD, contém 500 000 transações com 9 *features* das quais 28 0000 correspondem a transações fraudulentas. Os modelos foram avaliados segundo as métricas de performance exatidão, precisão, revocação e pontuação f-1.

Os conjuntos de dados necessitaram de ser equilibrados e para isso, Nguyen e os restantes autores, (2020), aplicaram as técnicas de *Random Under Sampler* (RUS), *Near Miss Sampling* (NM) e SMOTE nos três conjuntos, de maneira a avaliar o impacto e resultados de cada um. Para além disso, os autores transformaram as variáveis categóricas em numéricas no conjunto SCD e removeram a coluna "*Transaction Date*" por ter apenas valores nulos. No conjunto TCD removeram a coluna "*custID*" por ter apenas valores únicos e não adicionar informação relevante ao conjunto de dados. No final, aplicaram a normalização utilizando o *StandardScaler* para avaliar a performance dos modelos e dividiram os três conjuntos de dados em teste, treino e validação.

Na Figura 19 está representado o desempenho entre os três conjuntos de dados segundo a métrica pontuação f-1. Em (T. Nguyen et al., 2020) observaram que, em geral, o desempenho da validação diminui à medida que o tamanho dos conjuntos de dados aumenta. No conjunto ECD, o desempenho é estável, com pouca variação entre o desempenho de validação e teste. Os autores explicam que valores ausentes para o 2DCNN no SCD e TCD foram devido ao menor número de características no conjunto de dados, o que impossibilitou a criação de uma matriz de características. Já os valores ausentes RF e SVM no conjunto TCD aconteceram devido ao treino mais demorado que está associado a estes classificadores.

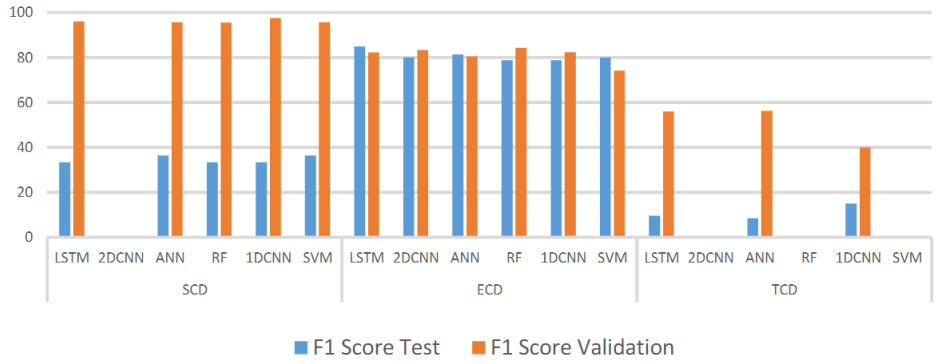

Figura 19 – Pontuação f-1 obtida no SCD, ECD e TCD em (T. Nguyen et al., 2020)

Já a Figura 20 demonstra os resultados obtidos no conjunto ECD quando aplicadas as técnicas de NM, RUS e SMOTE. É possível observar que o modelo obteve melhores resultados com a técnica SMOTE e, por isso, Nguyen e os restantes autores, (2020), decidiram comparar os resultados da aplicação do SMOTE no conjunto de teste com os resultados do conjunto de teste sem qualquer técnica de *oversampling* ou *undersampling* aplicada (Figura 21).

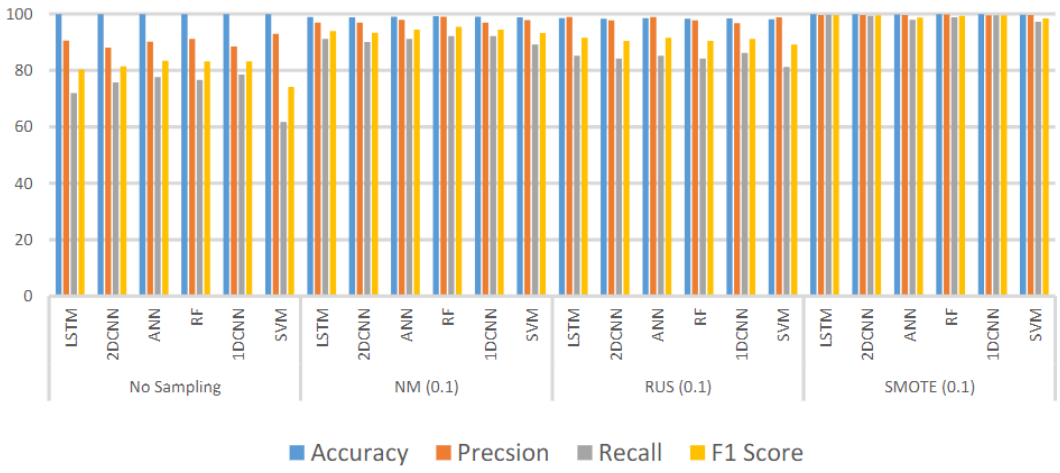

Figura 20 – Resultados das técnicas de *oversampling* e *undersampling* aplicadas ao conjunto ECD em (T. Nguyen et al., 2020)

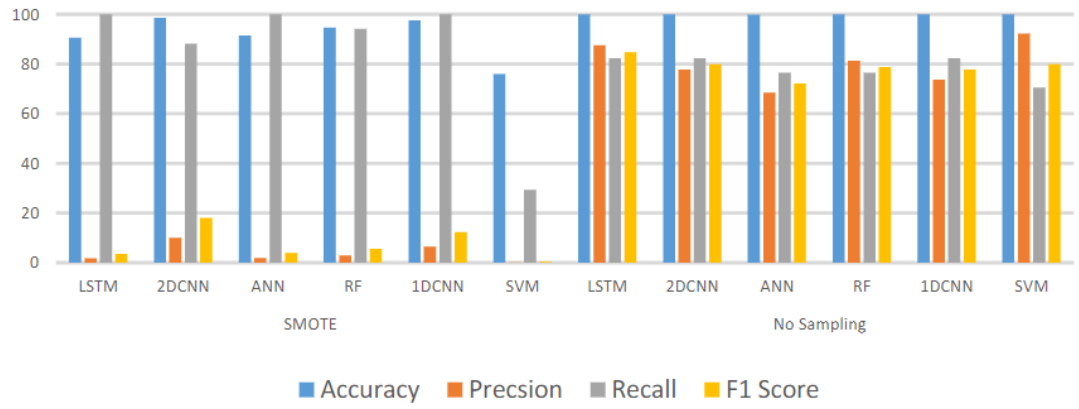

Figura 21 – SMOTE *vs* Distribuição normal no conjunto de teste ECD em (T. Nguyen et al., 2020)

Nguyen e os restantes autores, (2020), verificaram que, como expectável, a revocação aumentou, mas a precisão e pontuação f-1 diminuíram drasticamente com o SMOTE, o que significa que o modelo previu os casos considerados fraude com muita precisão, mas classificou incorretamente a maioria dos casos que não correspondiam a fraude. Os autores concluíram também que o algoritmo de *deep learning* LSTM obteve a melhor performance.

Em termo de conclusão, Nguyen, Tahir, Abdelrazek e Babar (2020) afirmam que o uso de algoritmos de *deep learning* como CNN e LSTM em sistemas de deteção de fraude renderam um melhor desempenho em comparação com os algoritmos tradicionais de *machine learning* com o LSTM a obter uma pontuação f-1 de 84,85%. Os autores mencionam também que, quanto a trabalho futuro, será necessário explorar hiper-parâmetros que são utilizados para construir algoritmos com melhor desempenho.

2.5 Sumário

No que respeita aos artigos estudados e mencionados no ponto 2.4, podemos concluir que todos eles se focam na criação de um modelo de deteção de fraude no *e-commerce*, com algumas particularidades diferentes. Por um lado, Raja, Raman e Ushakiruthika, (2021), baseiam-se em padrões comportamentais e propõem um sistema dividido em três fases em que cria perfis dos utilizadores com base nos comportamentos do mesmo, usa técnicas de machine learning para identificar certos padrões e treina uma coleção de classificadores para detetar atividades suspeitas de fraude. Os resultados compravam que o sistema proposto é mais eficaz quando comparado com outros sistemas. Quanto a JiaoLong Li, (2022), o autor também se foca em nos comportamentos dos utilizadores com a diferença que implementa BDM no seu sistema. Assim, JiaoLong Li, (2022), desenvolve um modelo de deteção de fraudes que combina inteligência artificial com *data mining*. Posteriormente, compara com dois modelos já existentes como o SVM e LR, concluindo que, o modelo proposto, obteve melhores resultados.

Por sua vez, Mathew, Nithya, Vishwanatha, Shetty, Priya e Kavya, (2022) também propõem um sistema de deteção de fraude utilizando técnicas de *machine learning* que, quando comparado com outras técnicas já existentes, obteve melhores resultados em termos de performance. Abhirami, Pani, Manohar e Kumar, (2021) e Saputra e Suharjito, (2019), comparam vários algoritmos de aprendizagem automática disponíveis para detetar fraudes no *e-commerce* com a particularidade que Saputra e Suharjito, (2019) utilizam ainda a técnica de SMOTE para equilibrar o conjunto de dados. Em (Abhirami et al., 2021) o LR obteve melhores resultados quando comparado com as técnicas de SVM, DT, KNN e RF, enquanto que em (Saputra & Suharjito, 2019), afirmam que o SMOTE aumenta o desempenho dos modelos de classificação e o modelo de NB + SMOTE obteve a melhor pontuação f-1.

Numa abordagem diferente e através de *deep learning*, em (T. Nguyen et al., 2020) os autores abordam a aplicação do algoritmo LSTM e CNN num sistema de deteção de fraude. Para isso, aplicaram o algoritmo em três conjuntos de dados diferentes e utilizaram ainda técnicas de *sampling* como RUS, SMOTE e NM de maneira a avaliar a influência de cada um nos resultados pretendidos. No final do estudo, Nguyen, Tahir, Abdelrazek e Babar, (2020), afirmam que o LSTM obteve a melhor performance com uma pontuação f-1 de 84,85%.

As métricas mais utilizadas para avaliar o desempenho de modelos de classificação foram, em concordância, a exatidão, precisão, revocação e pontuação f-1.

Após a análise dos artigos estudados, é possível responder às questões de pesquisa elaboradas no capítulo 2.1.1. Relativamente à **QP1**, conclui-se que são utilizados algoritmos de *machine learning* sempre que é necessário prever ou detetar algo. Estes algoritmos permitirão identificar padrões complexos e prever qual a probabilidade de uma transação ou atividade ser fraudulenta com base em variáveis relevantes. Mathew, Nithya, Vishwanatha, Shetty, Priya e Kavya, (2022), Abhirami, Pani, Manohar e Kumar (2021), Raja, Raman e Ushakiruthika, (2021), JiaoLong Li, (2022), Saputra e Suharjito, (2019) afirmam que algoritmos como LR, RF, NBC, SVM, KNN, DT e XGBOOST são técnicas de aprendizagem supervisionada que podem ser utilizadas no desenvolvimento de um sistema de deteção de fraudes. JiaoLong Li, (2022) utilizou ainda *data mining* para extrair dados desconhecidos ou "ocultos" de inúmeros dados incompletos que podem ser úteis. Saputra e Suharjito, (2019) utilizaram também técnicas como o PCA e SMOTE para o processamento e tratamento de dados, vantajosos para um sistema de deteção de fraudes. Já Nguyen, Tahir, Abdelrazek e Babar (2020) afirmam que as abordagens de *deep learning* LSTM e CNN são as mais apropriadas para sistemas de deteção de fraude. Relativamente à **QP2**, Raja, Raman e Ushakiruthika, (2021) afirmam que mudanças repentinas no padrão de compra de um utilizador pode ser um indicativo de suspeita de fraude. Esta alteração pode ser o aumento repentino nas compras, mudança de produtos ou serviços, entre outros. Por fim, quanto à **QP3**, os autores afirmam que a aplicação de práticas como a verificação de identidade e endereço, autenticação por duas etapas, monitorizar padrões de comportamento e, claro, utilizar um sistema com base em algoritmos de machine learning, ajuda a garantir as medidas necessárias para o combate a atividades fraudulentas.

Em suma, neste capítulo foi assim descrito a metodologia de pesquisa utilizada para a revisão sistemática, assim como as questões de pesquisa, bases de conhecimento, termos de pesquisa, critérios de inclusão e exclusão e todo o processo de extração de dados envolvente. É também feita uma formalização teórica e estado de arte das áreas relacionadas. Por fim, são abordados trabalhos realizados por outros autores sobre o tema em questão e é feita uma breve comparação dos mesmos.

3 Experimentação

No presente capítulo será descrito o conjunto de dados escolhido para o desenvolvimento do sistema de deteção de fraudes, assim como uma análise e avaliação de todos as características incluídas no conjunto de dados. Após esta revisão, a ética e proteção de dados é abordada sob a forma como é aplicada ao conjunto de dados. É também referido todo o processo de tratamento de dados aplicado ao conjunto de dados e o porquê das decisões tomadas ao longo deste procedimento. Por fim, são descritos os algoritmos e ferramentas testados, bem como os testes e validações efetuados com o conjunto de dados escolhido e o respetivo sumário com as conclusões do capítulo.

3.1 Conjunto de dados

Sendo a confidencialidade de dados financeiros uma preocupação de extrema importância para as empresas, um conjunto de dados relativo a transações financeiras é de difícil acesso. Empresas financeiras, como bancos, e qualquer tipo de lojas online, são responsáveis por proteger as informações financeiras dos seus clientes mantendo a confidencialidade dos dados. Isto significa que as empresas adotam medidas de segurança rigorosas para proteger estas informações, como a criptografia de dados, além de seguirem regulamentações rigorosas sobre a privacidade de dados (Duggineni, 2023).

Como resultado desta preocupação, os conjuntos de dados sobre transações financeiras são, geralmente, mantidas em sigilo pelas empresas competentes. Face a este problema, a obtenção de um conjunto de dados para o possível desenvolvimento de um sistema de deteção de fraudes foi uma tarefa complexa.

Após uma longa e cuidadosa análise, em busca de um conjunto de dados relativo a transações financeiras, o conjunto de dados fornecido pela *Vesta Corporation* destacou-se como a escolha mais adequada. Este conjunto de dados surgiu fonte de uma junção entre investigadores da *IEEE Computational Intelligence Society* (IEEE-CIS), sociedade que trabalha em diversas áreas da inteligência artificial, que se uniram com a *Vesta Corporation*, líder global em soluções de pagamento no comércio eletrónico, que oferece tecnologias de prevenção de fraudes patenteadas (Addison Howard Bernadette Bouchon-Meunier, 2019). A razão pela qual este conjunto de dados se destaca dos restantes reside na sua origem colaborativa, que reúne dados financeiros reais, em oposição a dados sintéticos, fruto da junção da IEEE-CIS com a *Vesta Corporation*, especialistas em inteligência artificial e prevenção de fraudes.

O conjunto de dados selecionado para o desenvolvimento da presente tese é constituído por dois ficheiros que contêm informações sobre transações financeiras e a identidade de quem as realiza. Assim, o primeiro ficheiro, *Transactions*, apresenta informações sobre as transações efetuadas contendo:

- quantidade transferida;
- o produto/item referente a essa transação;
- a data em que a transação ocorreu;
- informação sobre o cartão de crédito;
- informações sobre a local onde a transação ocorreu;
- endereções de e-mail do vendedor e destinatário;
- entre outras.

Já o segundo ficheiro, *Identity*, apresenta informações relativas à identidade das pessoas envolvidas na transação, tais como:

- informações de rede;
- tipo de dispositivo;
- informações sobre o dispositivo na qual a transação foi feita;
- entre outras.

As tabelas, contidas nos ficheiros, estão ligadas por uma coluna denominada *TransactionID* que, tal como o nome indica, se refere ao ID da transação efetuada. No entanto, nem todas as transações possuem informações sobre a identidade correspondente, podendo assim haver transações sem informações sobre a identidade das pessoas envolvidas nas transações.

É importante referir que a coluna *isFraud*, presente na *Transaction Table*, é responsável por classificar uma transação como fraudulenta ou legítima. Neste caso, uma transação legítima apresenta o valor 0 e uma transação fraudulenta possui o valor 1. A lógica desta classificação baseou-se no facto de que, se um cliente reclamou uma transação, ela foi classificada como fraudulenta. Nesta reclamação, também conhecido como *chargeback*, o cliente alegou que a transação não foi autorizada pelo mesmo sendo então rotulada como fraude. Além disso, quaisquer transações posteriores e diretamente relacionadas com a conta do utilizador suspeita de fraude foram automaticamente rotuladas como fraude. Em contrapartida, quando nenhuma destas condições foi verificada e nenhum sinal de fraude foi encontrado após 120 dias, a transação foi considerada como legítima e então atribuído o valor 0.

As seguintes tabelas, Tabela 11 e Tabela 12, foram criadas com o propósito de proporcionar uma melhor organização e compreensão do conteúdo do conjunto de dados descrito, permitindo uma análise mais detalhada e precisa dos dados.

Parâmetros	Descrição
TransactionID	Um ID único relativo à transação efetuada;
TransactionDT	Um *timedelta* de uma determinada data e hora. O primeiro valor é 86400, que corresponde ao número de segundos de um dia. O valor máximo é de 15811131, que corresponde ao dia 183. Assim, conclui-

	se que o conjunto de dados de treino é relativo a 6 meses. Já o conjunto de dados de teste é relativo aos 6 meses posteriores a este, com um intervalo de 30 dias entre eles. A junção do conjunto de dados de treino e teste é relativo assim a um ano;
isFraud	Este campo classifica a transação como fraude ou não fraude, possuindo o valor 1 e 0, respetivamente;
TransactionAmt	O montante da transação em dólares americanos;
ProductCD	Refere-se ao código do produto em cada transação. No entanto, um produto não é necessariamente um item e pode corresponder a qualquer tipo de serviço;
Card1 – Card6	Conjunto de características relativos a informações sobre o cartão de crédito como: tipo de cartão, categoria do cartão, banco emissor, país, entre outros;
Addr1 – Addr2	Corresponde à região e país onde a transação foi efetuada;
Dist1 - Dist2	Corresponde à distância entre dois pontos;
P_email_domain	Endereço de e-mail do vendedor;
R_email_domain	Endereço de e-mail do destinatário;
C1 - C14	Conjunto de características que servem como contadores, como por exemplo, quantas moradas estão associadas a um cartão de crédito, quantos endereços de e-mail, entre outras;
D1 - D15	Conjunto de características que correspondem a *timedelta*s, como os dias que passaram entre cada transação;
M1 - M9	Conjunto de características que correspondem a correspondências como: nome no cartão, morada, entre outros;
V1 – V339	Conjunto de recursos fornecidos pela Vesta que inclui classificações, contagens e outras relações de entidades.

Tabela 11 – Características da tabela de transações

Parâmetros	Descrição
TransactionID	Um ID único relativo à transação efetuada;
ID01 – ID38	Conjunto de IDs que correspondem a informações sobre a identidade como: informações sobre a rede (IP, Proxy), entre outras;
DeviceType	Tipo de dispositivo onde a transação foi efetuada;
DeviceInfo	Informações sobre o dispositivo onde a transação foi efetuada.

Tabela 12 – Características da tabela de identidade

É fundamental mencionar que, este conjunto de dados, foi coletado pelo sistema de proteção contra fraudes da *Vesta Corporation* e pelos seus parceiros de segurança digital. A empresa utiliza tecnologias avançadas de proteção de dados e trabalha em estreita colaboração com os seus parceiros de segurança digital para garantir que o conjunto de dados seja tratado com o maior nível de proteção e segurança possível. A *Vesta Corporation* está em conformidade com todas as regulamentações aplicáveis de proteção de dados, especialmente quando se trata de dados tão sensíveis e importantes como transações financeiras. Assim, muitas das informações foram mascaradas de maneira a manter a privacidade dos dados. No capítulo seguinte, 3.1.1, é tratado de forma mais aprofundada a questão da ética e proteção de dados envolvida.

3.1.1 Ética e proteção de dados

A inteligência artificial é uma área que se encontra em constante evolução e rápido crescimento devido aos progressos tecnológicos desenvolvidos. A criação de novos métodos e tecnologias para solucionar problemas permite que a IA seja sinónimo de inovação. No entanto, a ética e proteção de dados é um fator crucial na área de *machine learning* pois os sistemas de IA são alimentados por grandes quantidades de dados pessoais e confidenciais (Szász et al., 2022). Estes dados necessitam de uma proteção adequada no tratamento dos mesmos, de maneira a prevenir possíveis divulgações não autorizadas de informação e, consequentemente, possíveis fraudes e crimes cibernéticos.

Neste contexto, e como em qualquer outro, é importante seguir regulamentos e diretrizes de proteção de dados, como o Regulamento Geral de Proteção de Dados (RGPD) da União Europeia para garantir que os dados sejam coletados e armazenados de forma segura e ética. Este regulamento "aplica-se a uma empresa ou entidade que efetue o tratamento de dados pessoais no âmbito das atividades de uma das suas sucursais estabelecida na União Europeia, independentemente do local onde são tratados ou uma empresa constituída fora da União Europeia que oferece bens/serviços (pagos ou gratuitos) ou controla o comportamento de pessoas na União Europeia." (Comissão Europeia, 2022). Algumas dessas regulamentações incluem a minimização dos dados utilizados, de forma a diminuir a divulgação de informações desnecessárias, e o consentimento dos titulares. Os dados devem ser utilizados para um fim específico e assim que não forem mais necessários deverão ser excluídos e devem ser tratados com integridade e confidencialidade.

Assim, para o desenvolvimento do presente projeto, o conjunto de dados escolhido e mencionado no capítulo 3.1, não compromete a integridade dos titulares presentes no seu conjunto de dados. Todas as regras de segurança, proteção e ética de dados foram aplicados no tratamento e análise dos dados em questão. Como referido anteriormente, a empresa segue rigorosamente as diretrizes e regulamentações do RGPD em relação aos dados fornecidos. Foram utilizadas técnicas de criptografia e políticas de privacidade e segurança robustas em compromisso com a privacidade dos dados e em conformidade com o RGPD. Desta forma, a confidencialidade dos dados dos clientes foi garantida.

3.2 Análise exploratória de dados

Neste capítulo serão apresentados os resultados obtidos a partir da análise exploratória dos dados, incluindo informações sobre a distribuição dos dados, correlações entre variáveis e padrões encontrados no conjunto de dados. Esta análise é crucial no desenvolvimento do presente trabalho pois permite obter uma avaliação preliminar do conjunto de dados, assim como retirar as primeiras conclusões sobre os dados em questão. Nesta análise, o conjunto de dados é submetido a diversas técnicas estatísticas e gráficas, como histogramas, *box plots* e *scatter plots*, com o objetivo de identificar padrões e tendências nos dados, bem como identificar possíveis problemas como *outliers* e dados em falta.

O primeiro passo passou por avaliar a classe alvo do conjunto de dados *(isFraud)*, também conhecida como a classe *target*. Esta variável representa o resultado que se deseja prever a partir de outras variáveis do conjunto de dados. Num problema de classificação, como um sistema de deteção de fraudes, esta classe permite determinar se uma determinada transação é classificada como fraude ou não fraude, como mencionado no ponto 3.1. Esta classe é também utilizada para posteriormente avaliar o desempenho do modelo criado a partir do conjunto de dados, a partir do conjunto de dados de teste. Assim, depois de importar o conjunto de dados, foi possível concluir que os dados se encontram bastante desequilibrados (Figura 22).

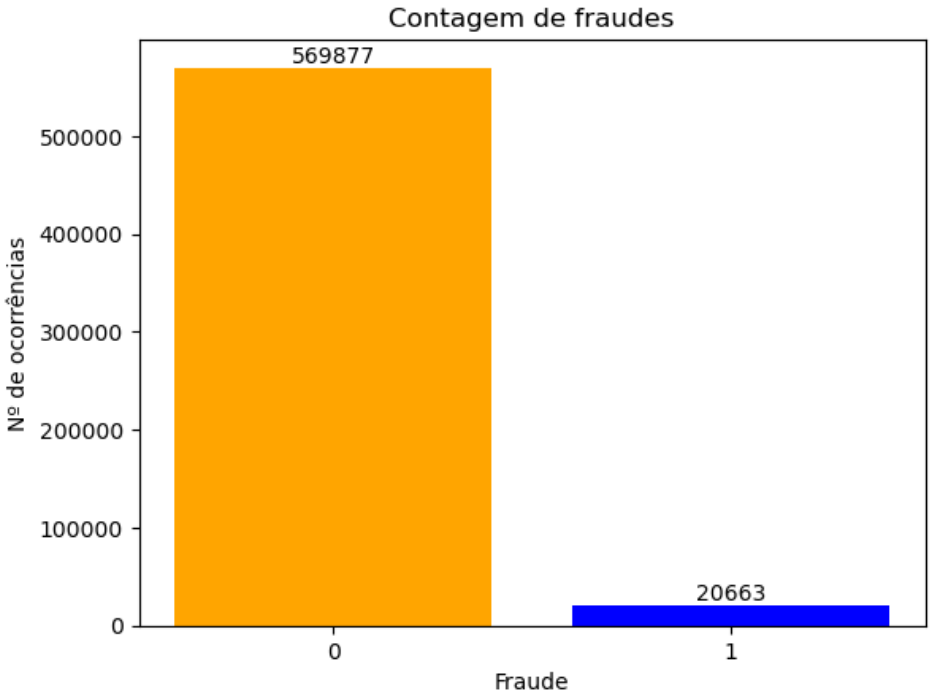

Figura 22 - Contagem de fraudes

O conjunto de dados atual é, então, composto por 590 540 transações financeiras, das quais apenas 3.5% representam transações fraudulentas e os restantes 96.5% não correspondem a qualquer tipo de fraude. Infere-se assim que o conjunto de dados se encontra bastante desequilibrado e será necessário recorrer a técnicas de *oversampling* para equilibrar

os dados. Sem este equilíbrio, a probabilidade de ocorrer o problema de *overfitting* é muito elevada pois o modelo teria um excelente desempenho nos dados de treino, contundo, nos dados de teste, os resultados seriam muito pouco precisos e enganadores (Saputra & Suharjito, 2019). Neste caso, o modelo preditivo assumiria que grande parte das transações não são fraudulentas, quando na verdade são. Assim, a técnica de *oversampling* a aplicar será aprofundada no capítulo seguinte, 3.2.1.

É também primordial analisar os valores nulos presentes no conjunto de dados visto que estes valores podem afetar diretamente a interpretação dos dados e, consequentemente, os resultados do modelo pretendido. Sem esta remoção, os valores nulos podem afetar a precisão do modelo e à perde de informações importantes. Face a este problema, identificou-se quais os parâmetros que possuem valores nulos e inferiu-se que grande parte dos registos da classe *Identity* possuem valores nulos (Figura 23). As colunas com um rácio de valores nulos elevado serão posteriormente tratadas de forma a não comprometer a qualidade e precisão dos resultados e conclusões que se pretendem obter.

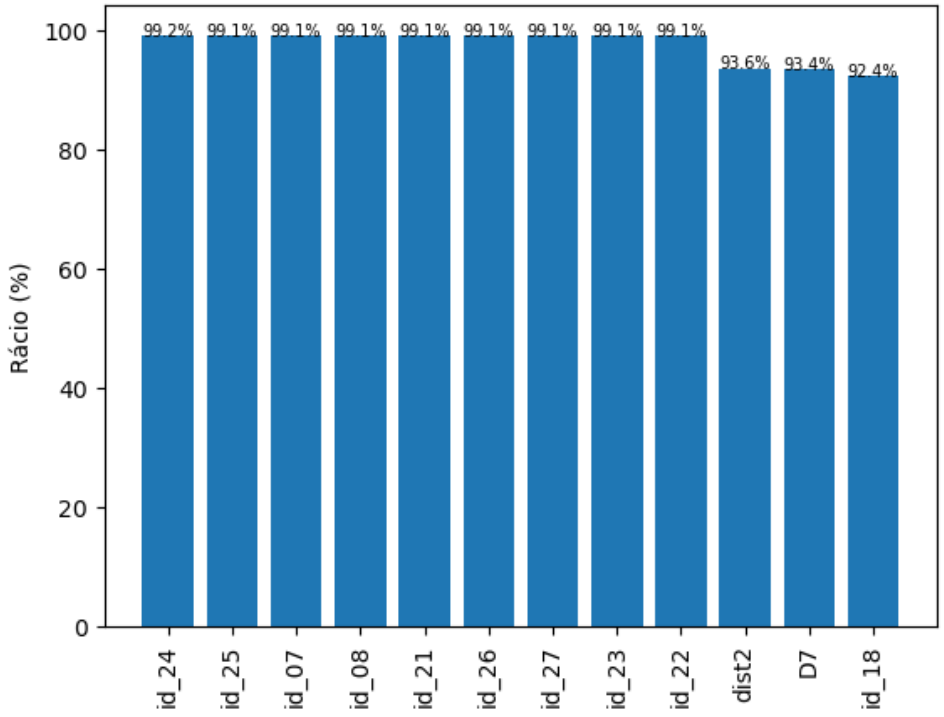

Figura 23 – Rácio de valores nulos

O próximo passo passou por analisar algumas variáveis categóricas presentes no conjunto de dados. As seguintes variáveis categóricas, que assumem um conjunto finito de valores, foram analisadas:

- "*ProductCD*";
- "*P_email_domain*";

- “*R_email_domain*”;
- “*Card1 – Card6*”;
- “*DeviceType*”;
- “*DeviceInfo*”.

A escolha, em particular, desta variáveis prendeu-se com o facto de possuírem características que são facilmente compreensíveis e interpretáveis, proporcionando uma maior clareza na análise dos dados.

- **ProductCD**

Esta variável refere-se a um código que está associado a um produto em cada transação e pode assumir um de cinco códigos possíveis: C, H, R, S e W. Foi também calculada a percentagem de transações fraudulentas para cada código do produto presente no conjunto de dados. O número de ocorrências de cada código de produto e a respetiva percentagem de fraude está representada na Figura 24.

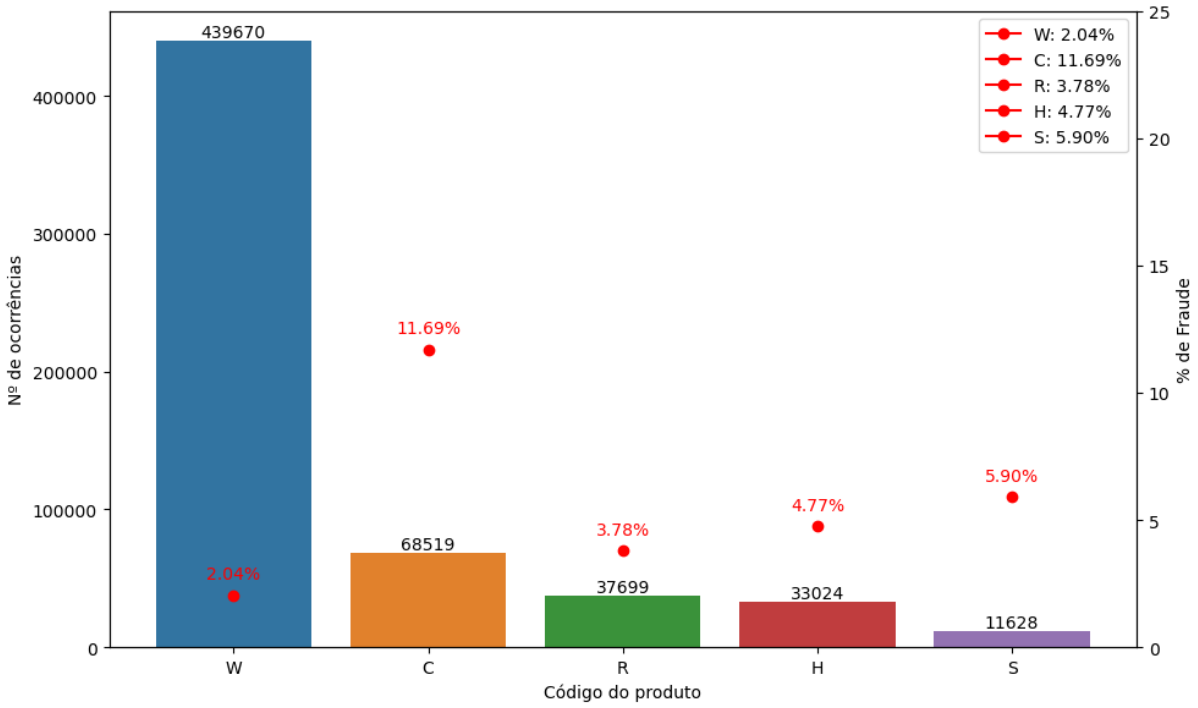

Figura 24 – “*ProductCD*” e respetiva percentagem de fraude

Conforme evidenciado na figura acima, o código de produto “W” constitui cerca de 74,45% do conjunto de dados com 439 670 transações associadas ao mesmo. Relativamente a transações fraudulentas, cerca de 2,04% destas transações são consideradas fraude. Por outro lado, o código “S” apresenta o menor número de ocorrências no conjunto de dados com cerca de 11 628 transações. Destas, 5.90% correspondem a transações consideradas fraude.

- **P_email_domain e R_email_domain**

Estas variáveis representam os endereços de e-mail do vendedor e do destinatário. Primeiramente agrupou-se endereços de e-mail semelhantes em categorias mais amplas e preencheu-se os valores nulos destas variáveis com “Sem informação”. Isto foi feito através de um mapeamento, onde cada endereço de e-mail foi atribuído a uma nova categoria. Por exemplo, todos os endereços que contêm “Gmail” foram agrupados na categoria “Google”, enquanto que aqueles com “Yahoo” foram agrupados na categoria “Yahoo Mail”, entre outros. Aqueles que aparecem menos de 1000 vezes foram agrupados na categoria “Outros”. As Figura 25 e Figura 26 representam os domínios de e-mail utilizados pelos vendedores e destinatários, assim como a percentagem de fraude que está associado a cada categoria distinta.

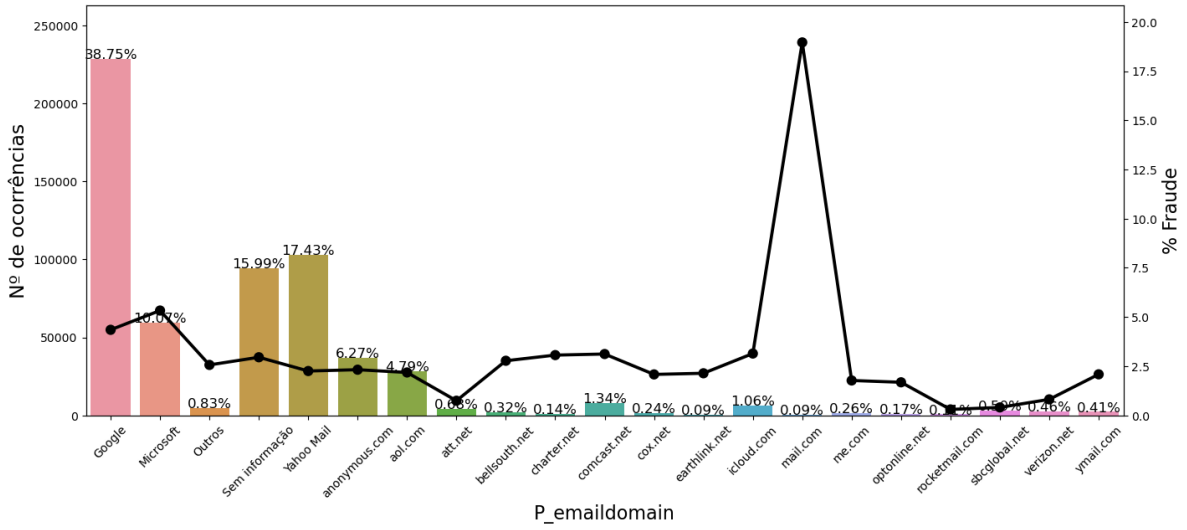

Figura 25 – “*P_email_domain*” e respetiva percentagem de fraude

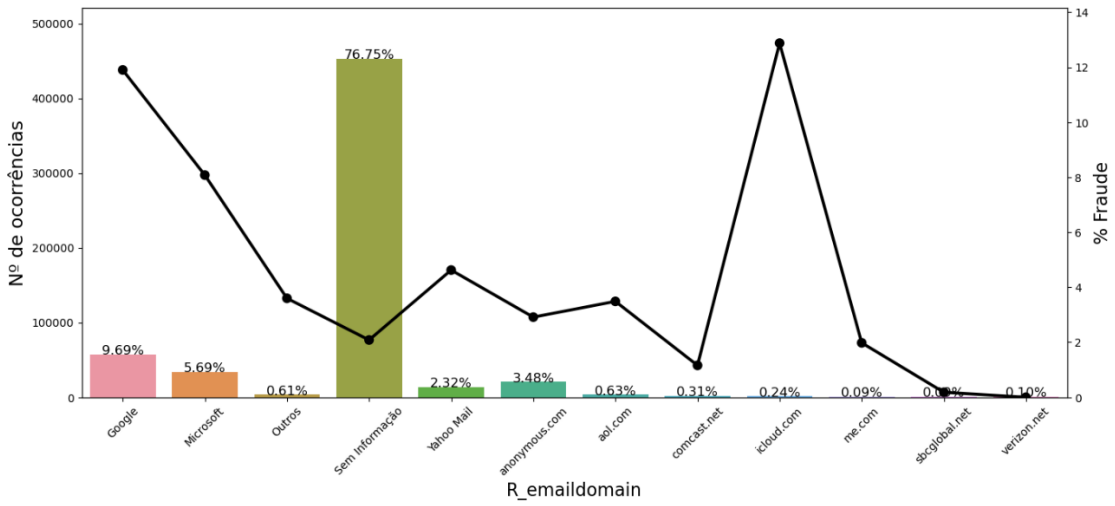

Figura 26 - “*R_email_domain*” e respetiva percentagem de fraude

Desta análise, infere-se que o domínio mais utilizado nos endereços de e-mail dos vendedores são os endereços “Google” onde cerca de 5% correspondem a transações fraudulentas. Por outro lado, relativamente aos endereços dos destinatários, cerca de 76,75% desta variável apresentava valores nulos e foram então etiquetados com “Sem informação”. É também de salientar que dos 9,69% de endereços “Google”, cerca de 12% correspondem a transações fraudulentas.

- **DeviceType**

Esta variável representa o tipo de dispositivo onde a transação financeira foi efetuada e, apesar de a maior parte dos registos estar vazios, esta variável pode assumir dois valores, entre eles: “d*esktop*” ou “*mobile*”. O tipo “*desktop*” significa que a transação foi efetuada através de um computador e, por outro lado, “mobile” indica que a transação foi concretizada via dispositivo móvel. A Figura 27 representa a distribuição desta variável presente no conjunto de dados.

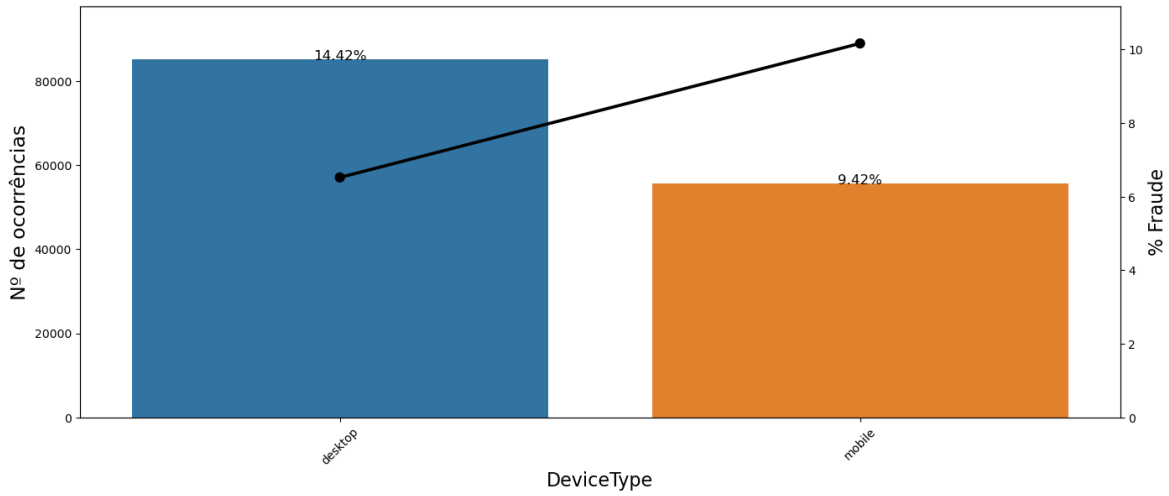

Figura 27 - “*DeviceType*” e respetiva percentagem de fraude

É possível verificar que a maioria das transações, que têm associado um tipo de dispositivo, foram efetuadas através de um “*desktop*”.

- **DeviceInfo**

Este atributo contém informações adicionais sobre o dispositivo utilizado para realizar as transações financeiras e complementa os dados fornecidos pelo atributo “DeviceType”. Estas informações podem incluir detalhes sobre o modelo do dispositivo, assim como o respetivo fabricante e sistema operacional. Tal como nos endereços de e-mail, agrupou-se os dispositivos semelhantes, através de um mapeamento, em categorias mais amplas e preencheu-se os

valores nulos com “Informação desconhecida”. A Figura 28 representa a distribuição deste atributo assim como a respetiva percentagem de fraude de cada um.

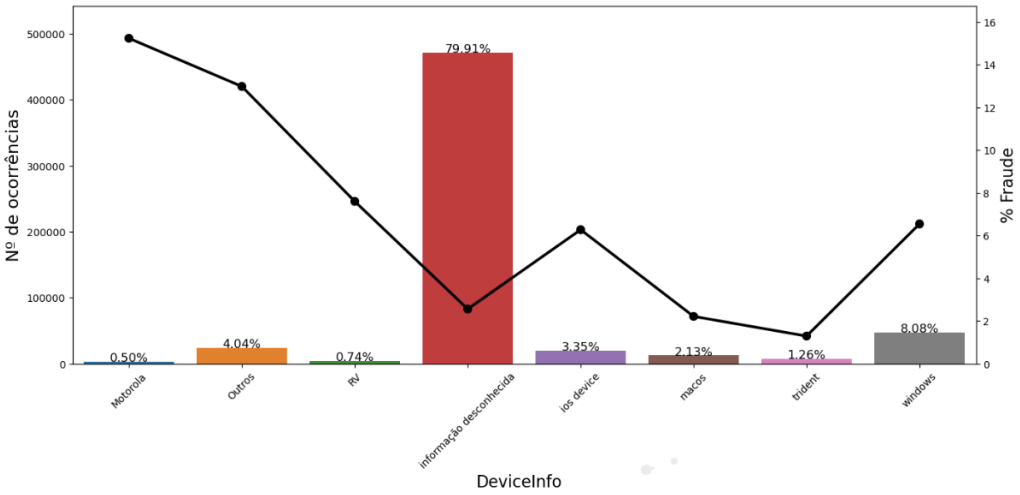

Figura 28 – “DeviceInfo” e respetiva percentagem de fraude

É possível inferir que 79.91% dos registos se encontram sem informação, o que significa que esta variável se encontra com muitos valores não definidos e por isso vazios. Ainda assim, conclui-se que os restantes dispositivos pertencem, na sua maioria, à categoria “*Windows*”.

- **Card1 – Card6**

Este conjunto de atributos abrange informações relativas aos cartões de crédito utilizados durante as transações financeiras. Com o objetivo de salvaguardar a privacidade dos dados, algumas características foram mascaradas e convertidas em valores numéricos, entre elas: “*card1*”, “*card2*”, “*card3*” e “*card5*”. Já as variáveis “*card4*” e “*card6*” assumem um conjunto de valores que identificam tanto o tipo de cartão como a rede em que estão vinculados. As Figura 29 e Figura 30 representam os valores mais frequentes destes atributos, presentes no conjunto de dados, e a respetiva percentagem de fraude associados aos mesmos.

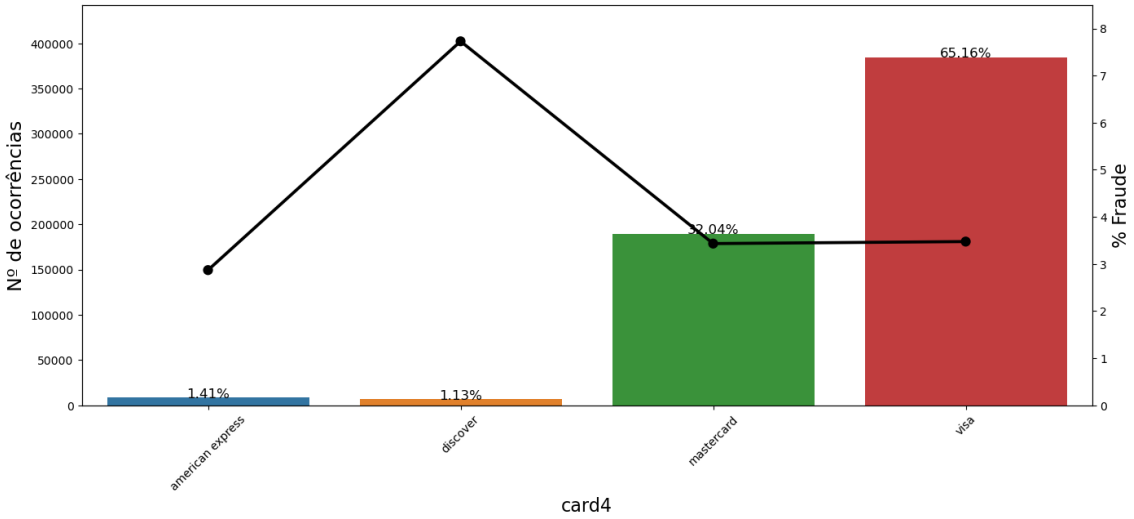

Figura 29 – "*Card4*" e respetiva percentagem de fraude

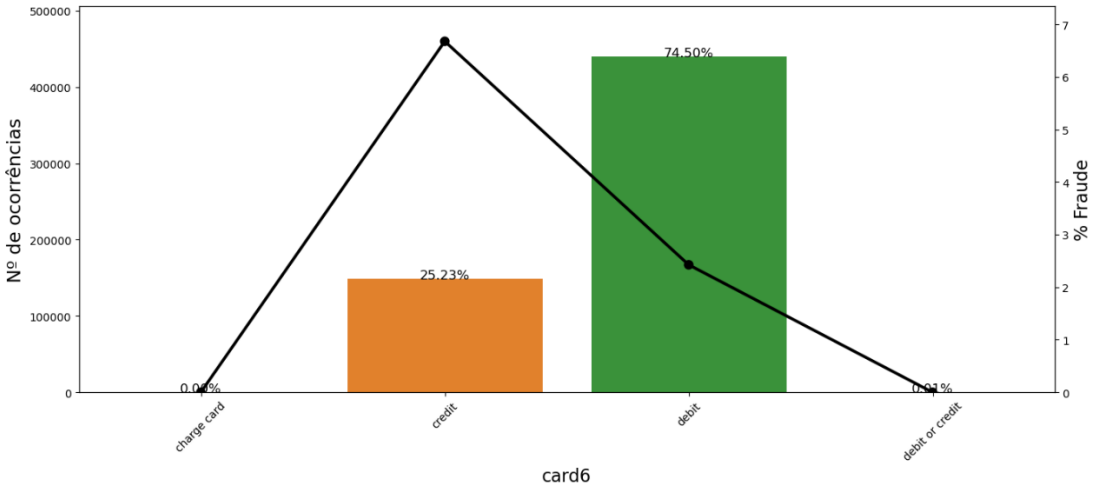

Figura 30 - "*Card6*" e respetiva percentagem de fraude

De acordo com os gráficos acima representados, é possível verificar que a grande maioria dos cartões de crédito utilizados nas transações financeiras efetuadas pertence à rede "*Visa*" ou "*Mastercard*", representando aproximadamente 97,2% do total. Por sua vez, também se conclui que estes cartões de crédito são do tipo crédito ou débito.

3.2.1 Tratamento de dados

Desde o início foram implementadas diversas estratégias para assegurar a qualidade dos dados utilizados na elaboração do presente trabalho. Todas estas decisões foram tomadas com o propósito de preparar o conjunto de dados de modo a treinar o modelo pretendido da forma mais eficaz e otimizada possível, e com o objetivo de minimizar possíveis distorções e erros nos resultados obtidos a partir dos algoritmos que serão aplicados ao conjunto de dados.

3.2.1.1 Remoção de nulos e informação repetida

Inicialmente foram removidas as colunas que apresentavam mais de 90% de valores nulos, assim como aquelas onde mais de 90% da informação contida era repetida. Desta filtragem, 66 colunas foram removidas, entre elas 12 pela percentagem de valores nulos e as restantes 54 por conterem 90% de informação repetida. As colunas removidos são: "V103", "V286", "dist2", "id_23", "id_24", "V119", "id_21", "id_26", "V132", "id_18", "id_08", "V134", "V297", "V111", "V124", "V115", "V298", "V320", "V122", "V284", "V110", "V116", "V319", "id_07", "V106", "V121", "V296", "V295", "V109", "V112", "V125", "V129", "V137", "V311", "V136", "V105", "V305", "V321", "V104", "V135", "V301", "D7", "V107", "V101", "id_25", "V133", "V281", "V98", "V108", "id_27", "V120", "V118", "V300", "V114", "V123", "V309", "V117", "V293", "V316", "V102", "C3", "V290", "V299", "V318", "id_22", "V113". Desta remoção, o conjunto de dados ficou reduzido então a 368 colunas.

3.2.1.2 Tratamento de variáveis

O próximo passo passou por tratar os valores nulos das variáveis categóricas e numéricas. Para as variáveis categóricas, preencheu-se os campos vazios com a informação "*No Info*" de forma a indicar que não há dados disponíveis para aquele registo. Relativamente às variáveis numéricas, preencheu-se os valores nulos com a média da coluna em específico. Com esse intuito, calculou-se para cada atributo numérico a média correspondente e, os campos vazios, foram preenchidos com a média obtida. Esta abordagem permite, ao invés de remover os valores nulos e perder informações valiosas, preservar a distribuição geral dos dados, mantendo a integridade dos mesmos e, obter resultados mais precisos. Outra consideração importante é que alguns algoritmos de *machine learning* não lidam bem com valores nulos e, preencher estes valores com a respetiva média nas variáveis numéricas, torna o conjunto de dados compatível com esses algoritmos (Ng & Winkler, 2014). Após este processo, o conjunto de dados não possui qualquer valor nulo.

3.2.1.3 Codificação de variáveis categóricas

De seguida, foi necessário codificar as variáveis categóricas em numéricas através do *encoding*. O *encoding* é o processo de transformar dados categóricos numa representação numérica que torna os algoritmos de *machine learning* mais eficazes (Himanshu Tripathi, 2019). A técnica de codificação utilizada foi o *Label Encoding* que permite atribuir um valor numérico a cada categoria de uma variável categórica. Assim, aplicou-se esta técnica a todas as variáveis categóricas, de forma a tornar o conjunto de dados compatível com os algoritmos de *machine learning* que serão implementados posteriormente. Este processo facilita também o processamento de dados, permitindo que os algoritmos extraiam padrões e executem previsões com mais eficácia (Himanshu Tripathi, 2019).

Com o conjunto de dados tratado, foi então efetuada a divisão entre o conjunto de treino e teste.

3.2.1.4 Divisão dos dados em conjunto de treino e teste

No processo de divisão do conjunto de dados em treino e teste, utilizou-se o método "*train_test_split*" com uma proporção de 70% dos dados destinados ao conjunto de treino e os restantes 30% para o conjunto de teste. Esta abordagem permite avaliar a performance do modelo com dados desconhecidos, neste caso, o conjunto de dados de teste (Pawluszek-Filipiak & Borkowski, 2020). Ao aplicar o método "*train_test_split*", os dados são divididos de forma aleatória em conjuntos distintos. O conjunto de treino será utilizado para treinar o modelo, que o permitirá aprender as relações e padrões presentes nos dados. Já o conjunto de teste será utilizado para avaliar a eficácia do modelo. Ao escolher dividir os dados numa proporção de 70% para treino e 30% para teste, assegura-se que uma quantidade considerável de dados é utilizada para treinar o modelo, permitindo que este aprenda e entenda os padrões necessários para identificar fraudes.

Desta forma, o conjunto de dados de treino é agora composto por 413 378 transações das quais 14 516 são fraude e as restantes 398 862 não correspondem a qualquer suspeita de fraude. Por outro lado, o conjunto de dados de teste é composto por 177 015 transações, das quais 6 147 são transações fraudulentas e as restantes 171 015 são transações legítimas.

3.2.1.5 Conjuntos de dados para treinar o modelo *e Oversampling/Undersampling*

Depois de aplicada a divisão dos dados conforme descrito, foi necessário lidar com o desequilíbrio presente no conjunto de dados e conforme descrito anteriormente no capítulo 3.1. Como tal, com recurso a técnicas de *oversampling* e *undersampling*, descritas ao longo deste subcapítulo**Erro! A origem da referência não foi encontrada.**, foi possível alcançar um equilíbrio no conjunto de dados. A decisão de aplicar estas duas técnicas recai sobre o facto de que serão utilizados diferentes conjuntos de dados para treinar o modelo de *machine learning* pretendido e, consequentemente, avaliar qual o conjunto de dados que obterá melhores resultados e, proporcionará um desempenho superior. A utilização de conjuntos de dados distintos, permitirá observar como cada técnica afeta o desempenho do modelo. Nesse sentido, o modelo será treinado sob três conjuntos de dados distintos:

1. **Conjunto de dados [A]:** conjunto de todos os dados de treino, sem nenhuma modificação adicional, depois de aplicada a divisão entre treino e teste. Neste conjunto, existem 413 378 transações das quais 14 516 são fraudulentas e as restantes 398 862 não correspondem a fraude;

2. **Conjunto de dados [B]:** conjunto de dados de treino proveniente da aplicação da técnica de *oversampling*, *SMOTE*. Consiste em 797 724 transações, sendo que metade (398 862) são consideradas fraudulentas e a outra metade (398 862) são transações fidedignas;

3. **Conjunto de dados [C]:** conjunto de dados de treino proveniente da aplicação da técnica de *undersampling*, *RandomUnderSampler*. Consiste em 29 032 transações, das quais 50% das transações são fraudulentas (14 516) e os restantes 50% (14 516) são transações legítimas.

- ***Oversampling* (SMOTE)**

Como referido em 2.4.2, o *SMOTE* é uma técnica de *oversampling* que permite aumentar o número de casos da classe minoritária de forma equilibrada, ao aumentar o número de classes positivas até estes serem iguais às classes negativas (Sharma et al., 2018). O *SMOTE* foi aplicado ao conjunto de dados depois de efetuada a divisão entre os dados de treino e teste. É importante aplicar esta técnica após a divisão do conjunto de dados em treino e teste, pois assegura-se assim, que o modelo será avaliado sem influência das instâncias sintéticas criadas. Isto garante uma avaliação mais realista do desempenho do modelo pois evita que informações do conjunto de teste sejam usadas na criação de dados sintéticos.

Após esta divisão, o conjunto de dados de treino é então composto por 413 378 transações, das quais 398 862 correspondem a fraude e as restantes 14 516 a transações legítimas. A aplicação da técnica *SMOTE* permitirá subir o número de transações legítimas para 398 862, igualando as transações fraudulentas (Figura 31).

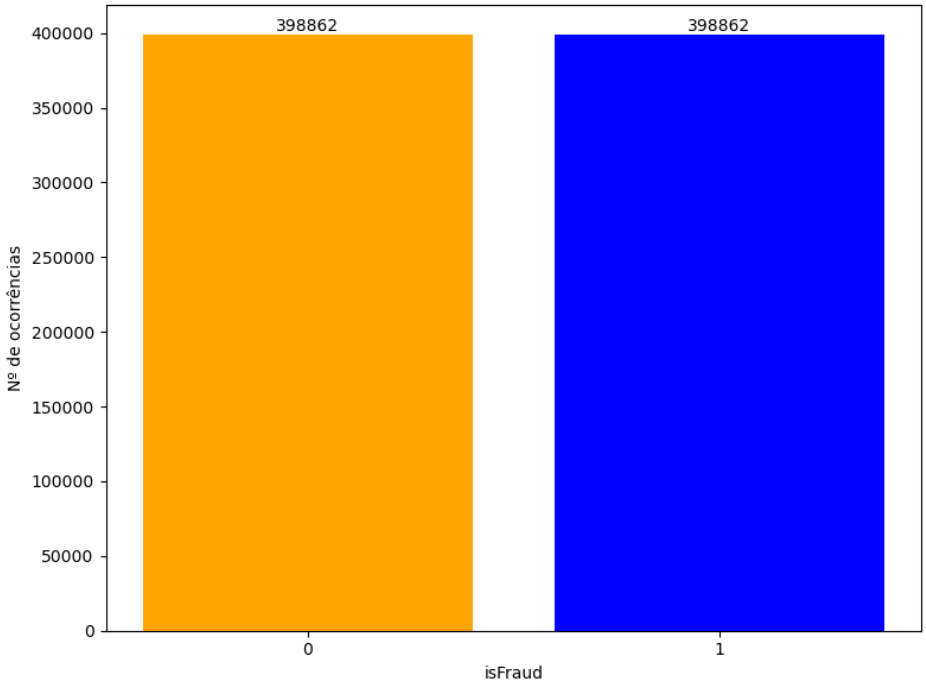

Figura 31 – Distribuição dos dados após a aplicação da técnica *SMOTE*

Este conjunto de dados será utilizado como uma das opções para treinar o modelo e analisar os resultados obtidos. O objetivo é determinar com qual conjunto de dados o modelo obterá melhores resultados em termos de eficácia e avaliar a relevância da técnica *SMOTE* neste contexto.

- ***Undersampling* (RandomUnderSampler)**

O *RandomUnderSampler* é uma técnica de *undersampling* que seleciona e remove aleatoriamente exemplos da classe majoritária do conjunto de dados de treino até que se alcance uma distribuição equilibrada (Dowlagar & Mamidi, 2022). Ao contrário da técnica de *SMOTE*, que cria novas amostras sintéticas da classe minoritária, o *RandomUnderSampler* limita-se a reduzir o tamanho do conjunto de dados, removendo aleatoriamente amostras da classe predominante (Dowlagar & Mamidi, 2022). Tal como no SMOTE, é importante aplicar esta técnica após a divisão do conjunto de dados em conjunto de treino e teste para garantir que a redução do tamanho seja realizada apenas no conjunto de treino. Desta forma, asseguramos que o modelo será avaliado de forma imparcial, permitindo que o conjunto de dados de teste contenha dados que ainda não foram observados.

Após a aplicação desta estratégia, o conjunto de dados ficou então reduzido a 29 032 transações financeiras, das quais 14 516 correspondem a transações fraudulentas e as restantes 14 516 correspondem a transações que não cometem qualquer ilegalidade. A Figura 32 representa o número de ocorrências da variável alvo "*isFraud*" no conjunto de dados de treino, após a técnica de *RandomUnderSampler*.

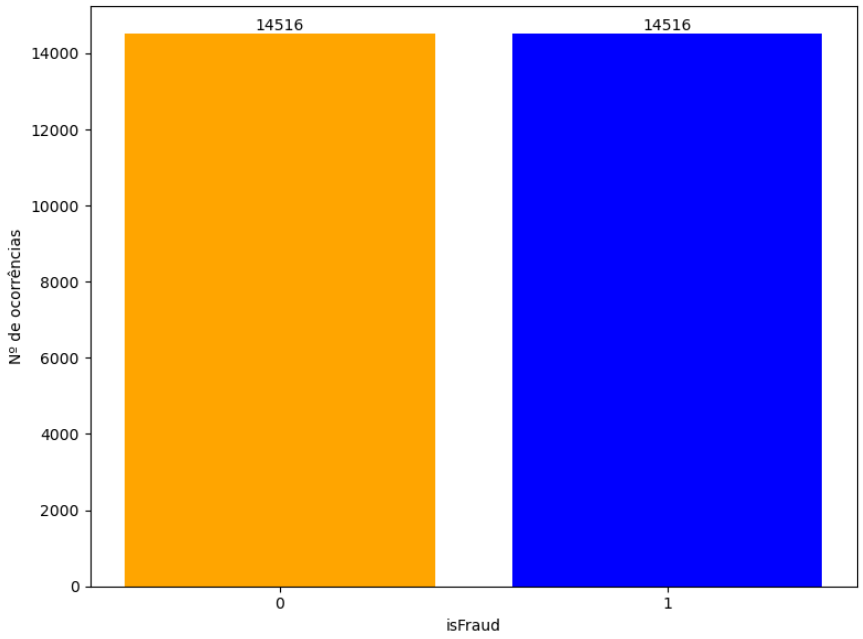

Figura 32 - Distribuição dos dados após a aplicação da técnica *RandomUnderSampler*

Tal como no processo de *oversampling*, este conjunto de dados será utilizado como uma das opções para treinar o modelo e avaliar os resultados obtidos.

3.2.1.6 Normalização dos dados

Por fim, o último passo do tratamento de dados passou por normalizar os três conjuntos de dados já mencionados. A normalização é um processo essencial em análises de dados e certos algoritmos de *machine learning*, como é o caso do SVM, LR e KNN, explicados em 2.3.1, por serem algoritmos sensíveis à escala. O objetivo da normalização é transformar as variáveis

do conjunto de dados numa escala comum de forma a evitar problemas de escala e garantir que os algoritmos de *machine learning* sejam aplicados de forma correta e consistente (Singh & Singh, 2020).

O algoritmo SVM é sensível à escala das variáveis presentes no conjunto de dados, o que significa que, se as variáveis não estiverem todas na mesma escala, o algoritmo pode não funcionar corretamente (Graf Arnulf B.A. and Borer, 2001). A normalização dos dados é assim fundamental, para garantir que as margens de separação entre as diferentes classes sejam determinadas de maneira adequada (Ahsan et al., 2021). Da mesma forma, é importante normalizar os dados antes de implementar o algoritmo de LR para garantir que os coeficientes atribuídos a cada variável estejam na mesma escala, melhorando assim a interpretação dos coeficientes e a estabilidade do modelo (Ahsan et al., 2021). Relativamente ao KNN, a normalização é necessária pois é um algoritmo que tem por base a distância entre dois pontos. Se as variáveis não forem normalizadas e tiverem escalas diferentes, aquelas com escalas maiores terão um peso desproporcional na computação das distâncias, o que provoca distorção nos resultados (Ahsan et al., 2021).

Por outro lado, os algoritmos tradicionais DT e RF, assim como arquiteturas de *deep-learning* como LSTM e CNN, também explicados em 2.3.1, não são sensíveis à escala das variáveis e, então, a normalização dos dados não é necessária. Ainda assim, estes algoritmos foram aplicados aos conjuntos de dados com e sem normalização, de forma a avaliar os resultados obtidos de todos os cenários possíveis.

A técnica de normalização escolhida foi o *Standard Scaler* que é uma técnica comum de pré-processamento de dados. Este método transforma as variáveis de um conjunto de dados de forma que a sua distribuição tenha uma média igual a zero e um desvio padrão igual a um. Esta é uma prática comum na normalização de dados que tem como objetivo centralizar e padronizar a escala das variáveis. Esta técnica foi escolhida pois proporciona uma representação consistente dos dados e permite que o modelo obtenha melhores resultados (Ferreira et al., 2019).

3.3 Sumário

Foi selecionado um conjunto de dados fornecido pela *Vesta Corporation* e pela IEEE-CIS que cumprem rigorosamente as diretrizes e regulamentações do RGPD em relação aos dados fornecidos. Estas entidades utilizaram técnicas de criptografia e políticas de segurança de forma a garantir a confidencialidade dos dados dos clientes.

Ao longo do capítulo foi realizada uma análise exploratória dos dados em que foi possível compreender algumas características do conjunto de dados, assim como possíveis padrões e comportamentos. De modo a assegurar o melhor resultado possível do modelo proposto para um sistema de deteção de fraude, foi efetuado um tratamento ao conjunto de dados escolhido. Primeiramente, os dois ficheiros, *Transaction* e *Identity*, foram agregados num só de forma a criar uma única tabela. De seguida foi efetuado um tratamento ao conjunto de dados que

passou pela remoção de valores nulos e informação repetida, o tratamento de variáveis categóricas e numéricas e respetiva codificação e a divisão dos dados em conjuntos de treino e teste. Para além disso, foram também aplicadas técnicas de *oversampling* e *undersampling* com o objetivo de equilibrar os dados e treinar o modelo segundo diferentes conjuntos de dados. Assim será possível avaliar os diferentes resultados obtidos e poder inferir sobre a eficácia e importância de cada técnica.

Por último, foi necessário normalizar os dados para evitar problemas de escala e garantir que os algoritmos de *machine learning* sensíveis à escala obtenham os melhores resultados possíveis. Com isto, será também possível analisar o impacto da normalização do conjunto de dados nos algoritmos.

4 Implementação e Avaliação

No presente capítulo serão implementados os algoritmos de *machine learning* tradicional e arquiteturas de *deep-learning* aos conjuntos de dados provenientes do capítulo anterior. Serão explicados também as métricas de performance utilizadas e o porquê de serem escolhidas em comparação a outras existentes, juntamente com o objetivo de cada uma e a maneira como são calculadas. Além disso, é também descrito o modelo proposto que consiste na utilização de hiper-parâmetros modificados nos algoritmos com melhor desempenho e que permitem obter melhores resultados em comparação com os algoritmos existentes. É explicado todo o processo de raciocínio envolvente e o porquê dos parâmetros escolhidos. Ao longo do capítulo são feitas análises e avaliações dos resultados obtidos e no final é feita uma comparação final dos algoritmos implementados e do modelo proposto.

4.1 Algoritmos e métricas de performance

Como referido no capítulo 2 e 3, serão implementados os seguintes algoritmos de *machine learning* tradicionais aos conjuntos de dados resultantes do subcapítulo 3.2.1: LR, RF, SVM, KNN e DT, assim como arquiteturas de *deep-learning* LSTM e CNN. Além destes algoritmos, será ainda implementado um novo modelo, fruto da exploração dos hiper-parâmetros utilizados nos algoritmos, nos modelos com melhor desempenho. Neste caso, serão escolhidos o melhor algoritmo tradicional e o melhor algoritmo de *deep-learning,* e os hiper-parâmetros destes algoritmos serão modificados com o objetivo de melhorar a sua performance. Este modelo é analisado detalhadamente no subcapítulo 4.3, e será feita uma comparação dos resultados obtidos no subcapítulo 4.2 e 4.4**Erro! A origem da referência não foi encontrada.**.

Para avaliar os modelos, existem diversas métricas que são utilizadas na avaliação de problemas de classificação e que são bastante importantes no que toca à performance dos mesmos. Para tal, as métricas selecionadas para todos os algoritmos foram: exatidão (*accuracy*), precisão (*precision*), revocação (*recall*) e pontuação f-1 (*f-1 score*). Estas métricas foram também escolhidas pelos autores dos artigos estudados em 2.4 e são calculadas através de uma tabela que demonstra o desempenho de um modelo de classificação, denominada matriz de confusão e ilustrada na Tabela 13 (Karimi, 2021) . Esta matriz é uma representação visual do desempenho de um algoritmo, que permite a análise das previsões corretas e dos erros cometidos pelo modelo. Esta é composta por classes que representam a contagem de amostras classificadas em cada categoria, entre elas (Karimi, 2021):

- **Verdadeiros Positivos (TP)** – valores que são positivos e previstos como positivos. No estudo em questão, são transações que são fraude e previstas como fraude;

- **Verdadeiros Negativos (TN)** – valores que são negativos e previstos como negativos. No estudo em questão, são transações que não são fraudulentas e previstas como não fraudulentas;

- **Falsos Positivos (FP)** – valores que eram negativos e foram previstos como positivos. No estudo em questão, são transações que não são fraudulentas, mas foram previstas como fraude;

- **Falsos Negativos (FN)** – valores que eram positivos e foram previstos como negativos. No estudo em questão, são transações que são fraude, mas foram previstas como não fraudulentas.

A matriz de confusão, adaptada ao tema em questão, encontra-se ilustrada na tabela seguinte (Tabela 13).

		Valor previsto	
		Fraude	Não fraude
Valor real	Fraude	TP	FN
	Não Fraude	FP	TN

Tabela 13 – Matriz de confusão

Com base nestes valores fornecidos pela matriz de confusão, é possível calcular várias métricas de desempenho, entre elas (Tasnim et al., 2022):

- **Exatidão (*accuracy*):** mede a proporção de transações classificadas corretamente como fraude ou não fraude em relação ao número total de transações e é calculada da seguinte maneira:

$$Exatidão = \frac{TP + TN}{TP + TN + FN + FP}$$

- **Precisão (*precision*):** mede a proporção de transações classificadas corretamente como fraude em relação ao número total de transações classificadas como fraude. É uma métrica importante quando o objetivo é reduzir o número de casos de falsos positivos, ou seja, minimizar os casos e é calculada da seguinte maneira:

$$Precisão = \frac{TP}{TP + FP}$$

- **Revocação (*recall*):** mede a proporção de transações fraudulentas que foram corretamente identificadas em relação ao total de transações fraudulentas. É uma métrica importante quando o objetivo é minimizar o número de falsos negativos, ou seja, garantir que o modelo identifique corretamente o maior número possível de transações fraudulentas. É calculado da seguinte maneira:

$$Revocação = \frac{TP}{TP + FN}$$

- **Pontuação f-1 (*f-1 score*):** combina a precisão com a revocação numa única medida. Esta métrica fornece uma avaliação geral do desempenho do modelo na deteção de fraudes e é útil quando se deseja equilibrar essas duas métricas. É calculada da seguinte maneira:

$$Pontuação\ f-1 = \frac{2 * Precisão * Revocação}{Precisão + Revocação}$$

O modelo com melhor performance será analisado segundo a métrica de performance pontuação f-1. Como referido anteriormente, a pontuação f-1 é uma métrica que combina a precisão e a revocação numa única medida, e é útil quando o objetivo é encontrar um equilíbrio entre estas duas métricas (Tasnim et al., 2022). Num sistema de deteção de fraude, o objetivo principal é não só minimizar os falsos negativos, fraudes não identificadas, como também evitar falsos positivos, transações legítimas classificadas como fraude. Sendo as métricas relacionadas com a diminuição do número de falsos negativos e falsos positivos, a revocação e a precisão, respetivamente, é importante considerar os resultados obtidos destas duas métricas para determinar o melhor modelo (Tasnim et al., 2022). Quanto maior for a revocação, menor será o número de casos de transações fraudulentas consideradas como não fraude. Por outro lado, quanto maior for a precisão, menor será a quantidade de transações legítimas que são classificadas como fraude.

Desta forma, ao considerar a pontuação f-1 como métrica principal, é possível avaliar o desempenho do modelo de forma equilibrada pois encontra-se um equilíbrio entre identificar corretamente as fraudes e minimizar o número de falsos positivos, permitindo então determinar o melhor modelo aplicado em sistemas de deteção de fraudes.

4.1.1 Logistic Regression

O primeiro algoritmo utilizado para treinar o modelo foi o LR. Como referido em 2.3.1, a regressão logística é um método estatístico que utiliza a função logística para modelar a relação entre as variáveis independentes e a probabilidade de ocorrência de um evento. Para além disso, o LR é também um algoritmo que tem a capacidade de lidar com problemas de classificação binária (Zou et al., 2019).

O LR foi aplicado aos conjuntos de dados descritos em 3.2.1.5 e 3.2.1.6, e os resultados obtidos encontram-se representados nas tabela seguintes, tabelas Tabela 14Tabela 15.

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A]	0.96487	0.25477	0.00650	0.01269
Conjunto [B]	0.75368	0.0880	0.65170	0.15512
Conjunto [C]	0.75213	0.0873	0.65056	0.15407

Tabela 14 – Resultados do LR nos conjuntos de dados não normalizados

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A] normalizado	0.96555	0.53752	0.05011	0.09167
Conjunto [B] normalizado	0.45840	0.02021	0.30778	0.03794
Conjunto [C] normalizado	0.82738	0.13522	0.73678	0.22851

Tabela 15 - Resultados do LR nos conjuntos de dados normalizados

É possível verificar que o conjunto [A] obteve melhores resultados, com e sem normalização, quanto à métrica de exatidão, com cerca de 96,5%. Este conjunto, que representa os dados sem qualquer técnica de *oversampling* ou *undersampling* aplicada, obteve maior exatidão e precisão que os outros dois conjuntos em ambos os casos. Conclui-se também que o conjunto [A] obteve melhores resultados quanto à precisão, onde os valores subiram de 25,47% para 53,75%. Ainda assim, a pontuação f-1 obtida em ambas foi muito baixa, com apenas 9,16% no conjunto normalizado. Infere-se também que o conjunto [C] melhorou os seus resultados nas quatro métricas com a normalização do conjunto. Esta situação era esperada, pois tal como referido em 3.2.1.6, o LR é um algoritmo sensível à escala onde é necessário garantir que os coeficientes atribuídos a cada variável estejam na mesma escala.

Contrariamente ao conjunto [A] e [C], o conjunto de dados [B] diminuiu a performance dos resultados obtidos quando aplicado ao conjunto normalizado. Uma possível justificação para este acontecimento é o impacto da técnica SMOTE na distribuição dos dados pois, mesmo no conjunto não normalizado, verificou-se uma precisão muito baixa de apenas 8,8%. O SMOTE gera dados sintéticos da classe minoritária para equilibrar a distribuição dos dados e, ao gerar mais de 380 000 casos sintéticos, pode ter introduzido ruído que resultou num desempenho inferior. Para além disso, a normalização pode tornar o modelo mais sensível a *outliers* onde e caso o SMOTE se gerar dados sintéticos próximos destes outliers, a normalização pode afetar negativamente o desempenho do modelo ao amplificar o impacto desses *outliers*.

Em síntese, o conjunto de dados [A] normalizado, foi o conjunto que obteve melhores resultados quando aplicado o algoritmo de LR e nas quatro métricas de performance avaliadas, comparativamente aos restantes conjuntos de dados.

4.1.2 Random Forest

O algoritmo RF permite lidar com problemas de classificação complexos, como já referido em 2.3.1. O RF é um conjunto de árvores de decisão, onde cada árvore é treinada numa amostra aleatória de dados e as previsões são feitas através de uma votação majoritária das árvores individuais (Rodrigues et al., 2022). O algoritmo permite também lidar com conjuntos de dados de grandes dimensões e grande quantidade de atributos, como é o caso, sem comprometer a eficiência computacional (Gracia et al., 2021).

O RF foi aplicado aos conjuntos de dados descritos em 3.2.1.5 e 3.2.1.6, e os resultados obtidos encontram-se representados nas tabelas seguintes, tabelas Tabela 16Tabela 17.

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A]	0.98045	0.94578	0.46250	0.63121
Conjunto [B]	0.97998	0.91199	0.46869	0.61917
Conjunto [C]	0.87357	0.19265	0.82855	0.31261

Tabela 16 – Resultados do RF nos conjuntos de dados não normalizados

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A] normalizado	0.96887	0.96916	0.10395	0.18813
Conjunto [B] normalizado	0.03710	0.03478	0.98751	0.06722
Conjunto [C] normalizado	0.86374	0.18200	0.82951	0.31314

Tabela 17 – Resultados do RF nos conjuntos de dados normalizados

Com os resultados obtidos podemos verificar que, nos conjuntos de dados normalizados, embora a exatidão e precisão do conjunto [A] sejam razoáveis, as restantes métricas diminuíram significativamente. Isto sugere que, tal como previsto, a normalização dos dados aplicada ao algoritmo do RF pode afetar negativamente a capacidade do modelo em identificar corretamente amostras positivas.

Conclui-se também que, no geral, o conjunto [A] não normalizado tem um desempenho geral melhor em termos de exatidão, precisão e pontuação f-1 em comparação com os conjuntos [B] e [C]. O valor mais baixo obtido por este conjunto é referente à revocação, com um valor de 0.46250, o que indica que o modelo tem uma taxa considerável de falsos negativos, ou seja, não está a identificar corretamente todas as instâncias positivas. Infere-se também que o conjunto [B] não normalizado obteve valores muito semelhantes ao conjunto [A], com diferenças mínimas na exatidão, precisão e pontuação f-1, e melhores resultados em comparação com o conjunto [C]. Isto significa que, com o algoritmo do RF, a técnica de *oversampling SMOTE*, obteve melhores resultados em comparação com a técnica de *undersampling RandomUnderSampler*.

4.1.3 Support vector machine

O algoritmo de SVM permite encontrar um hiperplano de separação entre duas classes, mapeando os dados num espaço de características de alta dimensão. É um método bastante utilizado em problemas de classificação e regressão e que é capaz de lidar tanto com dados linearmente separáveis como com dados não lineares (Anowar & Sadaoui, 2020). Além disso, o SVM tem a capacidade de lidar com problemas de alta dimensionalidade e é eficaz em casos onde o conjunto de dados se encontra desequilibrado (Zhou et al., 2019).

O SVM foi aplicado aos conjuntos de dados descritos em 3.2.1.5 e 3.2.1.6, e os resultados obtidos encontram-se nas tabelas seguintes, tabelas Tabela 18Tabela 19.

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A]	0.95530	0.95317	0.11720	0.22105
Conjunto [B]	0.77264	0.14627	0.69785	0.24184
Conjunto [C]	0.32210	0.03751	0.75174	0.07145

Tabela 18 – Resultados do SVM nos conjuntos de dados não normalizados

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A] normalizado	0.97034	0.97550	0.14901	0.25853
Conjunto [B] normalizado	0.86971	0.15876	0.77501	0.26354
Conjunto [C] normalizado	0.85855	0.16439	0.75353	0.26990

Tabela 19 – Resultados do SVM nos conjuntos de dados normalizados

É possível observar que a normalização do conjunto de dados teve um efeito positivo na performance do modelo, levando a um melhor desempenho por parte do mesmo. Este resultado era esperado pois o algoritmo SVM é sensível à escala dos dados, como referido em 3.2.1.6.

Nos três conjuntos de dados observou-se uma melhoria em todas as métricas de performance avaliadas, com especial destaque para o conjunto [C] que viu todas as métricas evoluírem com o conjunto normalizado. A exatidão subiu de 32,21% para 85,85%, a precisão de 3,75% para 16,43% e a pontuação f-1 de 7,14% para 26,99%, sendo este o valor mais elevado da métrica alvo nos três conjuntos. Ainda assim, são valores relativamente baixos, o que permite inferir que o algoritmo tem bastante dificuldade em classificar corretamente as transações. Quanto ao conjunto [A] normalizado, este obteve uma exatidão de 97,03% e uma precisão de 97,55%, o que indica que das transações classificadas como fraude pelo modelo,

aproximadamente 97,5% delas são realmente fraude. No entanto, este conjunto apresenta um valor significativamente baixo de revocação, o que indica que o modelo tem dificuldade em identificar corretamente os casos positivos, ou seja, classifica transações que são fraudulentas como não fraude. Já o conjunto [B] normalizado, apresenta uma revocação elevada, com cerca de 77,5%, mas uma precisão baixa, o que indica que existe uma quantidade significativa de falsos positivos.

4.1.4 K-Nearest neighbours

Outro algoritmo utilizado para treinar o modelo foi o KNN. É um algoritmo que classifica novas amostras com base na maioria das classes dos *k* vizinhos mais próximos. A sua principal vantagem é a simplicidade e facilidade de implementação, para além de que não faz suposições sobre a distribuição dos dados e pode ser aplicado a problemas de classificação ou de regressão (Almohaimeed & Gampa, 2019).

Após realizar vários testes e avaliar o desempenho do algoritmo KNN com diferentes valores de *k*, verificou-se que o mais adequado para obter os melhores resultados foi com *k* igual a 2. O k representa o número de vizinhos mais próximos considerados pelo algoritmo para classificar uma nova instância. Ao analisar as métricas de performance, verificou-se que *k* igual a 2 proporcionou um equilíbrio entre essas métricas, resultando num desempenho geral superior a outros valores de *k* testados.

O KNN foi assim aplicado aos conjuntos de dados mencionados em 3.2.1.5 e 3.2.1.6, e os resultados obtidos encontram-se representados na tabelas seguintes, tabelas Tabela 20 e Tabela 21.

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A]	0.96527	0.49759	0.08410	0.14389
Conjunto [B]	0.89799	0.14079	0.38018	0.20548
Conjunto [C]	0.81765	0.08720	0.44948	0.14607

Tabela 20 – Resultados do KNN nos conjuntos não normalizados

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A] normalizado	0.97512	0.90277	0.31722	0.46948
Conjunto [B] normalizado	0.97141	0.57476	0.67659	0.62153
Conjunto [C] normalizado	0.89270	0.20331	0.716935	0.31678

Tabela 21 – Resultados do KNN nos conjuntos normalizados

É possível observar que a normalização do conjunto de dados levou a melhorias no desempenho do algoritmo do KNN. Houve um aumento na exatidão, precisão, revocação e pontuação f-1 nos conjuntos normalizados, indicando que o modelo está a classificar corretamente uma proporção maior de amostras positivas, reduzindo o número de falsos positivos e equilibrando melhor a precisão e a revocação. Infere-se assim que a normalização dos dados teve um impacto positivo no desempenho do algoritmo KNN nos conjuntos de dados fornecidos, tal como esperado.

Em todos os conjuntos verificou-se valores de exatidão bastante elevados, com destaque para o conjunto [A] normalizado com 97.51%. Em termos de precisão, apenas o conjunto [A] normalizado teve um valor significativamente elevado, o que significa que as técnicas de *oversampling* e *undersampling* aplicadas não foram vantajosas quando aplicadas ao algoritmo do KNN. Quanto à revocação, verificou-se que o conjunto [B] e [C] normalizados obtiveram melhores valores quanto a esta métrica, indicando que o modelo está a identificar uma proporção maior de transações fraudulentas e a minimizar a quantidade de falsos negativos. No entanto, uma revocação maior é geralmente acompanhada por uma diminuição na precisão, ou sejam uma proporção maior de falsos positivos, como se verifica pelos resultados obtidos. Quanto à métrica alvo, pontuação f-1, o conjunto [B] normalizado obteve a melhor performance com 62,15%.

4.1.5 Decision Tree

O algoritmo do DT permite dividir iterativamente o conjunto de dados com base nos valores dos atributos, formando uma estrutura hierárquica de decisão. Por um lado, cada nó interno da árvore representa uma condição sobre um atributo, enquanto que as folhas representam as classes ou valores de saída (Mustaqim et al., 2021). É um algoritmo que com base na probabilidade dos acontecimentos que já aconteceram, prevê e seleciona a melhor solução possível (Zhou et al., 2019).

O DT foi aplicado aos conjuntos de dados descritos em 3.2.1.5 e 3.2.1.6, e os resultados obtidos encontram-se representados nas tabelas seguintes, tabelas Tabela 22 e Tabela 23.

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A]	0.96856	0.54212	0.60501	0.57184
Conjunto [B]	0.95962	0.43269	0.52643	0.47497
Conjunto [C]	0.77327	0.11246	0.80299	0.19729

Tabela 22 – Resultados do DT nos conjuntos não normalizados

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A] normalizado	0.90519	0.11475	0.25801	0.15885
Conjunto [B] normalizado	0.77327	0.11245	0.80299	0.19728
Conjunto [C] normalizado	0.76352	0.11274	0.79462	0.19778

Tabela 23 – Resultados do DT nos conjuntos normalizados

Com os resultados obtidos observa-se que, o conjunto de dados normalizados, obteve uma queda geral no desempenho do modelo em todas as métricas, tal como esperado, pois o algoritmo de DT não é sensível à escala e os dados não necessitam de ser normalizados. Ainda assim, os resultados do conjunto [C] normalizado foram muito semelhantes ao conjunto [C] não normalizado, e a revocação do conjunto [B] normalizado foi superior ao conjunto [B] não normalizado com cerca de 80.29%, indicando assim que o modelo continua a identificar a maioria das instâncias positivas após a normalização.

De resto, verifica-se, no geral, melhores resultados nos conjuntos de dados não normalizados, com particularidade no conjunto [A] que obteve melhor performance em todas as métricas utilizadas e uma pontuação f-1 de 57,18%. Ainda assim, os valores obtidos são moderadamente baixos nos restantes conjuntos de dados. Infere-se assim que as técnicas de *SMOTE* e *RandomUnderSampler* não foram vantajosas quando aplicadas ao algoritmo DT.

4.1.6 Long Short-Term Memory

Como mencionado em 2.3.1, as LSTM são uma arquitetura de rede neuronal recorrente (RNN) que foi projetada para lidar com problemas que envolvem dependências de longo prazo entre os elementos de uma sequência (Wang et al., 2020). Este método introduziu as células de memórias as RNN e o mecanismo de controlo de fluxo de informação, que permite combater o problema do gradiente desvanecente. Cada célula de memória é responsável por armazenar informações por longos períodos de tempo. A capacidade de lembrar informações relevantes por longos períodos de tempo torna as LSTM adequadas para tarefas que envolvem sequências

de dados e permite também capturar padrões complexos e detetar anomalias (Venna et al., 2019).

O algoritmo de LSTM foi aplicado aos conjuntos de dados descritos em 3.2.1.5 e 3.2.1.6, e os resultados obtidos encontram-se representados nas tabelas seguintes, tabelas Tabela 24 eTabela 25.

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A]	0.97226	0.72995	0.31836	0.44336
Conjunto [B]	0.94895	0.33987	0.50154	0.40517
Conjunto [C]	0.75487	0.09777	0.73710	0.17264

Tabela 24 – Resultados do LSTM nos conjuntos não normalizados

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A] normalizado	0.96828	0.88169	0.11777	0.20778
Conjunto [B] normalizado	0.95627	0.29561	0.48209	0.36649
Conjunto [C] normalizado	0.72519	0.08604	0.71921	0.15370

Tabela 25 - Resultados do LSTM nos conjuntos normalizados

A função de ativação "*sigmoid*" foi utilizada na implementação do algoritmo LSTM que é comumente usada em problemas de classificação binária. Esta função mapeia os valores de saída para o intervalo entre 0 e 1, o que é útil para interpretar a saída do modelo como uma probabilidade (K. & K., 2022). Um valor próximo de 0 indica baixa probabilidade de a transação ser fraude, enquanto que um valor próximo de 1 significa uma elevada probabilidade de ser fraude. Sendo assim, definiu-se um *threshold* de 0.5, onde as transações com probabilidade acima deste limite foram classificadas como fraude e aquelas abaixo são classificadas como não fraude.

Com base nos resultados obtidos, conclui-se que, no geral, o modelo LSTM obteve um desempenho razoável no conjunto [A] não normalizado, com uma elevada taxa de exatidão, 97.23%, e uma precisão decente com 72.99%. Já a pontuação f-1 foi superior aos restantes conjuntos, sem e com normalização, com 44,34%, ainda que seja um valor relativamente baixo. Infere-se também que a normalização dos dados não melhorou significativamente o desempenho do modelo LSTM nos três conjuntos de dados pois o algoritmo LSTM não é sensível à escala como o LR, KNN ou SVM. Este possui mecanismo internos, como células de memória e portões, que ajudam a aprender e lembrar padrões ao longo de sequência temporais (Venna et

al., 2019). Ainda assim verificou-se que a precisão aumentou no caso do conjunto [A], contudo, a revocação e a pontuação f-1 continuam baixos. Isso sugere que outros ajustes no modelo ou mesmo no pré-processamento dos dados podem ser necessários para melhorar o seu desempenho.

4.1.7 Convolutional Neural Networks

As redes neuronais Convolucionais (CNN), ou *Convolutional Neural Network* em inglês, são um tipo especializado de arquitetura *feed-forward* de redes neuronais profundas. A ideia principal de uma CNN é a aplicação de filtros Convolucionais em diferentes partes dos dados de entrada, permitindo a extração de caraterísticas relevantes de forma hierárquica. Esses filtros, também conhecidos como *kernels*, são pequenas janelas que percorrem a entrada e são multiplicados ponto a ponto com valores de entrada em cada localização (T. Nguyen et al., 2020).

As CNN tornaram-se algoritmos de última geração para visão computacional, processamento de linguagem natural e problemas de reconhecimento de padrões. Para além disso, são altamente eficazes na extração automática de características relevantes dos dados de entrada, permitindo que as redes neuronais aprendam representações hierárquicas e obtenham um desempenho superior em várias tarefas de análise de imagem (Ghosh et al., 2020).

O algoritmo de CNN foi aplicado aos conjuntos de dados descritos em 3.2.1.5 e 3.2.1.6, e os resultados obtidos encontram-se representados nas tabelas seguintes, Tabela 26 e Tabela 27.

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A]	0.98416	0.72953	0.40589	0.52158
Conjunto [B]	0.97324	0.74572	0.34732	0.47391
Conjunto [C]	0.90868	0.22603	0.67317	0.33843

Tabela 26 – Resultados do CNN nos conjuntos não normalizados

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A] normalizado	0.96945	0.93563	0.12774	0.22473
Conjunto [B] normalizado	0.19005	0.04013	0.97510	0.07710
Conjunto [C] normalizado	0.84579	0.15034	0.74182	0.25028

Tabela 27 - Resultados do CNN nos conjuntos normalizados

Assim como no algoritmo LSTM, foi aplicado a função de ativação "*sigmoid*" na implementação das CNN para treinar o modelo, tanto na camada convolucional como na camada densa da rede neuronal.

Com base nos resultados obtidos, observa-se que o modelo obteve um melhor desempenho com os conjuntos de dados não normalizados, o que permite concluir que as CNN podem lidar com variáveis em diferentes escalas. Neste caso específico, em que foi aplicado a um sistema de deteção de fraude, obteve-se melhores resultados nos conjuntos não normalizados que os conjuntos normalizados. Os conjuntos [A] e [B] obtiveram resultados semelhantes, com uma exatidão e precisão elevada. A nível de revocação, o resultado foi um pouco mais baixo, onde o conjunto [C] obteve a melhor classificação, com 67,31%. Já em termos de pontuação f-1, métrica principal, o conjunto [A] obteve a melhor performance com 52,16%.

4.2 Avaliação dos algoritmos

Após a aplicação dos algoritmos tradicionais, como o LR, RF, SVM, KNN e DT, assim como a implementação de abordagens *deep-learning* como redes LSTM e CNN, avaliou-se que, por meio da pontuação f-1, o melhor modelo tradicional foi o RF e o melhor modelo de *deep-learning* foi a rede neuronal CNN. Esta seleção foi decidida com base no desempenho equilibrado que os modelos demonstraram, ao combinar uma elevada precisão com uma revocação também elevada.

De um ponto de vista, o algoritmo tradicional RF, obteve uma precisão de 94,58% e uma revocação de 46,25% quando aplicado ao conjunto [A] não normalizado, o que permitiu obter uma pontuação f-1 de 63,12%, superior a qualquer outro resultado obtido pela implementação dos restantes algoritmos tradicionais. Também no conjunto [B] não normalizado, obteve-se uma pontuação f-1 de 61,92%, através de uma precisão de 91,19% e uma revocação de 46,87%. No conjunto [C], os resultados do RF apresentaram um desempenho aquém do esperado, com apenas 31,26% de pontuação f-1, o que pode indicar que o RF pode ser mais eficaz em conjuntos de dados maiores. Contudo, é importante destacar a revocação obtida neste conjunto com 82,85%, o que indica que o modelo foi capaz de identificar a grande maioria das transações fraudulentas presentes no conjunto de dados.

De outro ponto de vista, o modelo de *deep-learning* com melhor performance foi as CNN, com uma precisão de 72,95% e uma revocação de 40.58%, permitindo assim obter uma pontuação f-1 de 52,15% quando aplicado ao conjunto [A] não normalizado. Ainda que não seja um valor bastante elevado, foi o modelo que obteve melhor desempenho em comparação com as LSTM. O conjunto [B] também obteve valores semelhantes com uma precisão de aproximadamente 74,58% e uma revocação de 34,73%, obtendo assim uma pontuação f-1 de 47,39%. Por sua vez, o conjunto [C] apresenta o melhor resultado de revocação com aproximadamente 67,31%.

Estes resultados indicam que tanto o algoritmo RF como o modelo CNN, ainda que sejam os modelos com melhores resultados, podem ser aprimorados através de técnicas de otimização de hiper-parâmetros. Esta otimização pode ser conseguida através da exploração de diferentes combinações de parâmetros, como o número de árvores, a profundidade máxima, o tamanho do filtro ou até o número de camadas das CNN, de maneira a ajustar o modelo às características dos dados da forma mais precisa, e a melhorar a sua capacidade em detetar fraudes (Weerts et al., 2020).

4.3 Modelo proposto – algoritmos com hiper-parâmetros otimizados

Neste capítulo serão implementados os algoritmos que demonstraram os melhores resultados do capítulo 4.1, destacando-se o RF e a CNN. Estes algoritmos foram selecionados devido à sua capacidade comprovada de lidar com problemas de classificação em conjuntos de dados complexos. Como referido no capítulo 4.1, esta avaliação adveio da pontuação f-1 obtida pelo modelo, por ser a métrica que combina a precisão e revocação numa única medida. O objetivo num sistema de deteção de fraude prende-se com diminuir a percentagem de transações fraudulentas não identificadas, falsos negativos, e diminuir a taxa de transações legítimas classificadas incorretamente como fraudes (Tasnim et al., 2022).

Para melhorar ainda mais o desempenho do modelo, serão explorados os hiper-parâmetros específicos a estes algoritmos. Os hiper-parâmetros são parâmetros ajustáveis que determinam como o algoritmo irá aprender e como os dados serão processados durante o treino. Ao ajustar os hiper-parâmetros, é possível encontrar a configuração ideal que resulta no melhor desempenho do modelo para a tarefa específica que é, neste caso, classificar corretamente o maior número de transações possíveis (Weerts et al., 2020).

No caso do RF, que é um algoritmo de *machine learning* com base em árvores de decisão, alguns hiper-parâmetros considerados incluem (Probst et al., 2019):

1. **Número de árvores na floresta (*n_estimators*):** determina a quantidade de árvores de decisão presentes na floresta aleatória que influenciará a capacidade de generalização do modelo e a estabilidade das previsões;

2. **Profundidade máxima das árvores (*max_depth*):** controla a profundidade máxima permitida para cada árvore de decisão na floresta aleatória, controlando assim o número máximo de níveis de decisão que a árvore pode ter;

3. **Número mínimo de amostras para dividir um nó (*min_samples_split*):** define o número mínimo de amostras necessárias para dividir um nó interno de uma árvore de decisão na floresta aleatória que ajudará a regular o crescimento da árvore;

4. **Número mínimo de amostras necessárias em um nó (*min_samples_leaf*)**: controla o tamanho mínimo das folhas da árvore após a divisão, evitando que se gerem folhas muito pequenas que poderiam levar a um ajuste excessivo;

Ajustar estes hiper-parâmetros pode evitar o *overfitting* e aumentar a eficiência do modelo em relação ao conjunto de dados. Por outro lado, no caso das CNN, e assim como no RF, os hiper-parâmetros desempenham um papel fundamental no modelo. Estes hiper-parâmetros definem-se em diferentes conjuntos de hiper-parâmetros desde a arquitetura de rede neuronal, da camada convolucional, do otimizador, entre outros. Foram então considerados os seguintes hiper-parâmetros (Gafsi, 2018):

1. **Número de camadas convolucionais (*layers*):** determina quantas camadas convolucionais serão usadas na arquitetura da CNN. Cada camada é responsável por extrair características das imagens de entrada por meio da aplicação de filtros;

2. **Número e tamanho de filtros convolucionais (*kernels and kernel size*):** determina quantas características diferentes cada camada está a aprender a detetar e tamanho do filtro, que é uma pequena matriz que é deslizada ao longo dos dados de entrada, determina quantos elementos da entrada são considerados em cada etapa da convolução;

3. **Tamanho dos passos de convolução (*strides*):** define o tamanho do passo que o filtro convolucional utiliza ao percorrer a imagem de entrada. O passo de convolução afeta a sobreposição das áreas cobertas pelos filtros e pode influenciar a capacidade da CNN em capturar informações espaciais;

4. **Função de ativação (*activation function*):** estas funções são aplicadas após as operações de convolução para introduzir não linearidade nas CNN. Diferentes funções de ativação como "*ReLU*", "*Sigmoid*" ou "*Tanh*" têm diferentes propriedades e impactam a capacidade da rede em aprender representações complexas;

5. **Função perda (*loss*):** medida que representa o erro entre as previsões do modelo e os valores reais. Desempenha um papel crucial no processo de treino pois é a métrica que o otimizador tenta minimizar;

6. **Tamanho dos lotes de treino (*batch size*):** define o número de exemplos de treino que são propagados através da CNN antes de uma atualização de pesos. Usar lotes maiores pode aumentar a eficiência computacional e fornecer estimativas de gradientes mais estáveis, contudo, o tamanho do lote pode afetar a generalização do modelo e a memória necessária para efetuar o treino;

7. **Épocas de treino (*epochs*):** número de vezes que todo o conjunto de treino é apresentado ao modelo durante o processo de treino. Cada época envolve um ciclo completo ao conjunto de treino, calculando os gradientes e atualizando os pesos do modelo;

8. **Optimizador (*optimizer*):** refere-se ao algoritmo que é utilizado para otimizar os parâmetros de uma rede neuronal durante o processo de treino e é responsável por ajustar os parâmetros do modelo de maneira a diminuir a função de perda.

De forma a assegurar a melhor performance do modelo utilizou-se a técnica *Grid Search Cross-Validation* (*GridSearchCV*) onde se define um conjunto de valores possíveis para diferentes hiper-parâmetros do modelo. Depois disso, o modelo é treinado e avaliado segundo cada combinação diferente definida.

A validação cruzada é uma técnica que permite estimar o desempenho do modelo em dados não vistos ao dividir os dados de treino em dobras e alternando entre usar cada dobra como conjunto de validação e as restantes dobras como conjunto de treino(Belete & D H, 2021). O número de dobras utilizadas é também definido manualmente pelo parâmetro *cv* e significa que os dados de treino serão divididos em X partes iguais e o modelo será treinado e avaliado X vezes, utilizando sempre uma parte diferente do conjunto de validação e as outras partes como conjunto de treino (Liashchynskyi & Liashchynskyi, 2019). Este valor é definido como 5, tanto no algoritmo RF como no algoritmo CNN.

Por sua vez, o parâmetro *scoring* é também definido na técnica *GridSearchCV* e determina qual a métrica que será utilizada para avaliar o desempenho do modelo. Neste caso, o *scoring* é definido como a pontuação f-1, tanto no RF como na CNN.

Os algoritmos RF e CNN com hiper-parâmetros modificados foram implementados exclusivamente nos conjuntos de dados não normalizados, uma vez que foi neste cenário que estes apresentaram os melhores resultados. Ao focar nos dados não normalizados, a análise e classificação de transações fraudulentas demonstraram-se mais eficazes, possivelmente devido à preservação das características originais e fatores inerentes a padrões de fraude presentes nos dados.

4.3.1 *Random Forest* com hiper-parâmetros otimizados

Após diversas iterações de teste com o propósito de identificar os melhores conjuntos de hiper-parâmetros utilizando o *GridSearchCV*, constata-se que a combinação ideal difere consoante os diferentes conjuntos de dados. Os valores dos hiper-parâmetros que conduziram à melhor performance do modelo, segundo o algoritmo RF, encontra-se definidos na Tabela 28 e os resultados obtidos com estes encontram-se representados na Tabela 29.

	Conjunto [A]	Conjunto [B]	Conjunto [C]
n_estimators	800	1000	300
max_depth	*None*	5	*None*
min_samples_split	5	4	2
min_samples_leaf	2	2	1

Tabela 28 – Hiper-parâmetros do RF que obtiveram melhores resultados

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A]	0.96004	0.86361	0.70244	0.77315
Conjunto [B]	0.71310	0.81486	0.55137	0.66024
Conjunto [C]	0.56833	0.83341	0.27527	0.41458

Tabela 29 – Resultados do RF com hiper-parâmetros otimizados

Com os resultados obtidos é possível concluir que tanto a exatidão como a precisão diminuíram no modelo com hiper-parâmetros modificados, com exceção do conjunto [C] onde a precisão subiu significativamente para 83,34%. Pelo contrário e, no que toca à revocação, os conjuntos [A] e [B] subiram para 70,24% e 55,18% respetivamente, e o conjunto [C] diminuiu para 27,53%. Relativamente à pontuação f-1, métrica principal para a avaliação do modelo como referido em 4.1, os resultados demonstram que o ajuste dos hiper-parâmetros teve um impacto positivo em todos os conjuntos de dados. Este modelo foi capaz de equilibrar melhor a precisão e a revocação, resultando numa pontuação f-1 superior. Esta melhoria deve-se também ao facto se definir o *scoring* no *GridSearchCV* como "*f1*", ficando evidente que esta abordagem resultou em melhorias consoante esta métrica. No caso do conjunto [A], a pontuação f-1 subiu de 63,12% para 77,32%, no conjunto [B] de 61,92% para 66,02% e no conjunto [C] de 31,26% para 41,46%.

Infere-se também que, assim como em 4.1.2, o conjunto [A] obteve melhores resultados em comparação com os outros conjuntos de dados, comprovando mais uma vez que, com algoritmo do RF, as técnicas de *oversampling* (SMOTE) e *undersampling (RandomUnderSampler)* aplicadas não foram benéficas na obtenção de melhores resultados.

Conclui-se que a aplicação de hiper-parâmetros otimizados revelou-se vantajosa para o algoritmo RF. A seleção criteriosa destes hiper-parâmetros, alinhada com o foco na métrica de pontuação f-1, resultou em melhorias notáveis no equilíbrio entre precisão e revocação. Esta abordagem conduziu a um desempenho aprimorado do RF, permitindo que o modelo respondesse de maneira mais eficaz às tarefas de classificação das transações em fraude e não fraude.

4.3.2 *Convolutional Neural Networks* com hiper-parâmetros otimizados

Como referido no capítulo 4.3, foram aplicados hiper-parâmetros à arquitetura de *deep learning* CNN, no qual, após várias tentativas e experiências, foram encontrados os hiper-parâmetros mais eficazes e que proporcionam um melhor desempenho do modelo nos diferentes conjuntos de dados. Estes parâmetros encontram-se definidos na Tabela 30 e os resultados obtidos por estes, encontram-se representados na Tabela 31.

	Conjunto [A]	**Conjunto [B]**	**Conjunto [C]**
layers	1	1	1
kernels	64	64	64
kernel size	3	3	3
strides	1	1	1
convolutional layer activation	*sigmoid*	*sigmoid*	*sigmoid*
dense layer activation	*sigmoid*	*sigmoid*	*sigmoid*
loss	*binary_crossentropy*	*binary_crossentropy*	*binary_crossentropy*
batch size	128	256	16
epochs	15	25	10
optimizer	adam	adam	sgd

Tabela 30 - Hiper-parâmetros da CNN que obtiveram melhores resultados

	Métricas de performance			
	Exatidão	**Precisão**	**Revocação**	**Pontuação f-1**
Conjunto [A]	0.95944	0.63889	0.54960	0.59172
Conjunto [B]	0.89235	0.60521	0.57354	0.58933
Conjunto [C]	0.92168	0.24913	0.62420	0.35613

Tabela 31 – Resultado da CNN com hiper-parâmetros otimizados

Como referido em 4.3, o *GridSearchCV* foi a técnica adotada que permitiu encontrar a combinação ideal de hiper-parâmetros que resultou no melhor desempenho do modelo. No entanto, nem todos os hiper-parâmetros podem ser incluídos diretamente na pesquisa de grade devido a limitações técnicas. Neste caso, os hiper-parâmetros relacionados com a arquitetura da rede, como o número de camadas, o número e tamanho dos filtros, o tamanho dos passos de convolução, assim como as funções de ativação e perda, foram mantidas consistentes nos três conjuntos de dados. Esta escolha foi guiada pela necessidade de explorar as variações dos hiper-parâmetros restantes, como o tamanho do lote, número de épocas de

treino e otimizadores, enquanto se manteve uma base uniforme para a arquitetura de rede. Assim, a uniformidade destes elementos entre os diferentes conjuntos permitiu uma comparação mais precisa e justa dos resultados obtidos.

Relativamente aos resultados obtidos, infere-se que, no geral, os resultados com hiper-parâmetros otimizados são melhores em termos de pontuação f-1, que era o objetivo pretendido. No conjunto [A], a exatidão e a precisão diminuíram, contudo, a revocação aumentou de 40,59% para 54,96%, o que permitiu que a pontuação f-1 também tenha aumentado de 52,16% para 59,17%. O conjunto [B] teve um comportamento semelhante ao conjunto [A] e viu a pontuação f-1 aumentar de 47,39% para 58,93%. Por fim, no conjunto [C], apesar da revocação ter diminuído, todas as outras métricas aumentaram. A exatidão cresceu de 90,87% para 92,17%, a precisão de 22,60% para 24,91% e, por sua vez, a pontuação f-1 de 33,84% para 35,61%.

Conclui-se então que, e assim como no algoritmo RF, a aplicação de hiper-parâmetros otimizados trouxe vantagens significativas às CNN. O modelo tornou-se mais eficaz em tarefas de classificação, abrangendo tantos as instâncias positivas como as negativas.

4.4 Discussão e comparação de resultados com a literatura

Neste subcapítulo será realizada uma comparação minuciosa entre os resultados obtidos no âmbito deste estudo e os resultados previamente apresentados pelos artigos estudados e mencionados no capítulo 2.4, referente ao estado de arte. Através desta comparação, pretende-se analisar as semelhanças e discrepâncias entre os resultados obtidos, promovendo assim uma análise mais completa dos mesmos. Será feita também uma avaliação e discussão dos resultados obtidos pelo presente trabalho.

Com este propósito, foi elaborada uma tabela com os resultados de todos os algoritmos testados nos diferentes conjuntos de dados, com e sem hiper-parâmetros otimizados, assim como os resultados dos algoritmos estudados no capítulo 2.4 ( ). Para facilitar a compreensão dos dados apresentados na tabela Tabela 32, foi adotada uma convenção na qual a métrica exatidão é representada por “E”, precisão por “P”, revocação por “R” e pontuação f-1 por “F1”. É importante referir que nos artigos (Raja et al., 2021) e (T. Nguyen et al., 2020), os autores optaram por não mencionar em concreto os valores das métricas utilizadas. Invés disso, optaram por exibir gráficos, identificando os algoritmos que obtiveram melhor desempenho, além de terem fornecido uma análise das tendências e comparações entre os diferentes algoritmos. Por esta razão, a não inclui os valores exatos para as métricas analisadas nos artigos (Raja et al., 2021) e (T. Nguyen et al., 2020). Está também assinalado a negrito os algoritmos que obtiveram melhor performance segundo a métrica alvo, pontuação f-1.

	LR	RF	SVM	KNN	DT	LSTM	CNN	RF c/ HP	CNN c/ HP	Outro
Conjunto [A]	E: 0.96487 P: 0.25477 R: 0.00650 F1: 0.01269	E: 0.98045 P: 0.94578 R: 0.46250 F1: 0.63121	E: 0.95530 P: 0.95317 R: 0.11720 F1: 0.22105	E: 0.96527 P: 0.49759 R: 0.08410 F1: 0.14389	E: 0.96856 P: 0.54212 R: 0.60501 F1: 0.57184	E: 0.97226 P: 0.72995 R: 0.31836 F1: 0.44336	E: 0.98416 P: 0.72953 R: 0.40589 F1: 0.52158	**E: 0.96004** **P: 0.86361** **R: 0.70244** **F1: 0.77315**	E: 0.95944 P: 0.63889 R: 0.54960 F1: 0.59172	-
Conjunto [B]	E: 0.75368 P: 0.0880 R: 0.65170 F1: 0.15512	E: 0.97998 P: 0.91199 R: 0.46869 F1: 0.61917	E: 0.77264 P: 0.14627 R: 0.69785 F1: 0.24184	E: 0.89799 P: 0.14079 R: 0.38018 F1: 0.20548	E: 0.95962 P: 0.43269 R: 0.52643 F1: 0.47497	E: 0.94895 P: 0.33987 R: 0.50154 F1: 0.40517	E: 0.97324 P: 0.74572 R: 0.34732 F1: 0.47391	**E: 0.71310** **P: 0.81486** **R: 0.55137** **F1: 0.66024**	E: 0.89235 P: 0.60521 R: 0.57354 F1: 0.58933	-
Conjunto [C]	E: 0.75213 P: 0.0873 R: 0.65056 F1: 0.15407	E: 0.87357 P: 0.19265 R: 0.82855 F1: 0.31261	E: 0.32210 P: 0.03751 R: 0.75174 F1: 0.07145	E: 0.81765 P: 0.08720 R: 0.44948 F1: 0.14607	E: 0.77327 P: 0.11246 R: 0.80299 F1: 0.19729	E: 0.75487 P: 0.09777 R: 0.73710 F1: 0.17264	E: 0.90868 P: 0.22603 R: 0.67317 F1: 0.33843	**E: 0.56833** **P: 0.83341** **R: 0.27527** **F1: 0.41458**	E: 0.92168 P: 0.24913 R: 0.62420 F1: 0.35613	-
Conjunto [A] normalizado	E: 0.96555 P: 0.53752 R: 0.05011 F1: 0.09167	E: 0.96887 P: 0.96916 R: 0.10395 F1: 0.18813	E: 0.97034 P: 97550 R: 0.14901 F1: 0.25853	**E: 0.97512** **P: 0.90277** **R: 0.31722** **F1: 0.46948**	E: 0.90519 P: 0.11475 R: 0.25801 F1: 0.15885	E: 0.96828 P: 0.88169 R: 0.11777 F1: 0.20778	E: 0.96945 P: 0.93563 R: 0.12774 F1: 0.22473	-	-	-
Conjunto [B] normalizado	E: 0.45840 P: 0.02021 R: 0.30778 F1: 0.03794	E: 0.03710 P: 0.03478 R: 0.98751 F1: 0.06722	E: 0.86971 P: 0.15876 R: 0.77501 F1: 0.26354	E: 0.97141 **P: 0.57476** **R: 0.67659** **F1: 0.62153**	E: 0.77327 P: 0.11245 R: 0.80299 F1: 0.19728	E: 0.95627 P: 0.29561 R: 0.48209 F1: 0.36649	E: 0.19005 P: 0.04013 R: 0.97510 F1: 0.07710	-	-	-
Conjunto [C] normalizado	E: 0.82738 P: 0.13522 R: 0.73678 F1: 0.22851	E: 0.86374 P: 0.18200 R: 0.82951 F1: 0.31314	E: 0.85855 P: 0.16439 R: 0.75353 F1: 0.26990	**E: 0.89270** **P: 0.20331** **R: 0.716935** **F1: 0.31678**	E: 0.76352 P: 0.11274 R: 0.79462 F1: 0.19778	E: 0.72519 P: 0.08604 R: 0.71921 F1: 0.15370	E: 0.84579 P: 0.15034 R: 0.74182 F1: 0.25028	-	-	-
(Mathew et al., 2022)	E: 0.99 P: 0.86 R: 0.57 F1: 0.68	E: 0.99 P: 0.94 R: 0.82 F1: 0.87	-	**E: 0.94** **P: 0.95** **R: 0.95** **F1: 0.95**	E: 0.99 P: 0.73 R: 0.82 F1: 0.77	-	-	-	-	-

(Abhirami et al., 2021)	**E: 0.96446 P: 0.98837 R: 0.95604**	E: 0.92893 P: 0.95505 R: 0.93406	E: 0.93908 P: 0.97647 R: 0.91208	E: 0.94416 P: 0.96551 R: 0.92307	E: 0.92385 P: 0.89583 R: 0.94505	-	-	-	-	-
(Raja et al., 2021)	-	**Gráficos das Figura 8 e Figura 9**	-	-	-	-	-	-	-	-
(Li, 2022)	E ≈ 0.800	-	E ≈ 0.850	-	-	-	-	-	-	**E ≈ 0.970**
(Saputra & Suharjito, 2019)	-	**E: 0.950 P: 0.955 R: 0.550 F1: 0.698**	-	-	E: 0.910 P: 0.598 R: 0.590 F1: 0.568	-	-	-	-	-
(Saputra & Suharjito, 2019) com SMOTE	-	**E: 0.950 P: 0.805 R: 0.581 F1: 0.943**	-	-	E: 0.910 P: 0.916 R: 0.604 F1: 0.912	-	-	-	-	-
(T. Nguyen et al., 2020)	-	Gráficos das Figura 20 e Figura 21	-	-	-	**Gráficos das Figura 20 e Figura 21**	Gráficos das Figura 20 e Figura 21	-	-	-

Tabela 32 - Resultados obtidos vs Resultados dos artigos estudados

Dado o exposto, é possível concluir que:

- O algoritmo RF foi utilizado em 5 (Mathew et al., 2022; Abhirami et al., 2021; Raja et al., 2021; Saputra & Suharjito, 2019; T. Nguyen et al., 2020) dos 6 artigos estudados;

- O algoritmo LR e DT foram utilizados em 3 dos 6 artigos estudados, (Mathew et al., 2022; Abhirami et al., 2021; Li, 2022) e (Mathew et al., 2022; Abhirami et al., 2021; Saputra & Suharjito, 2019) respetivamente;

- O algoritmo SVM e KNN foram utilizados em 2 dos 6 artigos estudados, (Abhirami et al., 2021; Li, 2022) e (Mathew et al., 2022; Abhirami et al., 2021) respetivamente;

- Os algoritmos LSTM e CNN apenas foram abordados no artigo (T. Nguyen et al., 2020) por serem abordagens de *deep-learning*, ao contrário dos algoritmos de *machine learning* tradicionais;

- O artigo (Li, 2022) não foi incluído na discussão do melhor algoritmo de *machine learning* por utilizar exclusivamente a métrica exatidão para avaliação do modelo, não calculando qualquer outra métrica;

- Todos os artigos estudados utilizam as diferentes métricas escolhidas para fins de avaliação. Em (Li, 2022), o autor utiliza apenas a métrica de exatidão para avaliar os diferentes algoritmos;

- Relativamente à técnica de *oversampling*, SMOTE:
  - Em (Saputra & Suharjito, 2019) os autores inferem que foi eficaz ao permitir lidar com o desequilíbrio dos dados e permitiu também aumentar o desempenho dos modelos de classificação. Por sua vez, em (T. Nguyen et al., 2020), os autores referem que esta técnica permitiu aumentar a revocação do modelo, mas a precisão e pontuação f-1 diminuíram drasticamente. Esta diminuição pode derivar do facto de que o SMOTE cria instâncias de transações fraudulentas sintéticas que se sobrepõem às transações não fraudulentas e, por isso, alteram assim o limite de decisão;
  - Com os algoritmos LR, SVM e KNN, e sendo a métrica alvo a pontuação f-1, como referido em 4.1, esta técnica permitiu obter melhor performance em determinados conjuntos de dados;
  - Nos restantes algoritmos RF, DT, LSTM e CNN, constata-se que não proporcionou vantagens notáveis. Assim como referido em (T. Nguyen et al., 2020), a criação de transações sintéticas pelo SMOTE pode introduzir informações artificiais nos dados, o que pode não ser benéfico para algoritmos que dependem de padrões específicos nos dados originais. Conclui-se então que, no âmbito deste estudo, a aplicação desta técnica não conferiu benefícios significativos.

- Relativamente à técnica de *undersampling*, *RandomUnderSampler*:
  - Esta técnica permitiu obter melhores resultados apenas no algoritmo LR, nos conjuntos com e sem normalização;
  - Em (T. Nguyen et al., 2020), os autores referem que este método não permitiu obter melhores resultados em comparação com outros conjuntos de dados (Figura 20);
  - Em (Abhirami et al., 2021) a técnica foi necessária para resolver o problema de não ser possível inferir nenhuma conclusão com os resultados antes do pré-processamento, porque os valores obtidos eram todos muito próximos;
  - Infere-se assim que, a implementação do *RandomUnderSampler,* no presente trabalho, não foi vantajosa para aprimorar o desempenho dos modelos de classificação. Esta abordagem pode resultar na perda de informações importantes ao reduzir aleatoriamente registos da classe

majoritária, neste caso, transações que não correspondem a fraude, prejudicando assim a capacidade do algoritmo em compreender e aprender padrões e comportamentos nos dados originais.

- Relativamente ao melhor modelo de *machine learning* em sistemas de classificação de transações fraudulentas:
  - Em (Mathew et al., 2022), os autores referem o RF e DT como os melhores algoritmos, ao obterem melhores resultados segundo a métrica de exatidão. Se considerarmos a métrica alvo do presente trabalho, pontuação f-1, o KNN é o modelo com melhor performance;
  - Em (Abhirami et al., 2021), os autores referem que o LR obteve, no geral, os melhores resultados, considerando então como o melhor algoritmo entre cinco testados. Em termos de pontuação f-1, o LR foi também o que obteve melhor performance;
  - Em (Raja et al., 2021), os autores concluem que o algoritmo RF juntamente com o mecanismo de *feedback* obtém os melhores resultados;
  - Em (Saputra & Suharjito, 2019) os autores não concluíram qual foi o modelo de *machine learning* com melhores resultados, uma vez que o foco principal do estudo passou por investigar e analisar a eficácia e os impactos da técnica SMOTE. Ainda assim, com os resultados obtidos, é possível inferir que o algoritmo RF obteve a melhor pontuação f-1 com 94,3%;
  - Em (T. Nguyen et al., 2020) os autores não referem os valores específicos obtidos pelo RF e pelas diferentes abordagens de *deep learning* aplicadas, contudo, constatam o LSTM apresenta um desempenho ligeiramente superior entre todos os algoritmos testados.
  - No presente trabalho, o algoritmo RF e a abordagem de *deep-learning* CNN, obtiveram os melhores resultados com uma pontuação f-1 de 63,12% e 52,16%, respetivamente. Estes algoritmos foram posteriormente implementados com hiper-parâmetros otimizados. A otimização dos hiper-parâmetros permitiu aumentar a pontuação f-1 para 77,32% no RF e 59,17% na CNN.

- Relativamente à utilização de hiper-parâmetros modificados:
  - A implementação de algoritmos com hiper-parâmetros otimizados revelou-se bastante vantajosa nos modelos de classificação pretendidos, resultando em melhorias nos resultados obtidos. Através da modificação desses hiper-parâmetros, é possível ajustar o modelo de forma mais precisa às características específicas dos dados em questão. A utilização do mecanismo *GridSearchCV* também desempenhou um papel crucial neste processo. Através da exploração

de diversas combinações de hiper-parâmetros, permitiu identificar configurações mais adequadas, otimizando assim a capacidade do modelo em classificar as transações como fraude ou não fraude.

4.5 Sumário

Foram implementados cinco algoritmos de *machine learning* tradicional, entre elas LR, RF, SVM, KNN e DT, e duas arquiteturas de *deep-learning* como as LSTM e CNN. Estes métodos foram treinados segundo diferentes conjuntos de dados, referidos em 3.2.1.5, de forma a avaliar também o impacto das técnicas de *oversampling*, SMOTE, e *undersampling*, *RandomUnderSampler*, nos modelos de classificação.

Para avaliação da eficácia do modelo, utilizou-se as métricas de performance exatidão, precisão, revocação e pontuação f-1. Estas métricas de avaliação são calculadas através de uma tabela que demonstra o desempenho de um modelo de classificação, denominada matriz de confusão. O modelo com melhor performance foi analisado segundo a métrica de performance pontuação f-1. Num sistema de deteção de fraudes é importante não só minimizar o número de falsos negativos, transações fraudulentas que não foram identificadas, como também evitar falsos positivos, transações legítimas que foram classificadas como fraude. Sendo a métrica relacionada com o número de falsos negativos a revocação, e o número de falsos positivos a precisão, era importante estabelecer um equilíbrio entre as duas. A pontuação f-1, ao combinar estas duas métricas, foi então considerada a melhor indicadora da performance geral do modelo, e assim, a métrica que determinou os melhores algoritmos de *machine learning*.

Os métodos de *machine learning* RF e CNN, ao obterem os melhores resultados, foram posteriormente modificados através da otimização dos hiper-parâmetros. Esta otimização permitiu aumentar o desempenho destes algoritmos, pois foi possível encontrar combinações de parâmetros ideais que proporcionaram melhores resultados nos modelos de classificação pretendidos.

Ao analisar os resultados obtidos nos conjuntos de dados [A], [B] e [C] constatou-se que o conjunto [A], composto pelos dados originais, demonstrou um desempenho superior com os algoritmos implementados. Tanto a técnica de *SMOTE* aplicada ao conjunto [B], como o *RandomUnderSampler* aplicada ao conjunto [C], contribuíram para abordar os desequilíbrios presentes no conjunto de dados, no entanto, os resultados indicaram que o conjunto [A] foi aquele que maximizou a capacidade dos algoritmos em classificar, com mais precisão e consistência, as transações como fraude ou não fraude.

5 Conclusões finais

Neste estudo, foi abordado a importância de sistemas de deteção de fraude no ramo do e-commerce e a aplicação de algoritmos de *machine learning* neste contexto, de maneira a desenvolver um modelo capaz de identificar transações suspeitas de fraude.

A fraude no comércio eletrónico é uma constante e complexa ameaça que prejudica a confiança dos consumidores, impõem custos financeiros às empresas e ameaça a integridade e segurança do comércio. Com base neste problema, o principal objetivo do presente trabalho prendeu-se com a criação de um sistema que através de técnicas de *machine learning*, incluindo abordagens de *deep-learning*, detetará transações fraudulentas. Nesse contexto, foram definidos os seguintes objetivos, referidos em 1.3:

- **O1:** Analisar e estudar o estado de arte atual de sistemas de deteção de fraudes no comércio online;
- **O2:** Que modelos, métodos e/ou algoritmos são necessários implementar num sistema de forma a este obter resultados de maior rigor e precisão possível;
- **O3:** Extrair e aprimorar um conjunto de dados com informação necessária para a deteção de transações fraudulentas;
- **O4:** Aplicar vários modelos e técnicas para a deteção de transações fraudulentas;
- **O5:** Avaliar diferentes conjuntos de dados;
- **O6:** Avaliar a performance e precisão dos diferentes modelos e técnicas testados;
- **O7:** Aferir qual o melhor modelo para a deteção de transações fraudulentas.

Relativamente ao **O1**, este objetivo foi alcançado através da aplicação da metodologia de pesquisa PRISMA, onde se formulou um conjunto de questões de pesquisa, que foram detalhadamente abordadas ao longo do capítulo 2 e respondidas no capítulo 2.5. Assim, foi feita uma análise abrangente da literatura existente, onde se explorou o ramo de *machine learning* aplicado aos sistemas de deteção de fraude, assim como diferentes modelos de

deteção de fraude. Fruto deste objetivo, cumpre-se também o **O2**, que visava identificar os modelos necessários de implementar para criar um sistema de deteção de fraudes capaz de alcançar resultados precisos e rigorosos. Os algoritmos RF, LR, SVM, KNN, DT, LSTM e CNN, foram os selecionados para esta vertente. É importante referir que, embora existam outras técnicas disponíveis, estas foram as mais utilizadas pelos autores nos diversos artigos estudados, pois demonstraram ser os mais adequados em sistemas de classificação e, então, utilizados para resolver problemas associados à deteção de fraudes. Os algoritmos RF e CNN foram ainda otimizados através da exploração de hiper-parâmetros.

O terceiro objetivo deste estudo, **O3**, que se concentrava na extração e aprimoramento de um conjunto de dados contendo informações essenciais para a deteção de transações fraudulentas, foi também alcançado com sucesso. Esta tarefa foi, no entanto, um desafio devido à confidencialidade dos dados. Qualquer empresa é responsável por proteger as informações financeiras dos seus clientes e, por isso, os seus conjuntos de dados são, geralmente, mantidas em sigilo incluindo, muitas das vezes, a total encriptação dos mesmos. O conjunto de dados selecionado foi fornecido pela *Vesta Corporation* e pela IEEE-CIS que cumprem rigorosamente as diretrizes e regulamentações do RGPD em relação aos dados fornecidos. Este conjunto de dados foi então devidamente pré-processado onde se fez uma limpeza e transformação dos dados. O final do pré-processamento resultou em três conjuntos de dados distintos, um com os dados de treino sem nenhuma modificação adicional, conjunto [A], outro com os dados de treino proveniente da aplicação da técnica de *oversampling*, *SMOTE,* conjunto [B], e por fim o conjunto [C], proveniente da aplicação da técnica de *undersampling*, *RandomUnderSampler*. O objetivo dos conjuntos de dados passou por avaliar os diferentes resultados obtidos e inferir assim sobre a eficácia de cada um. Toda a análise ao conjunto de dados, pré-processamento evolvente e os três subconjuntos obtidos, encontra-se detalhadamente descrito ao longo do capítulo 3.

Os objetivos **O4**, **O5** e **O6** deste estudo foram alcançados em conjunto, visto que estes objetivos dependiam intrinsecamente uns dos outros. Assim, para avaliar os diferentes conjuntos de dados, era necessário aplicar as diferentes técnicas de *machine learning* escolhidas e, então, avaliar a performance dos modelos testados. Aplicou-se então o RF, LR, SVM, KNN, DT, LSTM e CNN aos diferentes conjuntos de dados, de forma a avaliar o desempenho de cada modelo em cenários distintos. Por sua vez, a avaliação da performance dos modelos foi realizada com base nas métricas de exatidão, precisão, revocação e pontuação f-1. Estas métricas foram também escolhidas pelos autores dos artigos estudados no capítulo do estado de arte por serem as mais adequadas para avaliar o desempenho de sistemas de classificação. A pontuação f-1 foi a métrica escolhida para avaliar o modelo com melhor desempenho, por combinar a revocação e a precisão numa só medida. Num sistema de deteção de fraudes, é fundamental encontrar um equilíbrio entre identificar transações fraudulentas com precisão (para evitar falsos positivos) e não perder transações legítimas (para evitar falsos negativos. Os objetivos 4, 5 e 6 foram então alcançados com sucesso e seu procedimento encontra-se descrito ao longo do capítulo 0.

Por fim, o objetivo **O7**, onde se pretende determinar qual o modelo mais eficaz na deteção de compras fraudulentas, conclui-se que este é subjetivo. No presente estudo, os modelos que demonstraram melhor desempenho foram o RF e as CNN. Estes modelos foram escolhidos para uma exploração mais profunda dos seus hiper-parâmetros, com o objetivo de otimizar ainda mais a sua performance. No entanto, é importante destacar que a escolha do melhor modelo pode variar dependendo das características específicas dos conjuntos de dados, das métricas de avaliação e do objetivo principal do sistema, provocando assim resultados diferentes. Toda a análise aos resultados obtidos pelos autores nos artigos mencionados, e aos resultados obtidos no presente trabalho, encontram-se descritos detalhadamente no capítulo 4.4. A otimização dos hiper-parâmetros permitiu aumentar a pontuação f-1 em ambos os modelos aplicados.

Conclui-se também que, face aos resultados obtidos, a implementação de um modelo de deteção de fraudes, utilizando algoritmos de *machine learning*, potencializa a capacidade de identificar transações fraudulentas online, respondendo assim à questão de investigação definida no capítulo 1.3. Os resultados obtidos ao longo deste estudo demonstraram que a implementação de tal modelo tem o potencial de proteger as empresas e clientes contra atividades fraudulentas no comércio eletrónico. Além disso, considerando a escalabilidade destes modelos, é possível explorar a integração de um sistema de deteção de fraudes em forma de API. Com essa integração, as empresas poderiam facilmente avaliar a probabilidade de uma transação ser fraudulenta com base nas informações fornecidas, permitindo uma resposta proativa a possíveis ameaças e, assim, fortalecer ainda mais a segurança das transações online.

Assim, ao longo desta tese, foram contextualizadas as áreas em que o presente trabalho se insere, bem como os principais objetivos para solucionar o problema em questão e os principais motivos e contributos para a área da engenharia da inteligência artificial. Foram também descritas as metodologias de investigação e pesquisa utilizadas na elaboração do presente documento, assim com todo o processo de extração de dados envolvente. Foram apresentados vários algoritmos de *machine learning* que são aplicáveis aos sistemas de deteção de fraude como o LR, RF, SVM, DT, KNN, LSTM e CNN. Foi demonstrado que estes algoritmos podem ser treinados com conjunto de dados que contém transações financeiras de maneira a aprenderem a identificar padrões e comportamentos que indicam possíveis atividades suspeitas de fraude. Além disso, foi mencionado a importância da avaliação do desempenho do modelo, utilizando métricas como a precisão, exatidão, revocação e pontuação f-1 de maneira a garantir que o modelo produza resultados confiáveis. A exploração dos hiper-parâmetros dos algoritmos permite obter resultados mais precisos e eficazes.

5.1 Limitações

Uma das principais limitações enfrentadas foi a aquisição de um conjunto de dados relevante que representasse fielmente uma situação real de transações financeiras online. Muitos conjuntos de dados disponíveis continham informações fictícias, uma vez que se tratam

de transações online que envolvem informações de cartões de crédito, os quais não podem ser divulgadas devido a questões de privacidade e segurança.

Devido à natureza extensa e complexa do conjunto de dados testado no presente trabalho, o treino dos modelos de *machine learning* exigiu um tempo considerável. Um computador com capacidades técnicas superiores teria sido mais rápido e vantajoso, permitindo uma iteração mais dinâmica na otimização dos modelos. Ainda assim, e apesar destes contratempos, foi possível alcançar e cumprir todos os objetivos propostos.

5.2 Trabalho futuro

No que diz respeito a trabalho futuro, seria importante um trabalho aprimorado na otimização do tratamento de dados. O sucesso de qualquer modelo de *machine learning* depende em grande parte da qualidade dos dados sobre qual o modelo será treinado. Neste contexto, o tratamento de *outliers* e seleção das características mais relevantes, seriam parâmetros a ter em conta ou a serem melhorados.

Referências

Abhirami, K., Pani, A. K., Manohar, M., & Kumar, P. (2021). An Approach for Detecting Frauds in E-Commerce Transactions using Machine Learning Techniques. *Proceedings - 2nd International Conference on Smart Electronics and Communication, ICOSEC 2021*, 826–831. https://doi.org/10.1109/ICOSEC51865.2021.9591720

Addison Howard Bernadette Bouchon-Meunier, I. C. I. S. inversion J. L. L. M. Prof. H. A. (2019). *IEEE-CIS Fraud Detection*. Kaggle. https://kaggle.com/competitions/ieee-fraud-detection

Ahsan, M. M., Mahmud, M. A. P., Saha, P. K., Gupta, K. D., & Siddique, Z. (2021). Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance. *Technologies*, *9*(3). https://doi.org/10.3390/technologies9030052

Akker, J. van den, Gravemeijer, K., McKenney, S., & Nieveen, N. (2006). *Educational design research*.

Almohaimeed, A., & Gampa, S. (2019). Applying k-Nearest Neighbors to Increase the Utility of k-Anonymity. *Conference Proceedings - IEEE SOUTHEASTCON*, *2019-April*. https://doi.org/10.1109/SOUTHEASTCON42311.2019.9020525

Anowar, F., & Sadaoui, S. (2020). Detection of Auction Fraud in Commercial Sites. *Journal of Theoretical & Applied Electronic Commerce Research*, *15*(1), 81–98. https://doi.org/10.4067/S0718-18762020000100107

Belete, D., & D H, M. (2021). Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. *International Journal of Computers and Applications*, *44*, 1–12. https://doi.org/10.1080/1206212X.2021.1974663

Bhatt, D. H., & Meniya, A. (2022). A Review on Machine Learning Methods for Credit Card Fraud Classification. *Proceedings of the 2nd International Conference on Artificial Intelligence and Smart Energy, ICAIS 2022*, 312–318. https://doi.org/10.1109/ICAIS53314.2022.9743014

Bradley. (2022, June). *5 Types of E-commerce Fraud*. Merchant Fraud Jornal.

Brzeziński, D. (2010). Mining data streams with concept drift. *Cs Put Pozna*, 89.

Carta, S., Fenu, G., Reforgiato Recupero, D., & Saia, R. (2019). Fraud detection for E-commerce transactions by employing a prudential Multiple Consensus model. *Journal of Information Security and Applications*, *46*, 13-13–22. https://doi.org/10.1016/j.jisa.2019.02.007

Chen, J. I.-Z., & Lai, K. (2021). Deep Convolution Neural Network Model for credit-card fraud detection and alert. *Journal of Artificial Intelligence and Casule Network*, *3*(2).

Cheong, C., Cheong, F., & Filippou, J. (2013). *Using design science research to incorporate gamification into learning activities*.

Comissão Europeia. (2022). *A quem se aplica a lei da proteção de dados?*

Dhankhad, S., Mohammed, E., & Far, B. (2018). Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study. *IEEE International Conference on Information Reuse and Integration (IRI)*, 122–125.

Dowlagar, S., & Mamidi, R. (2022). DepressionOne@LT-EDI-ACL2022: Using Machine Learning with SMOTE and Random UnderSampling to Detect Signs of Depression on Social Media Text. *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, 301–305. https://doi.org/10.18653/v1/2022.ltedi-1.45

Duggineni, S. (2023). Impact of Controls on Data Integrity and Information Systems. *Science and Technology*, *13*(2), 29–35.

Esenogho, E., Mienye, I. D., Swart, T. G., Aruleba, K., & Obaido, G. (2022). A Neural Network Ensemble with Feature Engineering for Improved Credit Card Fraud Detection. *IEEE Access*, *10*, 16400–16407. https://doi.org/10.1109/ACCESS.2022.3148298

Ferreira, P., C. Le, D., & Zincir-Heywood, N. (2019). Exploring Feature Normalization and Temporal Information for Machine Learning Based Insider Threat Detection. *2019 15th International Conference on Network and Service Management (CNSM)*, 1–7. https://doi.org/10.23919/CNSM46954.2019.9012708

Gafsi, S. (2018). *Convolutional Neural Networks: Hyperparameters tuning and numerical results-A case study*.

Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A., & De, D. (2020). *Fundamental Concepts of Convolutional Neural Network* (pp. 519–567). https://doi.org/10.1007/978-3-030-32644-9_36

Gracia, S. V. J. B., Ponsam, J. G., Preetha, S., & Subhiksha, J. G. K. (2021). Payment fraud detection using machine learning techniques. *Proceedings of the 2021 4th International Conference on Computing and Communications Technologies, ICCCT 2021*, 623–626. https://doi.org/10.1109/ICCCT53315.2021.9711887

Himanshu Tripathi. (2019). *Different Type of Feature Engineering Encoding Techniques for Categorical Variable Encoding*. Analytics Vidhya.

Jain, V., Agrawal, M., & Kumar, A. (2020). Performance Analysis of Machine Learning Algorithms in Credit Cards Fraud Detection. *8th International Conference on Reliability, , Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* , 86–88.

Jiang, Y., & Stylos, N. (2021). Triggers of consumers' enhanced digital engagement and the role of digital technologies in transforming the retail ecosystem during COVID-19 pandemic. *Technol. Forecast*, *172*.

K., V., & K., S. (2022). Towards activation function search for long short-term model network: A differential evolution based approach. *Journal of King Saud University - Computer and Information Sciences*, *34*(6, Part A), 2637–2650. https://doi.org/https://doi.org/10.1016/j.jksuci.2020.04.015

Karimi, Z. (2021). *Confusion Matrix*.

Khatri, S., Arora, A., & Agrawal, A. (2020). Supervised machine learning algorithms for credit card fraud detection. *10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 680–683.

Kim, J., Han, Y., & Lee, J. (2016). Data imbalance problem solving for smote based oversampling: Study on fault detection prediction model in semiconductor manufacturing process. *Advanced Science and Technology Letters* , 79–84.

Kulshrestha, S., & Saini, M. L. (2020). Study for the Prediction of E-Commerce Business Market Growth using Machine Learning Algorithm. *2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering, ICRAIE 2020 - Proceeding*. https://doi.org/10.1109/ICRAIE51050.2020.9358275

Li, J. (2022). E-Commerce Fraud Detection Model by Computer Artificial Intelligence Data Mining. *Computational Intelligence and Neuroscience*, *2022*. https://doi.org/10.1155/2022/8783783

Liashchynskyi, P., & Liashchynskyi, P. (2019). *Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS*.

Lim, K. S., Lee, L. H., & Sim, Y.-W. (2021). A review of machine learning algorithms for fraud detection in credit card transaction. *International Journal of Computer Science and Network Security*, *21*(9).

Liu, Q., Li, P., Zhao, W., Cai, W., Yu, S., & Leung, V. C. (2018). A survey on security threats and defensive techniques of machine learning: A data driven view. *IEEE*, *6*, 12103–12117.

Liu, Y., Hua, R., Gao, W., & Chen, H. (2021). *Decomposition and Measurement of Economic Effects of E-commerce Based on Static Feder Model and Improved Dynamic Feder Model; Decomposition and Measurement of Economic Effects of E-commerce Based on Static Feder Model and Improved Dynamic Feder Model*. https://doi.org/10.1109/ECIT52743.2021.00054

Marley, R. (2019, October). *Account Takeover Frauds – Impact, Causes, and Prevention*. Account Takeover Frauds – Impact, Causes, and Prevention - Shufti Pro.

Mathew, J. C., Nithya, B., Vishwanatha, C. R., Shetty, P., Priya, H., & Kavya, G. (2022). An Analysis on Fraud Detection in Credit Card Transactions using Machine Learning Techniques. *Proceedings of the 2nd International Conference on Artificial Intelligence and Smart Energy, ICAIS 2022*, 265–272. https://doi.org/10.1109/ICAIS53314.2022.9742830

Molina, M., Castro, E., & Castro, E. (2007). Teaching experiments within design research. *The International Journal of Interdisciplinary Social Sciences*, *2*(4), 435–440.

Mustaqim, A. Z., Adi, S., Pristyanto, Y., & Astuti, Y. (2021). The Effect of Recursive Feature Elimination with Cross-Validation (RFECV) Feature Selection Algorithm toward Classifier Performance on Credit Card Fraud Detection. *ICAICST 2021 - 2021 International Conference on Artificial Intelligence and Computer Science Technology*, 270–275. https://doi.org/10.1109/ICAICST53116.2021.9497842

Negi, S., Das, S. K., & Bodh, R. (2022). Credit Card Fraud Detection using Deep and Machine Learning. *Proceedings - International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2022*, 455–461. https://doi.org/10.1109/ICAAIC53929.2022.9792941

Ng, H. W., & Winkler, S. (2014). A data-driven approach to cleaning large face datasets. *2014 IEEE International Conference on Image Processing, ICIP 2014*, 343–347. https://doi.org/10.1109/ICIP.2014.7025068

Nguyen, T. T., Nguyen, C. M., Nguyen, D. T., & Nahavandi, S. (2019). Deep learning for deepfakes creation and detection: a survey. *ArXiv* .

Nguyen, T., Tahir, H., Abdelrazek, M., & Ali Babar, M. (2020). *Deep Learning Methods for Credit Card Fraud Detection*.

Nithya, B., & Ilango, V. (2020a). Optimized Machine Learning based Classifications of Staging in Gynecological Cancers using Feature Subset through Fused Feature Selection Process. *International Journal of Advanced Computer Science and Applications* , *11*(7).

Nithya, B., & Ilango, V. (2020b). Machine Learning Aided Fused Feature Selection based Classification Framework for Diagnosing Cervical Cancer. *Proceedings of the Fourth International Conference on Computing Methodologies and Communication*.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *Systematic Reviews*, *10*(1), 1–11. https://doi.org/10.1186/S13643-021-01626-4/FIGURES/1

Pahadi, T. C., Verma, A., & Ranjan, R. (2022). Artificial Intelligence and its Influence on E-Commerce. *Proceedings - International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2022*, 1–6. https://doi.org/10.1109/ICAAIC53929.2022.9792783

Pawluszek-Filipiak, K., & Borkowski, A. (2020). On the Importance of Train–Test Split Ratio of Datasets in Automatic Landslide Detection by Supervised Classification. *Remote Sensing*, *12*(18). https://doi.org/10.3390/rs12183054

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–47.

Probst, P., Wright, M. N., & Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, *9*(3). https://doi.org/10.1002/widm.1301

Raja, S. K. S., Raman, C. J., & UshaKiruthika, S. (2021). A novel fraud detection scheme for credit card usage employing random forest algorithm combined with feedback mechanism. *Journal of Information Technology Management*, *13*(Special Issue: Big Data Analytics and Management in Internet of Thing), 21–35. https://doi.org/10.22059/jitm.2021.80615

Rodrigues, V. F., Policarpo, L. M., da Silveira, D. E., da Rosa Righi, R., da Costa, C. A., Barbosa, J. L. V., Antunes, R. S., Scorsatto, R., & Arcot, T. (2022). Fraud detection and prevention in e-commerce: A systematic literature review. *Electronic Commerce Research and Applications*, *56*. https://doi.org/10.1016/j.elerap.2022.101207

Rtayli, N., & Enneya, N. (2020). Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization. *Journal of Information Security and Applications*, *55*. https://doi.org/10.1016/j.jisa.2020.102596

Sadaghiyanfam, S., & Kuntalp, M. (2018). Comparing the Performances of PCA (Principle Component Analysis) and LDA (Linear Discriminant Analysis) Transformations on PAF (Paroxysmal Atrial Fibrillation) Patient Detection. *Proceedings of the 2018 3rd International Conference on Biomedical Imaging*.

Saputra, A., & Suharjito. (2019). Fraud detection using machine learning in e-commerce. *International Journal of Advanced Computer Science and Applications*, *10*(9), 332–339. https://doi.org/10.14569/ijacsa.2019.0100943

Sharma, S., Bellinger, C., Krawczyk, B., Zaiane, O., & Japkowicz, N. (2018). Synthetic oversampling with the majority class: A new perspective on handling extreme imbalance. *IEEE International Conference on Data Mining (ICDM)*.

Sheth, J. (2020). Impact of Covid-19 on consumer behavior: will the old habits return or die? *J. Bus.*, *117*, 280–283.

Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 105524. https://doi.org/10.1016/J.ASOC.2019.105524

Szász, L., Bálint, C., Csíki, O., Nagy, B. Z., Rácz, B. G., Csala, D., & Harris, L. C. (2022). The impact of COVID-19 on the evolution of online retail: The pandemic as a window of opportunity. *Journal of Retailing and Consumer Services*, *69*. https://doi.org/10.1016/j.jretconser.2022.103089

Taiwo Oladipupo Ayodele. (2010). Types of Machine Learning Algorithms. In Y. Zhang (Ed.), *New Advances in Machine Learning*. InTech.

Tasnim, A., Saiduzzaman, Md., Rahman, M. A., Akhter, J., & Rahaman, A. S. Md. M. (2022). Performance Evaluation of Multiple Classifiers for Predicting Fake News. *Journal of Computer and Communications*, *10*, 1–21.

Triguero, I., Garcia-Gil, D., Maillo, J., Garcia, S., & Herrera, F. (2019). Transforming big data into smart data: an insight on the use of the k-nearest neighbours algorithm to obtain quality data. *WIREs Data Mining and Knowledge Discovery*, *9*(2).

Varga, G. (2022). *5 Types of Ecommerce Fraud: How to Prevent & Detect it in 2023 | SEON*. https://seon.io/resources/ecommerce-fraud-detection-and-prevention/

Venna, S. R., Tavanaei, A., Gottumukkala, R. N., Raghavan, V. V., Maida, S., & Nichols, S. (2019). A novel data-driven model for real-time influenza forecasting. *IEEE Access*, *7*, 7691–7701.

Wang, W., Tong, M., & Yu, M. (2020). Blood glucose prediction with VMD and LSTM optimized by improved particle swarm optimization. *IEEE Access*, *8*, 217908–217916.

Warfield, D. (2010). IS/IT Research: A Research Methodologies Review. *Journal of Theoretical & Applied Information Technology*, *13*.

Weerts, H. J. P., Mueller, A. C., & Vanschoren, J. (2020). *Importance of Tuning Hyperparameters of Machine Learning Algorithms*.

Yan, S. (2020). The perception difference analysis of the influence of coastal residents of big data mining technology on marine tourism development. *Journal of Coastal Research*, *115*(1), 265–267.

Yen, L. (2021, September 15). *6 Types of Ecommerce Fraud and How To Prevent Them From Harming Your Customers*.

Zhou, H., Sun, G., Fu, S., Jiang, W., & Xue, J. (2019). A scalable approach for fraud detection in online e-commerce transactions with big data analytics. *Computers, Materials & Continua*, *60*(1), 179–192. https://doi.org/10.32604/cmc.2019.05214

Zou, X., Hu, Y., Tian, Z., & Shen, K. (2019). Logistic Regression Model Optimization and Case Analysis. *2019 IEEE 7th Internat Ional Conference on Computer Science and Network Technology (ICCSNT)*, 135–139.