

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

MARCOS ANDRÉ KÜNZEL PALHA

**Uma solução de processamento de
transações de crédito para predição de
fraudes**

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre em Ciência da
Computação

Orientador: Prof. Dra. Renata Galante

Porto Alegre
2023

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Palha, Marcos André Künzel

Uma solução de processamento de transações de crédito para predição de fraudes / Marcos André Künzel Palha. – Porto Alegre: PPGC da UFRGS, 2023.

121 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2023. Orientador: Renata Galante.

1. Detecção de fraude. 2. Tarefa de predição. 3. Técnicas emergentes. 4. Sistemas de aprendizagem. 5. Sistemas de decisão baseados em modelos de pontuação. I. Galante, Renata. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitora: Prof^a. Patricia Pranke

Pró-Reitor de Pós-Graduação: Prof. Julio Otávio Jardim Barcelos

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenadora do PPGC: Prof. Claudio Rosito Jung

Bibliotecário-chefe do Instituto de Informática: Alexsander Borges Ribeiro

*“A única coisa que não deve se curvar ao julgamento da maioria é a consciência
de uma pessoa.”*

— O SOL É PARA TODOS (TO KILL A MOCKING BIRD), HARPER LEE

AGRADECIMENTOS

Meus mais profundos agradecimentos á Universidade Federal do Rio Grande do Sul por me permitir realizar esse sonho de muitos anos com uma formação acadêmica diferenciada e de qualidade. Sonho esse que se materializou através da professora Renata Galante que acreditou em mim desde o momento da seleção e que me ajudou a buscar na teoria da nossa pesquisa a sustentação para os meus objetivos pessoais e profissionais.

Mas gostaria de reforçar o meu agradecimento a pessoa, orientadora e amiga Renata Galante por todo o apoio que ela me deu nesses tempos difíceis que passamos nos últimos 2 anos, o que garantiu que eu tivesse a resiliência necessária para enfrentar vários obstáculos. Tempos difíceis de incerteza e insegurança sobre o futuro e a minha própria capacidade, mas que resultaram nesse trabalho do qual me orgulho de ter desenvolvido. Obrigado pela paciência, dedicação e parceria.

Em função da pandemia, a maioria dos meus companheiros nessa jornada do mestrado foram virtuais, mas isso não impediu que nossos debates e colaborações me estimulassem a dar o melhor de mim na minha pesquisa. Agradeço principalmente aos colegas, Jaqueline Bitencourt, Braian Dias e Leonardo Vianna, pela força que me deram para terminar trabalhos que pareciam intermináveis. Meus grandes amigos Andre Martinotto, Demetrio Tiburi, Fernando Otten e Gustavo Lazzaroto por estarem estrategicamente lá, nos momentos em que mais precisei deles.

Agradeço a minha família por ter me dado os valores que me levaram a priorizar essa oportunidade. Principalmente ao meu Pai, Antonio Mario Palha, que infelizmente não estará aqui para ver o final dessa jornada, ele que nunca mediu esforços para nos mostrar a importância da educação nas nossas vidas, aos meus irmãos, João Carlos, Claudio e Fabio que sempre me aconselharam com suas ações e decisões. Minha Mãe Lucilda, minha grande companheira desde a infância e que representa a voz da minha consciência quando a decisão é realmente difícil. Finalmente agradeço a minha pequena família, Eliane Reuter minha esposa e companheira que vem enfrentando todos os desafios da vida junto comigo há 15 anos e o meu pequeno filho, Murilo, de 4 anos que me faz querer realizar os meus sonhos para ele saber que vale a pena sonhar.

Agradeço a banca avaliadora por aceitarem participar dessa jornada.

Muito obrigado!

RESUMO

Ao longo das últimas décadas, com o rápido crescimento dos negócios alavancados por canais da internet, houve um crescimento dramático do volume de fraudes, não apenas em números, mas também em formas e técnicas adotadas pelos fraudadores. Nesse mesmo período, evoluções tecnológicas significativas aconteceram, com uma crescente onda de técnicas relacionadas a inteligência artificial, *big data* e aprendizado de máquina. Estas abordagens inovadoras, para o contexto desse trabalho, são chamadas de “Técnicas Emergentes”, muitas delas oferecem subsídios importantes para o combate a fraudes em diversas áreas e formas. Inicialmente, neste trabalho, foi realizada uma revisão sistemática para mapear a contribuição efetiva das técnicas emergentes para combater fraudes, se comparada as abordagens de detecção de fraude clássicas. A partir da revisão sistemática, duas taxonomias foram propostas de forma a estruturar os trabalhos por tipos de fraude e categoria de técnica emergente utilizada na abordagem de detecção de fraude. Como um desdobramento da análise dos trabalhos levantados nessa pesquisa, este trabalho propõe uma solução para predição de fraudes em transações de crédito, baseada em aprendizado de máquina. Uma implementação dessa proposta é desenvolvida, descrita e experimentos são conduzidos utilizando três algoritmos distintos de aprendizado de máquina (RF, XGB e MLP). Estes algoritmos são alimentados com dados de uma base de transações de crédito real, sobre a qual quatro estudos de caso foram desenvolvidos, treinando e testando os três modelos com um mesmo conjunto de *features* e dados. Como parte dos resultados desses experimentos, os modelos são comparados entre si e com o sistema de regras de decisão atualmente utilizado pela empresa (*baseline*). Os resultados obtidos demonstram que os modelos de aprendizado de máquina são uma boa alternativa aos sistemas de detecção de fraude clássicos, melhorando métricas significativas como acurácia balanceada de 78,6% (MPBR) para 86,9% (MLP). Na comparação entre os modelos testados o MLP foi escolhido para prototipação devido as suas características simplificarem o retreino. Na análise dos resultados foi proposta também uma futura combinação dos modelos com o sistema de decisão, de forma a reduzir custo operacional tanto na atualização periódicas das regras do MPBR quanto no volume de transações que demandam uma revisão manual por parte da equipe de combate a fraudes.

Palavras-chave: Detecção de fraude. tarefa de predição. técnicas emergentes. sistemas de aprendizagem. sistemas de decisão baseados em modelos de pontuação.

A fraud prediction solution to process credit transactions

ABSTRACT

Over the last few decades, with the rapid growth of business leveraged by internet channels, there has been a dramatic increase in the volume of fraud, not only in numbers but also in the forms and techniques adopted by fraudsters. In this same period, significant technological developments have taken place, with a growing wave of techniques related to artificial intelligence, *big data* and machine learning. These innovative approaches, in the context of this work, are called “Emerging Techniques” many of which offer important subsidies for combating fraud in a variety of areas and forms. Initially, in this paper, a systematic review was conducted to map the effective contribution of emerging techniques to fight fraud compared to classical fraud detection approaches. From the systematic review, two taxonomies were proposed in order to structure the papers by types of fraud and category of emerging techniques used in the fraud detection approach. As a result of the analysis of the works surveyed in this research, this paper proposes a solution for predicting fraud in credit transactions, based on machine learning. An implementation of this proposal is developed, described, and experiments are conducted using three distinct machine learning algorithms (RF, XGB, and MLP). These algorithms are fed with data from a real credit transaction dataset, on which four experimental case studies were developed, training and testing the three models with the same set of *features* and data. As part of the results of these experiments, the models are compared with each other and with the decision rule system currently used by the company (*baseline*). The results obtained show that the machine learning models are a good alternative to classical fraud detection systems, improving significant metrics such as balanced accuracy from 78.6% (MPBR) to 86.9% (MLP). In the comparison between the tested models, MLP was chosen for prototyping due to its features to simplify retraining. In the analysis of the results, it was also proposed a future combination of the models with the decision system, in order to reduce the operational costs both in the periodic renewal of the MPBR rules and in the volume of transactions that demand a manual review by the fraud ops team.

Keywords: Fraud detection, prediction task, emerging techniques, learning systems, decision systems based on score model.

LISTA DE ABREVIATURAS E SIGLAS

AI	<i>Artificial Intelligence</i>
ML	<i>Machine Learning</i>
ANN	<i>Artificial Neural Networks</i>
DL	<i>Deep Learning</i>
HMM	<i>Hidden Markov Models</i>
MDP	<i>Markov Decision Process</i>
MLP	<i>Multi-Layer Perceptron</i>
PCA	<i>Principal Component Analysis</i>
RL	<i>Reinforcement Learning</i>
RMSE	<i>Root-mean-square Error</i>
RF	<i>Random Forest</i>
SVM	<i>Support Vector Machine</i>
XGB	<i>Extreme Gradient Boost</i>
ACTR	<i>Autorização de Compras em Tempo Real</i>
MDCU	<i>Motor de Decisões de Crédito Universal</i>
SGCV	<i>Sistema de Gerenciamento de Contas de Varejo</i>
MPBR	<i>Modelo de Pontuação Baseado em Regras</i>

LISTA DE FIGURAS

Figura 2.1	Representação Gráfica - Random Forest	20
Figura 2.2	Representação Gráfica - Extreme Gradient Boost	20
Figura 2.3	Representação Gráfica de um perceptron	21
Figura 2.4	Representação Gráfica - Perceptron Multi Camadas	22
Figura 2.5	Representação Gráfica do método de sub-amostragem utilizado no balanceamento de classes.....	23
Figura 3.1	Fluxo de revisão sistemática de literatura adotado nesse trabalho	27
Figura 3.2	Conjunto de palavras-chave utilizadas em motores de busca de sites de pesquisa para conduzir a SLR	29
Figura 3.3	Análise de referência cruzada usando gráfico de bolhas	30
Figura 3.4	Número de artigos por ano	31
Figura 3.5	Gráfico de barras empilhadas representando classes de fraude e os respectivos métodos encontrados na literatura para predição de fraude.....	32
Figura 3.6	Quantidade de artigos por método de aprendizado	33
Figura 3.7	Taxonomia de tipos de fraude.....	34
Figura 3.8	Fraud Prediction Approaches - Taxonomy	35
Figura 3.9	Diagrama de Sankey descrevendo interrelação entre taxonomias de tipos de fraude e técnicas de detecção.....	37
Figura 5.1	Fluxo de trabalho do sistema de regras de decisão.....	47
Figura 5.2	Diagrama de integração dos sistemas	49
Figura 6.1	Visão de alto nível	52
Figura 6.2	Etapa 1 - Obtenção de dados	53
Figura 6.3	Etapa 2 - Engenharia de features	54
Figura 6.4	Etapa 3 - Criação dos Modelos.....	56
Figura 6.5	Etapa 4 - Otimização dos Modelos.....	57
Figura 6.6	Visão Completa das Etapas da Solução Proposta para Predição de Fraudes em Transações de Crédito	61
Figura 7.1	Processo de Explosão de Atributos de XML em Tabelas de um banco relacional	65
Figura 7.2	Exemplo de processo de combinação (<i>join</i>) de registros para reduzir a cardinalidade das relações.....	66
Figura 7.3	Exemplo de processo de mescla (<i>merge</i>) de tabelas para reduzir quantidade de objetos de banco necessários para representar os dados.....	66
Figura 7.4	Análise de correlação de <i>features</i> - Ilustração do coeficiente de Pearson para pares de <i>features</i> do conjunto de dados - primeiro conjunto de <i>features</i>	72
Figura 7.5	Análise de correlação de <i>features</i> - Ilustração do coeficiente de Pearson para pares de <i>features</i> do conjunto de dados - segundo conjunto de <i>features</i>	73
Figura 7.6	Lista de <i>features</i> mais relevantes para decisões do modelo RF pela métrica: média decrescente de impureza (<i>mean decrease in impurity</i>).	75
Figura 7.7	Acurácia balanceada para diferentes conjuntos de dados de treinamento utilizando <i>features</i> da classe numérica.....	89

Figura 8.1 Metodologia dos Experimentos - Linha do tempo descrevendo todos os eventos com relevância para afetar o resultado dos experimentos destacando os momentos em que cada estudo de caso foi executado.....	94
Figura B.1 Taxonomia detalhada de técnicas emergentes.....	121

LISTA DE TABELAS

Tabela 4.1	Trabalhos relacionados - Abordagens que se destacam entre as propostas na sua classe de algoritmos.....	41
Tabela 4.2	Trabalhos relacionados - Abordagens que compartilham ao menos um algoritmo utilizado nesse trabalho	43
Tabela 7.1	Tabela com resumo dos conjuntos de dados avaliados como parte da etapa de obtenção de dados.....	64
Tabela 7.2	Descrição das <i>features</i> mais relevantes criadas pelo processo de extração. ..	74
Tabela 7.3	Descrição das <i>features</i> que tiveram maior relevância para decisões.	76
Tabela 7.4	Quantidade de <i>Features</i> Utilizadas no Treinamento de Diferentes Versões dos Modelos.	78
Tabela 7.5	Modelos iniciais gerados utilizando Random Forest (RF).	80
Tabela 7.6	Modelos iniciais gerados utilizando XGBoost (XGB).	81
Tabela 7.7	Modelos iniciais gerados utilizando Multilayer Perceptron (MLP).	81
Tabela 7.8	Tabela como hiperparâmetros ajustados como parte dos experimentos com o algoritmo Random Forest ordenados pela sua importância relativa.	83
Tabela 7.9	Tabela como hiperparâmetros ajustados como parte dos experimentos com o algoritmo XGBoost ordenados pela sua importância relativa.....	85
Tabela 7.10	Tabela como hiperparâmetros ajustados como parte dos experimentos com o algoritmo Multilayer Perceptron ordenados pela sua importância relativa..	86
Tabela 7.11	Eventos que influenciaram a evolução dos modelos de aprendizado de máquina ao longo do tempo.	87
Tabela 7.12	Listagem de modelos Random Forest (RF) gerados durante os experimentos.	90
Tabela 7.13	Listagem de modelos Extreme Gradient Boost (XGB) gerados durante os experimentos.	91
Tabela 7.14	Listagem de modelos Multilayer Perceptron (MLP) gerados durante os experimentos.	92
Tabela 8.1	Matriz de confusão gerada a partir dos resultados de um mês de transações processadas para pelo sistema de regras de decisão corporativo (janeiro de 2021).	96
Tabela 8.2	Tabela de métricas calculadas para o sistema de regras de decisão referente ao período de janeiro de 2021.	96
Tabela 8.3	Comparação dos Resultados do Baseline e Experimentos para os Modelos com 369 <i>features</i> [Agosto de 2022]	98
Tabela 8.4	Comparação dos Resultados do Baseline e Experimentos para os Modelos com 467 <i>features</i> [Outubro de 2022].....	100
Tabela 8.5	Comparação dos Resultados do Baseline e Experimentos para os Modelos com 429 <i>features</i> [Novembro de 2022]	102
Tabela 8.6	Comparação dos Resultados do Baseline e Experimentos	104
Tabela A.1	Trabalhos relacionados - Lista de artigos propondo uma abordagem centrada em Instance-Based learning.....	115
Tabela A.2	Trabalhos relacionados - Lista de artigos propondo uma abordagem centrada em Logical Base learning	116
Tabela A.3	Trabalhos relacionados - Lista de artigos propondo uma abordagem centrada em Neural Networks.....	117

Tabela A.4 Trabalhos relacionados - Lista de artigos propondo uma abordagem centrada em Support Vector Machine	118
Tabela A.5 Trabalhos relacionados - Lista de artigos propondo uma abordagem centrada em Statistical learning	119
Tabela A.6 Trabalhos relacionados - Lista de artigos propondo uma abordagem centrada em Hidden Markov Models e outros sistemas de decisão (<i>reasoning</i>) ..	120

LISTA DE ALGORITMOS

1	LÓGICA DE TREINAMENTO PARA MODELOS BASEADOS EM REDES NEURAIIS ...	60
2	ROTULAR TRANSAÇÕES FRAUDULENTAS.....	69

SUMÁRIO

1 INTRODUÇÃO	15
1.1 Motivação	15
1.2 Organização do texto	17
2 FUNDAMENTAÇÃO TEÓRICA	18
2.1 Técnicas Emergentes	18
2.1.1 Algoritmos baseados em árvores de decisão	19
2.1.1.1 Random Forest	19
2.1.1.2 Extreme Gradient Boost - XGB	20
2.1.2 Redes Neurais	21
2.1.2.1 Multilayer Perceptron	21
2.2 Balanceamento das Classes de Dados	22
2.3 Métricas Utilizadas para Avaliar os Modelos	23
3 REVISÃO SISTEMÁTICA DA LITERATURA: TIPOS DE FRAUDE E TÉCNICAS DE PREVENÇÃO	26
3.1 Metodologia	26
3.1.1 Questões de Pesquisa - QP.....	27
3.1.2 Definição do Protocolo de Execução	28
3.2 Análise quantitativa e qualitativa	30
3.3 Taxonomia	33
3.3.1 Taxonomia de tipos de fraude	33
3.3.2 Taxonomia de abordagens de detecção de fraude - técnicas emergentes	35
3.4 Considerações finais	38
4 TRABALHOS RELACIONADOS	40
4.1 Abordagens que se destacam entre as propostas na sua classe de algoritmos ..	40
4.2 Abordagens que compartilham ao menos um algoritmo utilizado nesse trabalho	41
4.3 Considerações finais	43
5 MODELO DE PONTUAÇÃO BASEADO EM REGRAS (MPBR)	45
5.1 FICO Blaze Advisor	45
5.2 Customizações do framework implementadas para a necessidade do negócio	45
5.3 Caso de uso do negócio - predição de fraudes em transações comerciais	50
5.4 Considerações Finais	51
6 UMA SOLUÇÃO DE PROCESSAMENTO DE TRANSAÇÕES DE CRÉDITO PARA PREDIÇÃO DE FRAUDES	52
6.1 Visão geral - (Metodologia)	52
6.2 Obtenção de Dados	53
6.3 Engenharia de <i>Features</i>	54
6.4 Criação dos Modelos	55
6.5 Otimização dos Modelos	57
6.5.1 Métricas Utilizadas para Avaliar os Resultados Obtidos	57
6.6 Fluxo Completo	61
7 IMPLEMENTAÇÃO DA SOLUÇÃO PROPOSTA	63
7.1 Obtenção de Dados	63
7.1.1 Transformação	64
7.1.2 Integração.....	65
7.1.3 Criação das Visões	66
7.1.4 Remoção de Dados Restritos	67
7.1.5 Associação de Rótulos	68

7.2 Engenharia de <i>Features</i>	70
7.2.1 Seleção de <i>Features</i>	70
7.2.2 Extração de <i>Features</i>	74
7.2.3 <i>Features</i> Mais Relevantes	75
7.3 Criação dos Modelos	77
7.3.1 Definição dos Algoritmos	79
7.3.2 Modelos Iniciais.....	79
7.4 Otimização dos Modelos	80
7.4.1 Abordagem de Otimização de Hiperparâmetros.....	82
7.4.2 Hiperparâmetros <i>Random Forest</i> - RF	82
7.4.3 Hiperparâmetros <i>Extreme Gradient Boosting</i> - XGB	84
7.4.4 Hiperparâmetros <i>Multilayer Perceptron</i> - MLP	84
7.4.5 Linha do Tempo dos Modelos Gerados	87
8 RESULTADOS DOS EXPERIMENTOS	93
8.1 Configuração dos Experimentos	93
8.2 Execução dos Experimentos	95
8.2.1 Execução do Baseline	95
8.2.2 Execução dos Experimentos com os Modelos de Aprendizado de Máquina	96
8.2.3 Estudo de caso 1 - Modelos com 369 <i>features</i>	97
8.2.4 Estudo de caso 2 - Modelos com 467 <i>features</i>	99
8.2.5 Estudo de caso 3 - Modelos com 429 <i>features</i>	101
8.2.6 Estudo de caso com a versão final de cada modelo	103
8.3 Considerações sobre os resultados obtidos	103
9 CONCLUSÃO	106
9.1 Direções futuras	108
REFERÊNCIAS	109
APÊNDICE A — LISTA COMPLETA DE ARTIGOS DE PROPOSTA DE ABORDAGEM COBERTOS NA SLR	115
APÊNDICE B — TAXONOMIA DETALHADA	121

1 INTRODUÇÃO

As transações fraudulentas custam às empresas e aos consumidores, perdas financeiras significativas todos os anos (TAHA; MALEBARY, 2020). Como resultado, a academia e a indústria desenvolveram vários sistemas de detecção de fraude. No entanto, criar uma solução universal para todos os sistemas de detecção de fraudes em transação eletrônicas é inviável. Mesmo considerando um contexto empresarial específico, a detecção de fraudes em tempo real ainda é uma tarefa difícil. Além disso, como os fraudadores estão constantemente criando novas estratégias, os sistemas de detecção de fraude precisam evoluir para se adaptar e manter sua eficácia.

Os métodos clássicos de automação da detecção de fraudes se baseiam em sistemas de regras de pontuação. Estas regras simbólicas e valores limiares estáticos identificam parte das transações como suspeitas (ZHU et al., 2021). A partir daí, as transações suspeitas podem ser auditadas manualmente. Se o padrão de fraude é novo e não coberto pelas regras do sistema, ela pode não ser percebida, levando a perdas monetárias. A principal vantagem de um sistema de pontuação baseado em regras é que as regras e limites são definidos por um grupo de especialistas familiarizados com as especificidades do problema. Entretanto, a fim de se adaptar a estratégias cada vez mais sofisticadas dos fraudadores e identificar transações ilícitas tão rapidamente quanto possível, o sistema baseado em regras requer atualizações manuais periódicas, pontuais (*ad hoc*). Além disso, o ciclo de atualização deste modelo de pontuação requer um trabalho maciço de especialistas e esforço de codificação.

1.1 Motivação

Outra abordagem para identificar automaticamente transações fraudulentas consiste da utilização de sistemas de aprendizagem de máquinas (ML) onde os modelos podem ser treinados com base em dados históricos para detectar fraudes. Uma vez que aprende como classificar as transações como legítimas ou fraudulentas, o modelo ML pode decidir sobre novos dados não-classificados. Como as estratégias de fraude evoluem, é imperativo atualizar o treinamento ML para se adaptar. Este processo de evolução é chamado de retreino, pode ser automatizado e ocorrer em uma escala de tempo mais rápida, exigindo assim menos esforço do que os sistemas baseados em regras.

O objetivo desse trabalho é propor uma solução baseada em aprendizado de má-

quina para predição de fraude em um contexto de financiamento de transações de compra. A abordagem proposta tem como premissa ser capaz de se adaptar a novos padrões de fraude sem a necessidade de intervenção humana, com níveis de precisão, cobertura e eficiência econômica tão bons ou melhores do que os apresentados por um sistema de regras de decisão.

Durante o processo de levantamento bibliográfico, diversas técnicas, abordagens e algoritmos relacionados às disciplinas de aprendizado de máquina e ciência de dados foram estudados e selecionados para fazer parte do processo de geração e otimização da solução de predição de fraudes em transações de crédito proposta.

A solução de predição de fraudes em transações de crédito, proposta nesse trabalho, compreende as seguintes etapas: (1) obtenção dos dados, (2) engenharia de *features*, (3) criação dos modelos e (4) otimização dos modelos gerados. Ao final desse processo obtemos um modelo de aprendizado de máquina treinado e testado para a tarefa de predição de fraudes em transações de crédito.

As contribuições desse trabalho são:

- Uma revisão sistemática da literatura planejada para identificar oportunidades de pesquisa na aplicação de técnicas emergentes de predição de fraude em diversos contextos de fraude;
- Uma taxonomia de técnicas emergentes, gerada a partir da revisão sistemática, dividida em 2 grandes grupos de abordagens: (1) sistemas de aprendizagem e (2) sistemas de decisão (*reasoning*);
- Uma taxonomia de tipos de fraude, gerada a partir da revisão sistemática, dividida em 4 categorias principais, incluindo uma categoria específica para fraudes do tipo financeiro;
- Uma solução para predição de fraudes em transações de crédito;
- Três modelos classificadores, baseados em diferentes algoritmos, otimizados e experimentados na tarefa de predição de fraudes;
- O resultado de experimentos utilizando um conjunto de transações de uma empresa especializada em concessão de crédito para compras em um formato análogo a uma transação de cartão de crédito;

1.2 Organização do texto

O restante do documento está organizado como segue. O Capítulo 2 apresenta os principais conceitos que embasam este trabalho. O Capítulo 3 descreve o levantamento bibliográfico desenvolvido para amparar essa pesquisa. O Capítulo 4 apresenta uma revisão do estado da arte. O Capítulo 5 descreve a aplicação sendo utilizada como medida de comparação para a solução proposta. O Capítulo 6 especifica os métodos utilizados para predição de fraude em transações de crédito. Por fim, os Capítulos 7 e 8 apresentam, respectivamente, os detalhes da implementação da proposta para um estudo de caso e os resultados dos experimentos realizados, enquanto o Capítulo 9 apresenta as conclusões e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda conceitos e algoritmos que fundamentam a proposta de solução para predição de fraudes apresentada e implementada como parte dessa dissertação. Inicialmente é apresentado o conceito de “Técnicas Emergentes” utilizadas na prevenção de fraudes. Em seguida, é apresentado um resumo dos principais algoritmos e abordagens adotados na implementação de predição de fraudes proposta como parte deste trabalho, bem como as métricas de avaliação utilizadas para mensurar os resultados dos modelos.

2.1 Técnicas Emergentes

Ao longo das últimas duas décadas, com evolução do comércio eletrônico, portais de venda e transações eletrônicas, a fraude cibernética teve um crescimento sem precedentes. Em paralelo, houve uma evolução incrível de tecnologias como inteligência artificial, big data e aprendizado de máquina, tecnologias essas que para o contexto deste trabalho foram chamadas de “Técnicas Emergentes”.

Para o contexto da classificação de abordagens proposta nesse trabalho, podemos dizer que todas as Técnicas Emergentes pertencem ao grupo de abordagens de Inteligência Artificial (IA). Sendo que a IA abrange uma enorme variedade de subcampos com características em comum, como a capacidade, raciocínio, aprendizagem, reconhecimento de padrões, e inferência. Segundo (GERON, 2019) *Machine learning* (ML) ou Aprendizado de máquina é um campo da inteligência artificial que se concentra no desenvolvimento de algoritmos e modelos que permitem que um sistema de computador aprenda a partir de dados, sem ser explicitamente programado para uma tarefa específica. O objetivo é criar modelos que possam ser treinados com dados e, em seguida, usados para fazer previsões ou tomar decisões com base em novos dados.

Neste esse subconjunto da IA, aqui definido como Técnicas Emergentes, que se caracterizam pelo seu potencial para endereçar o problema de predição de fraudes, existem duas grandes categorias de abordagens: (1) sistemas de decisão (*reasoning*) e (2) sistemas de aprendizado, também conhecidos como sistemas de aprendizado de máquina (ML). Em (BOTTOU, 2014) a diferença entre sistemas de decisão e aprendizado é definida como: “Sistemas de decisão podem ser treinados e, aprender a partir de um conjunto de dados disponível, assim como sistemas de aprendizado, porém, sistemas de decisão permitem a solução de novos problemas, não identificados originalmente, usando técni-

cas de decisão por dedução e indução”. A habilidade de compor um ‘motor de inferências’ que descreve como manipular e combinar símbolos de uma base de conhecimento para um propósito diferente é o que diferencia sistemas de decisão de sistemas de aprendizado de máquina.

Contudo, para o contexto desse trabalho, como parte da hierarquia de abordagens identificadas como sistemas de decisão (*reasoning*) estão técnicas as que uma lógica de decisão é proposta de forma programática, como sistemas especialistas, lógica Fuzzy e sistemas de recomendação. Entre os sistemas de aprendizado a hierarquia se divide em 5 subcategorias, são elas: Instance based, Redes Neurais, Baseados em Árvores de Decisão, Support Vector Machine e Aprendizado Estatístico.

Na elaboração e implementação dos experimentos deste trabalho, foram utilizados algoritmos de aprendizado de máquina pertencentes a categoria de aprendizado baseado em lógica e redes neurais. Uma visão geral sobre estas categorias de algoritmo de aprendizado de máquina e, os respectivos algoritmos, é apresentada no restante desse capítulo.

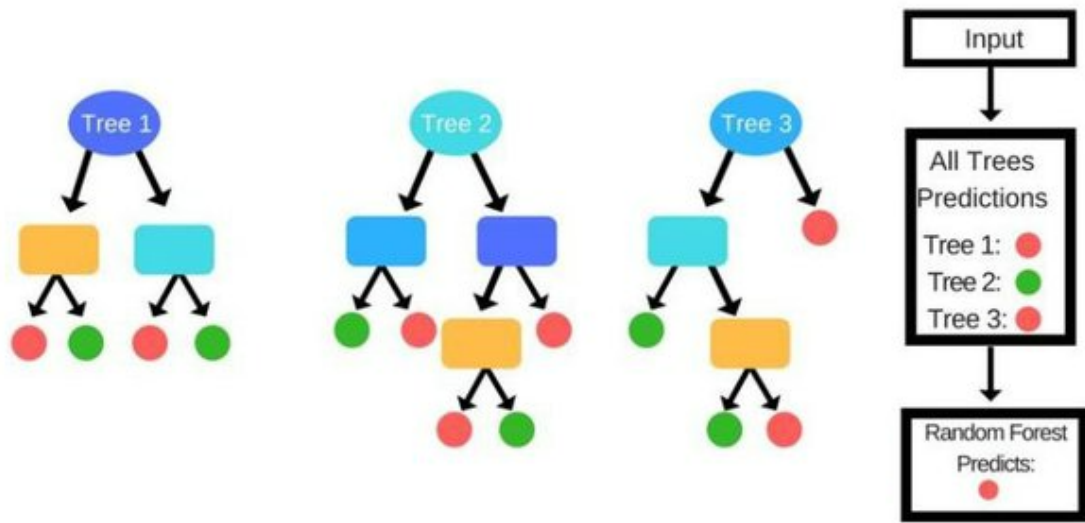
2.1.1 Algoritmos baseados em árvores de decisão

Esta seção trata de algoritmos de aprendizado de máquina baseados em árvores de decisão, categoria cujo principal representante é o método *Random Forest* (RF). Estas abordagens se caracterizam por segmentar o espaço de predição em regiões mais homogêneas, segundo as possíveis respostas. Pertencem à classe dos algoritmos de aprendizado supervisionados e podem ser usados tanto para classificação quanto para regressão, permitindo rastrear o caminho pelo qual uma predição foi realizada (caixa branca). Foram utilizados nesse trabalho os algoritmos baseados em árvore conhecidos como Random Forest (RF) e Extreme Gradient Boost (XGB).

2.1.1.1 Random Forest

O algoritmo de *Random Forest* (BREIMAN, 2001) consiste em um grande número de árvores de decisão individuais que funcionam coletivamente. Cada árvore individual na floresta aleatória produz uma predição de classe e a classe com mais votos torna-se a previsão do modelo, como representado na Figura 2.1.

Figura 2.1: Representação Gráfica - Random Forest



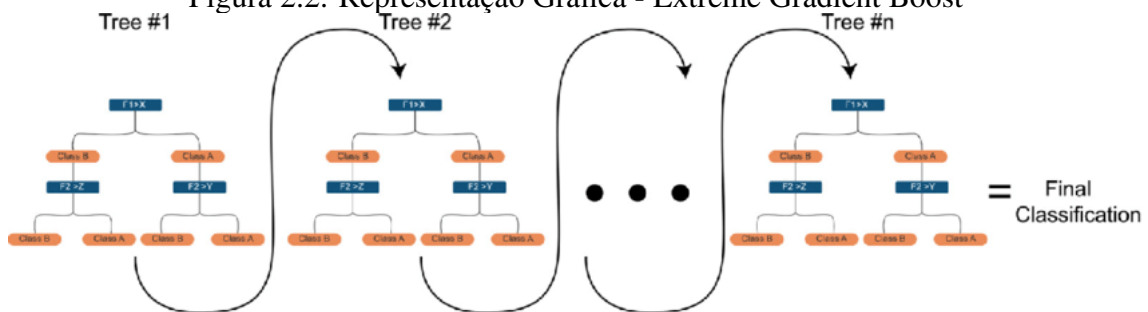
Fonte: Elaborado pelo autor

2.1.1.2 Extreme Gradient Boost - XGB

O algoritmo Extreme Gradient Boost, originalmente publicado em (CHEN; GUESTRIN, 2016), também se baseia em uma floresta de árvores de decisão, similar ao RF, a diferença está na ideia de melhorar o resultado de um único modelo fraco combinando as suas previsões para chegar a um modelo forte. A proposta é treinar iterativamente um conjunto de árvores de decisão rasas, com cada iteração usando os resíduos de erro do modelo anterior para ajustar o modelo seguinte. A previsão final é uma soma ponderada de todas as previsões das árvores.

Enquanto o algoritmo de “*bagging*” do Random Forest se propõe a minimizar a variância e “*overfitting*”, o método de “*boosting*” do XGB visa minimizar o viés e o “*underfitting*”. A Figura 2.2 apresenta uma representação gráfica do método.

Figura 2.2: Representação Gráfica - Extreme Gradient Boost



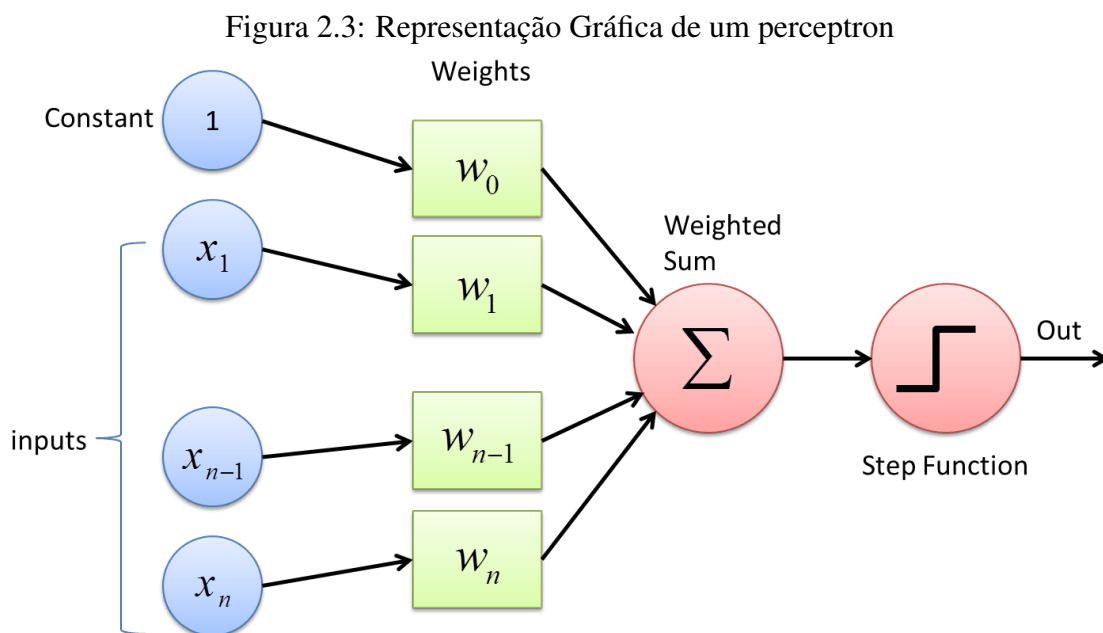
Fonte: Elaborado pelo autor

2.1.2 Redes Neurais

Esta categoria representa a classe de algoritmos popularmente conhecida como redes neurais e baseia-se em conceitos de uma área da IA chamada de *Deep Learning*. O funcionamento das redes neurais se inspira nos neurônios do cérebro humano, sendo constituída por nós que se interconectam para processar entradas e gerar saídas. Sob esta categoria de algoritmos temos as Multilayer Perceptrons (MLPs), Radial Basis Function (RBF), além de diversas variações do Redes Neurais Artificiais como as *Convolutional Neural Networks*. Esta classe de algoritmos é muito popular e ampla, contemplando tanto algoritmos adequados para treinamento supervisionados (ex: MLP) quanto algoritmos adequados para treinamento não supervisionado (ex: Self-Organizing Map - SOM). Neste trabalho o tipo de rede neural utilizado foi o Multilayer Perceptron (MLP).

2.1.2.1 Multilayer Perceptron

O perceptron é um motor computacional simples, composto de um único neurônio e também chamado de “máquina linear”. Os perceptrons são constituídos de 4 partes principais incluindo, valores de entrada, pesos e viés, somatório líquido e uma função de ativação, como representado na Figura 2.3.



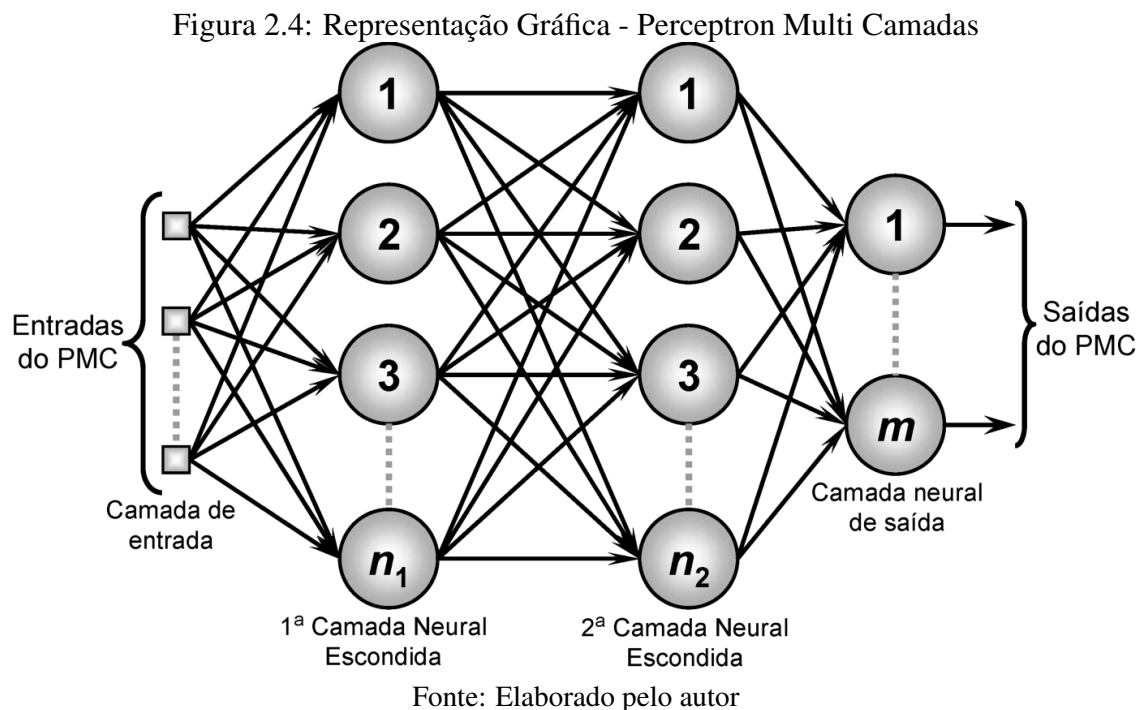
Fonte: Elaborado pelo autor

No artigo (MURTAGH, 1991) foi proposta a utilização de um perceptron de várias camadas (MLP), sendo uma classe de rede neural composta por pelo menos 3 nós. O

perceptron multicamadas se caracteriza por todos os seus nós serem neurônios de ativação não linear, com exceção do nó de entrada. Os nós do MLP estão dispostos em camadas divididas como:

- A camada de entrada
- A camada de saída
- Camadas ocultas: camadas entre a entrada e a saída

A Figura 2.4 representa uma rede neural MLP, demonstrando a arquitetura da rede e como as entradas são processadas por meio das camadas ocultas até a camada de saída para produzir a saída final.



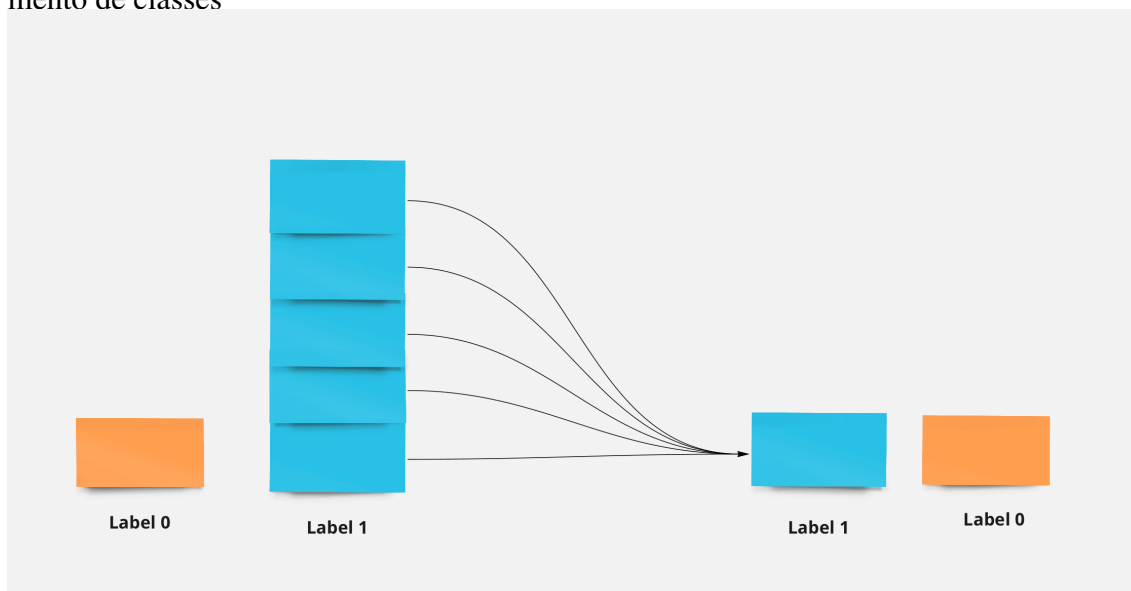
2.2 Balanceamento das Classes de Dados

Uma característica do problema de predição de fraudes é possuir um domínio de classes altamente desbalanceado. Isto acontece porque o volume de registros fraudulentos é muito inferior ao número de registros legítimos. Esta característica se traduz em um impacto para o processo de aprendizado que se torna muito menos eficaz em função dessa disparidade entre as classes.

Para efetuar o balanceamento das classes de dados neste trabalho foi utilizada uma técnica de sub-amostragem (*undersampling*) de transações legítimas, equiparando a

quantidade de transações fraudulentas e legítimas. A Figura 2.5 representa o processo utilizado para o balanceamento das classes de dados do problema adotado para melhorar as métricas da tarefa de predição de fraudes. As barras do lado esquerdo da imagem representam um conjunto de dados desbalanceado, onde a classe com *label* 1 (em azul) tem muito mais exemplos do que a classe *label* 0 (em laranja). A primeira etapa do algoritmo de *undersampling* é escolher aleatoriamente alguns exemplos da classe com *label* 1 para remover, de forma que a classe com *label* 0 tenha o mesmo número de exemplos que a classe com *label* 1. No lado direito temos o resultado do balanceamento após a remoção de 4 exemplos da classe com *label* 1.

Figura 2.5: Representação Gráfica do método de sub-amostragem utilizado no balanceamento de classes



Fonte: Elaborado pelo autor

2.3 Métricas Utilizadas para Avaliar os Modelos

A listagem abaixo descreve as métricas utilizadas para avaliar os resultados dos experimentos ordenadas pelo seu grau de relevância:

- (1) Verdadeiro Negativo (VN): quantidade de ocorrências em que ambos o rótulo e a predição são Negativas N .
 - Quanto **Maior** o valor **Melhor**.
- (2) Verdadeiro Positivo (VP): quantidade de ocorrências em que ambos o rótulo e a predição são Positivas P .

- Quanto **Maior** o valor **Melhor**.
- (3) Falso Negativo (FN): quantidade de ocorrências em que a predição é Negativa N e o rótulo é Positivo P .
 - Quanto **Menor** o valor **Melhor**.
- (4) Falso Positivo (FP): quantidade de ocorrências em que a predição é Positiva P e o rótulo é Negativo N .
 - Quanto **Menor** o valor **Melhor**.
- (5) Acurácia Balanceada
 - $\frac{TP+TN}{TP+FN+TN+FP}$
 - cálculo de todos os acertos (VP e VN) divididos por todos os acertos mais os erros (FP e FN)
 - Quanto **Maior** o valor **Melhor**.
- (6) Correlação de Matthews
 - $$\frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$
 - é uma medida equilibrada de precisão, que pode ser usada mesmo que uma classe tenha muito mais amostras do que outra.
 - Quanto **Maior** o valor **Melhor**.
- (7) Precisão - PPV (*Positive Predictive Value*)
 - $\frac{TP}{TP+FP}$
 - avalia a quantidade de verdadeiros positivos sobre a soma de todos os valores positivos
 - Quanto **Maior** o valor **Melhor**.
- (8) Sensitividade - TPR (*True Positive Rate*) - (*Recall*)
 - $\frac{TP}{P} = \frac{TP}{TP+FN}$
 - avalia a capacidade do método de detectar com sucesso resultados classificados como positivos
 - Quanto **Maior** o valor **Melhor**.

- (9) Especificidade - TNR (*True Negative Rate*)
 - $\frac{TN}{N} = \frac{TN}{TN+FP}$
 - avalia a capacidade do método de detectar resultados negativos
 - Quanto **Maior** o valor **Melhor**.
- (10) Acurácia
 - $\frac{(TP+TN)}{(P+N)}$
 - percentual de acertos
 - Quanto **Maior** o valor **Melhor**.
- (11) Pontuação F1
 - $\frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+FN}$
 - média harmônica calculada com base na precisão e na revocação
 - Quanto **Maior** o valor **Melhor**.

3 REVISÃO SISTEMÁTICA DA LITERATURA: TIPOS DE FRAUDE E TÉCNICAS DE PREVENÇÃO

Este capítulo apresenta a revisão sistemática da literatura realizada para identificar e apresentar um panorama sobre diferentes tipos de fraude e as técnicas desenvolvidas para sua prevenção em diferentes contextos, como uma forma de identificar nichos de pesquisa onde não exista uma profusão de estudos, visando responder as seguintes perguntas:

- Como as técnicas emergentes (*como: aprendizado de máquina, sistemas de decisão*) contribuíram para melhorar as soluções clássicas de predição de fraudes ao longo dos últimos 20 anos?
- Quais os tipos de fraude onde técnicas de predição são mais aplicadas atualmente?
- Quais as técnicas emergentes de predição de fraudes mais adotadas na última década? E quais fatores contribuem para essa popularidade?

O conteúdo aqui apresentado foi extraído da *survey*¹ desenvolvida em preparação para esse trabalho e está estruturado da seguinte forma: na Seção 3.1, detalhamos a metodologia empregada para a realização da revisão sistemática da literatura. Na Seção 3.2, é conduzida uma análise quantitativa e qualitativa dos dados levantados. Na Seção 3.3 uma taxonomia é proposta para estruturação dos resultados. Por fim, na Seção 3.4 são discutidos os resultados obtidos.

3.1 Metodologia

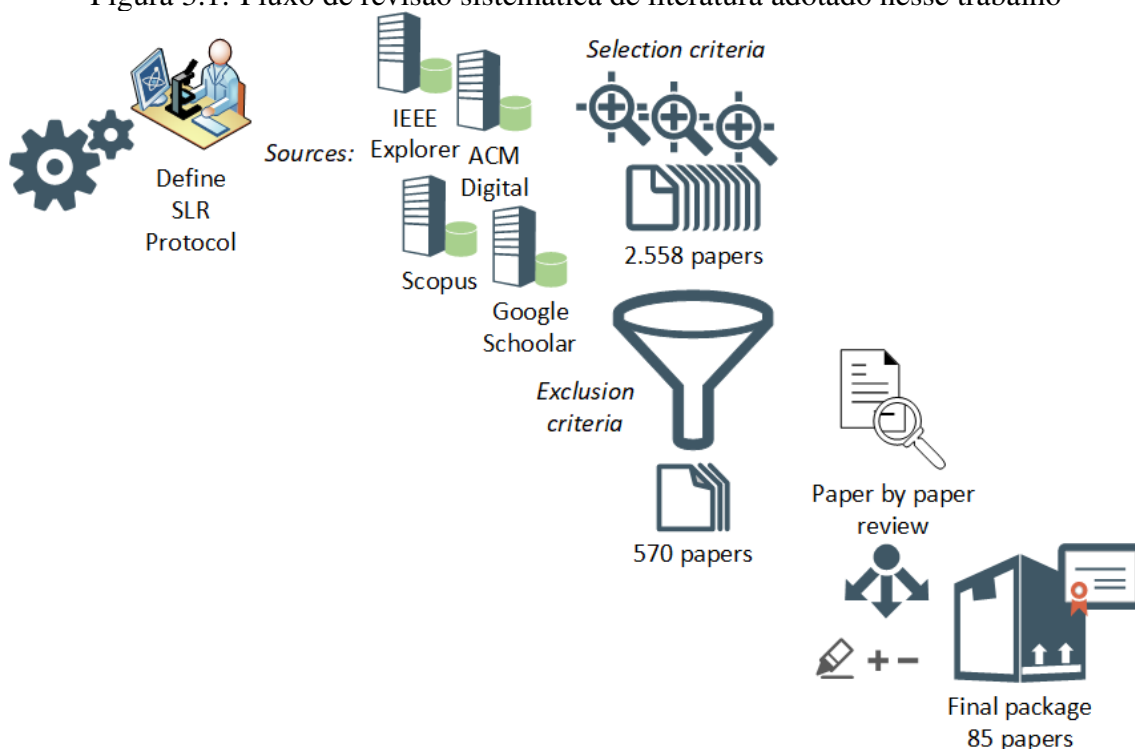
A revisão sistemática de literatura (do inglês SLR) aqui apresentada foi conduzida com o intuito de identificar trabalhos relevantes na área de detecção de fraudes e como as técnicas emergentes apresentadas no contexto da pesquisa agregaram valor para o contexto das soluções disponíveis para este domínio de problema. A metodologia usada nessa SLR é baseada no método Cochrane tendo utilizado um protocolo PICO (População, Intervenção, Controle e Objetivo) para formular questões de pesquisa bem definidas, que ajudam a identificar e avaliar os estudos relevantes para este assunto.

O fluxo da revisão sistemática conduzida como parte deste trabalho incluindo desde a definição do seu protocolo até a passo a passo da sua execução está represen-

¹Este artigo foi submetido para publicação no *journal* “Computers & Security”.

tado na Figura 3.1.

Figura 3.1: Fluxo de revisão sistemática de literatura adotado nesse trabalho



Fonte: Elaborado pelo autor

3.1.1 Questões de Pesquisa - QP

Considerando que o principal objetivo desta revisão sistemática é apontar as técnicas emergentes existentes para a tarefa de predição de fraudes de diferentes tipos em diferentes contextos, este foco é o que permeia as QPs primárias e secundárias da SLR, auxiliando assim a derivar outros parâmetros do protocolo.

- QP1 (primária): Como as técnicas emergentes (por exemplo: aprendizado de máquina, sistemas de decisão, etc) contribuíram para aperfeiçoar métricas de predição de fraudes ao longo dos últimos 10 anos?

Para obter dados e sumarizar diferentes aspectos dos tópicos estudados, e identificar lacunas nas pesquisas existentes, QPs secundárias foram definidas. As perguntas secundárias tiveram como objetivo detalhar mais granularmente os diferentes tipos de fraude identificados como objetivo para aplicar as técnicas emergentes (QP2). Abordando também técnicas que ganharam popularidade na automação da tarefa de prevenção

de fraudes (QP3). As QPs secundárias foram especialmente utilizadas para definir alguns dos critérios de inclusão e exclusão do protocolo da SLR.

- QP2: Quais os principais tipos de fraude abordados em pesquisas relacionadas a técnicas emergentes atualmente?
- QP3: Quais as principais técnicas emergentes adotadas de forma efetiva para melhorar métricas de prevenção de fraudes recentemente?

3.1.2 Definição do Protocolo de Execução

Para refinar a SLR proposta para esse pesquisa foram definidas as questões PICO a serem utilizadas para definir parâmetros quantitativos claros para a busca. A sigla PICO define os seguintes parâmetros para o contexto da SLR: **População** - contexto da busca em termos de estudos alvo para serem revisados, **Intervenção** - o que será observado nos resultados da SLR planejada, **Controle** - linha de base ou conjunto de dados iniciais que o pesquisador possui como entrada para o processo e **Objetivo** - tipos de resultados esperados ao final da SLR, combinando métricas adotadas para avaliar os resultados encontrados.

No contexto dessa pesquisa os parâmetros PICO foram definidos da seguinte forma:

- **População:** “*Estudos recentes (últimos 10 anos) que exploraram a adoção de técnicas emergentes para detecção de fraudes em algum contexto*”
- **Intervenção:** “*Entender quanto as técnicas emergentes efetivamente contribuíram para melhorar os resultados da detecção de fraudes*”
- **Controle:** “*Estudos relacionados a detecção de fraudes usando métodos tradicionais tais como sistemas de regras de decisão*”
- **Objetivo:** “*Confirmar e mensurar o impacto positivo das técnicas emergentes na prevenção de perdas financeiras causadas por fraudes*”

O processo de seleção de estudos é o mais estágio mais importante na execução da SLR, ela foi quebrada em algumas fases. Inicialmente 5 rodadas de sessões de busca foram conduzidas em 4 motores de busca. As chaves de busca foram baseadas nas seguintes combinações de palavras: “*Credit fraud analysis*”, “*fraud prediction*”, “*fraud outliers*”, “*fraud detection*” e “*fraud prevention*”. Estas combinações de palavras chaves foram usa-

Figura 3.2: Conjunto de palavras-chave utilizadas em motores de busca de sites de pesquisa para conduzir a SLR

Palavras-chave utilizadas nas buscas

Sessão de busca 1: “Credit fraud analysis”

Sessão de busca 2: “Fraud outliers”

Sessão de busca 3: “Fraud prediction”

Sessão de busca 4: “Fraud detection”

Sessão de busca 5: “Fraud prevention”

Fonte: Elaborado pelo autor

das nos seguintes portais de artigos acadêmicos: *ACM Digital*, *IEEE Explorer*, *Google Scholar*, e *Scopus*, sem nenhum filtro adicional. Este exercício resultou em uma lista com 2.558 trabalhos. A lista completa de artigos acadêmicos foi então carregada na ferramenta *StArt* (*State of Art tool*) onde rodadas adicionais de refinamento da lista foram conduzidas através da introdução de critérios de exclusão (CEs):

- CE1: remover resultados duplicados retornados por diferentes motores de busca,
- CE2: remover trabalhos publicados antes de 2010,
- CE3: pontuação de correspondência com a combinação de palavras chave inferior a 10 (*pontuação gerada pela ferramenta StArt de acordo com o número de ocorrências de cada palavra no título, resumo e palavras-chave do artigo*) and
- CE4: ausência de referências a palavra-chave “fraude”.

Ao aplicarmos esses critérios de exclusão, o número total de artigos foi reduzido para 570 registros. O gráfico de bolhas ilustrado na Figura 3.3 representa a distribuição dos estudos pelos tipos de fraude e abordagens de detecção de fraude. Nesse gráfico as colunas representam os diferentes tipos de fraude e as cores das bolhas representam as abordagens propostas pelos autores, abordagens essas descritas na legenda do lado direito do gráfico. O tamanho da bolha representa a quantidade de trabalhos para aquela combinação de abordagem e tipo de fraude. Nessa etapa do refinamento dos resultados da SLR ainda estavam inclusos trabalhos como propostas de abordagens não automatizadas (por exemplo: *not automated* e *fraud modeling*) e tipos de fraude sem conotação financeira (por exemplo: *fake reviews*).

Uma última rodada de filtros consistiu de remover tipos de fraude que não tivessem uma correlação com perda financeira e a remoção de abordagens com baixa representatividade, menos de 10 trabalhos no conjunto reduziu o conjunto de trabalhos no para 85

Figura 3.3: Análise de referência cruzada usando gráfico de bolhas



Fonte: Elaborado pelo autor

artigos acadêmicos.

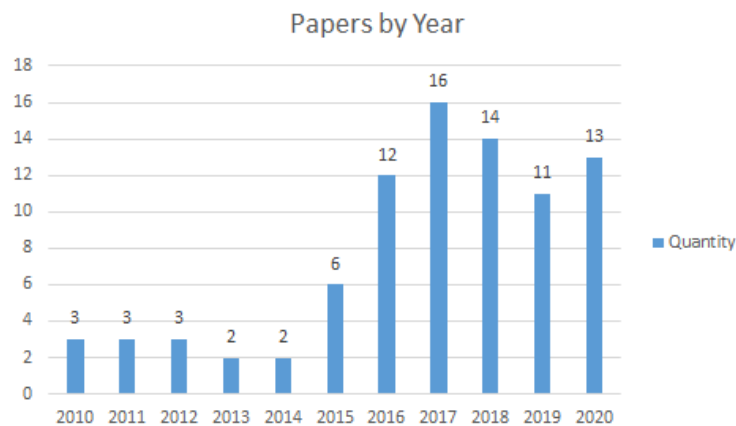
3.2 Análise quantitativa e qualitativa

Nesta seção são discutidos os resultados da SLR agrupados e categorizados para encaminhar as respostas para (QP2) “Quais os principais tipos de fraude abordados em pesquisas relacionadas a técnicas emergentes atualmente?” e (QP3) “Quais as principais técnicas emergentes adotadas efetivamente para melhorar métricas de prevenção de fraudes recentemente?”.

Conforme representado no diagrama da Figura 3.4 podemos ver um crescimento no número de pesquisas relacionadas a detecção de fraudes com técnicas emergentes ao longo do tempo. Esta tendência indica o interesse de ambas perspectivas (acadêmica e negócios) no desenvolvimento de abordagens mais efetivas para prevenção de fraudes e suas consequências.

Com a intenção de identificar o foco principal de estudos e lacunas potenciais para oportunidades de trabalhos futuros, no contexto de abordagens alinhadas com o escopo dessa SLR e as respectivas classes de fraude abordadas, uma análise de referência cruzada

Figura 3.4: Número de artigos por ano



Fonte: Elaborado pelo autor

foi feita por um gráfico barras empilhadas, no qual as cores representam as abordagens utilizadas e as colunas representam os tipos de fraude, este diagrama está representado na Figura 3.5.

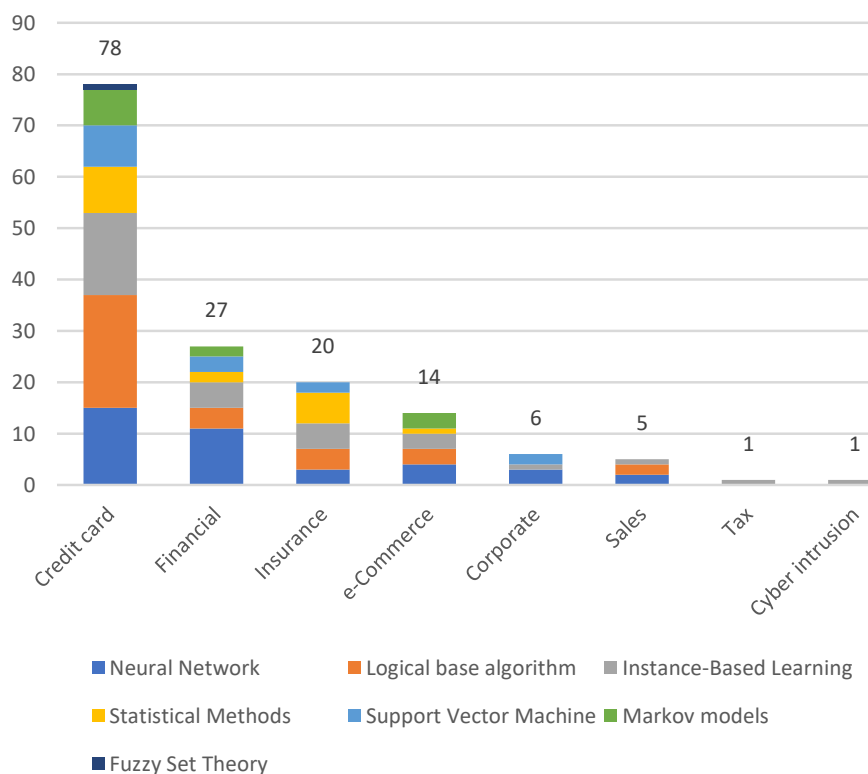
Nesta análise, o número total de ocorrências não bate exatamente com o número de artigos que resultaram da SLR, isso acontece porque em muitas das publicações o grupo de pesquisa responsável propôs múltiplas abordagens para endereçar o problema de detecção de fraudes e, para esta análise cada abordagem proposta foi contabilizada de forma separada, assim sendo, cada artigo pode ter sido contado em mais de um grupo de abordagens. Incluindo casos onde a abordagem proposta envolvia um método híbrido (*embedded*).

Além disso, houve casos em que uma única abordagem foi aplicada em experimentos que reproduziam múltiplos tipos de fraude. Desta forma, a partir de uma amostragem de 85 artigos acadêmicos, o gráfico listou 150 combinações únicas de abordagem de detecção de fraude por tipo de fraude onde a técnica foi utilizada.

O resultado dessa análise de referência cruzada entre abordagens e tipos de fraude demonstra que a grande maioria dos trabalhos de pesquisa tem foco em transações de cartão de crédito, seguido de perto por fraudes relacionadas a transações financeiras (ex: transações bancárias) e fraudes relacionadas a seguros residenciais e automotivos.

Entre as abordagens sugeridas para endereçar fraudes relacionadas a cartões de crédito, a abordagem mais representativa é a representada por soluções baseadas em lógica (ex: árvores de decisão e random forest) que se destacam pela sua acurácia balanceada, seguido de perto pelo número de técnicas baseadas em instâncias que são técnicas de aprendizado não-supervisionado em essência (ex: técnicas de clusterização como KNN), populares por não dependerem de dados rotulados para aprender.

Figura 3.5: Gráfico de barras empilhadas representando classes de fraude e os respectivos métodos encontrados na literatura para predição de fraude.



Fonte: Elaborado pelo autor

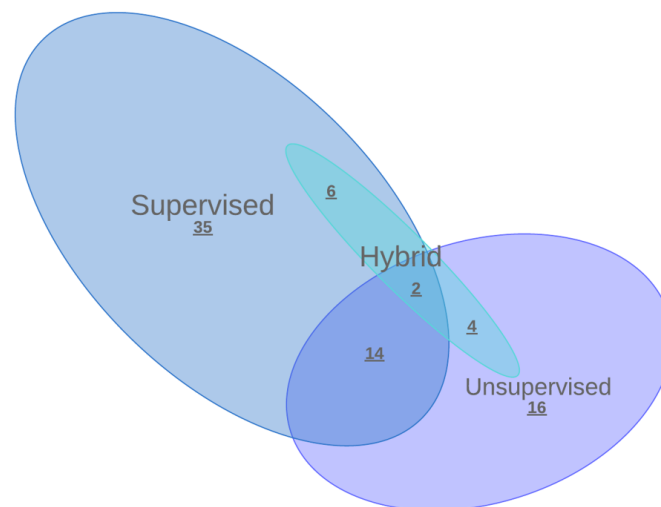
O último gráfico nesta seção representa um diagrama de Venn e mostra a distribuição das estratégias de aprendizado adotadas por cada abordagem de detecção de fraude proposta, conforme descrito na Figura 3.6.

De um total de 85 abordagens propostas por diferentes artigos nessa revisão bibliográfica, 77 (91%) tem a solução centrada em um método de aprendizado e 8 artigos focam em uma solução baseada em sistemas de decisão (*reasoning*), na grande maioria HMM e GA.

O gráfico em questão foca no método de aprendizado utilizado pela abordagem principal proposta por esses 77 trabalhos, dos quais 41 utilizam técnicas de aprendizado supervisionado e 20 utilizam técnicas de aprendizado não-supervisionado. Percebemos ainda que 16 das abordagens propostas combinam métodos de aprendizado supervisionado e não-supervisionado.

Uma categoria diferenciada foi destacada em azul representando algoritmos considerados híbridos (em alguns casos chamados de *embedded*) que se caracterizam por proporem uma nova abordagem que utiliza conceitos de diferentes algoritmos, essas abordagens se tornaram muito populares nos últimos anos. Ao todo, 12 abordagens foram mapeadas como híbridas.

Figura 3.6: Quantidade de artigos por método de aprendizado



Fonte: Elaborado pelo autor

3.3 Taxonomia

Para estruturar os resultados da revisão bibliográfica adequadamente, duas taxonomias foram propostas para mapear (1) a hierarquia das técnicas emergentes de detecção de fraude e (2) a hierarquia dos tipos de fraude onde abordagens de detecção de fraude estão sendo aplicadas. As próximas seções explicam estas duas taxonomias em detalhes.

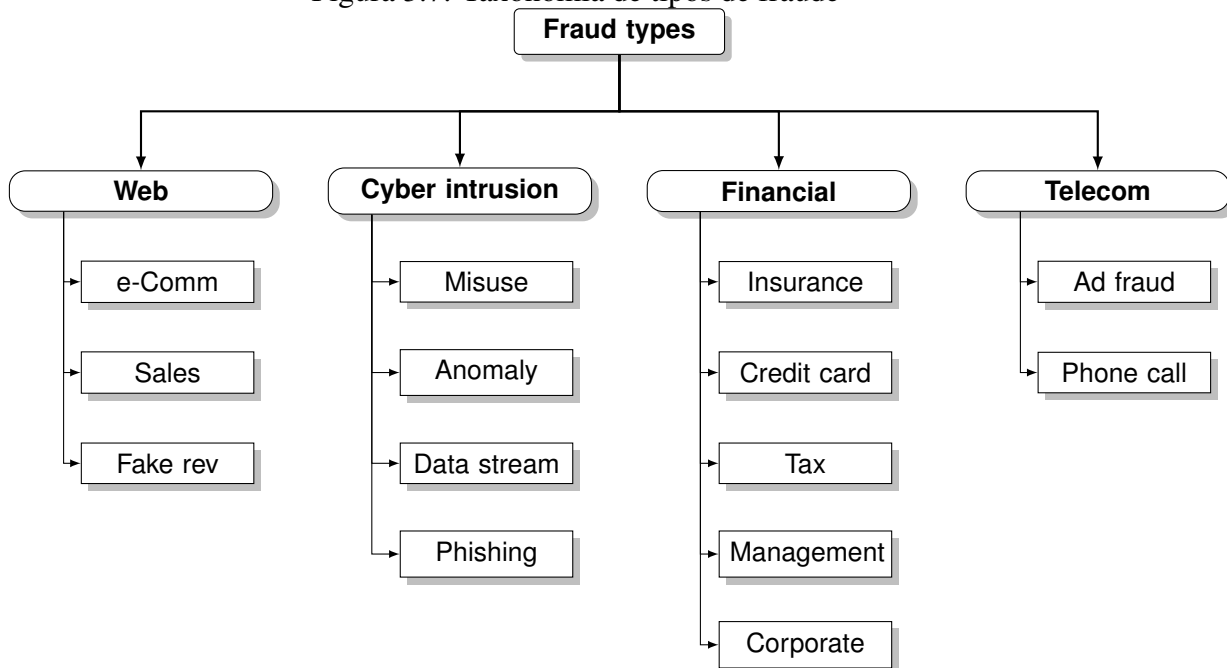
3.3.1 Taxonomia de tipos de fraude

Nesta seção, a estrutura da taxonomia de tipos de fraude abordadas por técnicas emergentes será explicada em detalhes. Existem vários tipos de fraude e múltiplas formas de combiná-los para compor uma taxonomia. Para exemplificar, no trabalho proposto por (BEALS; DELIEMA; DEEVY, 2015) uma taxonomia de fraudes proposta para organizar categorias de fraude sistematicamente em um *framework* que possa ser utilizado para endereçar as questões de pesquisa da nossa SLR e, desta forma, identificar oportunidades para trabalhos futuros.

Outro aspecto considerado na criação dessa taxonomia é garantir que a estrutura será simples e robusta o bastante para suportar o nível de detalhe necessário para segmentar os trabalhos pesquisados de forma que os dados nos ajudem na análise dos resultados. Além disso, essa hierarquia é útil para o leitor entender conceitualmente as categorias de fraude abordadas nessa pesquisa, delimitando as fronteiras entre fraude e outros tipos de transações financeiras errôneas ou enganosas, iniciadas propositalmente ou não, para

entender como esse comportamento pode evoluir para uma fraude.

Figura 3.7: Taxonomia de tipos de fraude



Fonte: Elaborado pelo autor

Uma taxonomia pode variar em estrutura, dependendo do objetivo ela foi planejada para atingir, para exemplificar a taxonomia proposta pelo time de pesquisadores em (BEALS; DELIEMA; DEEVY, 2015), consiste em apenas duas categorias agrupadas pelo alvo da fraude:

- Fraude contra um indivíduo,
- Fraude contra uma organização (empresa);

Com o propósito de categorizar todas as possibilidades de fraude de forma exaustiva e mutuamente exclusiva.

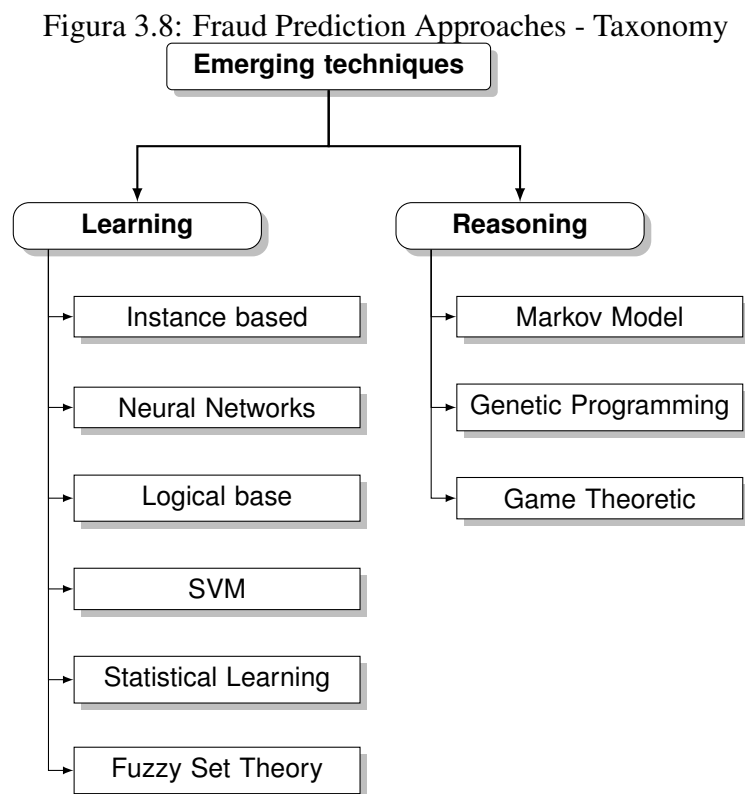
Para o propósito da nossa pesquisa, a taxonomia proposta foi estruturada para mapear como as abordagens baseadas em técnicas emergentes estão sendo aplicadas em diferentes contextos de fraude, para tanto o primeiro nível da taxonomia foi quebrado em 4 subcategorias:

- (1) Web,
- (2) Cyber intrusion,
- (3) Financial, and
- (4) Telecommunications;

Estas 4 categorias foram propostas em função da sua relevância em termos de

quantidade, e suas peculiaridades para garantir que na maior parte dos casos a classificação seja mutuamente exclusiva.

3.3.2 Taxonomia de abordagens de detecção de fraude - técnicas emergentes



Fonte: Elaborado pelo autor

A taxonomia de abordagens de detecção de fraude teve um foco em técnicas emergentes, cada abordagem proposta tem alguma relação com IA e seus subdomínios. Optamos por fazer “Técnicas Emergentes” a raiz da taxonomia, substituindo “Inteligência Artificial” utilizada na versão original.

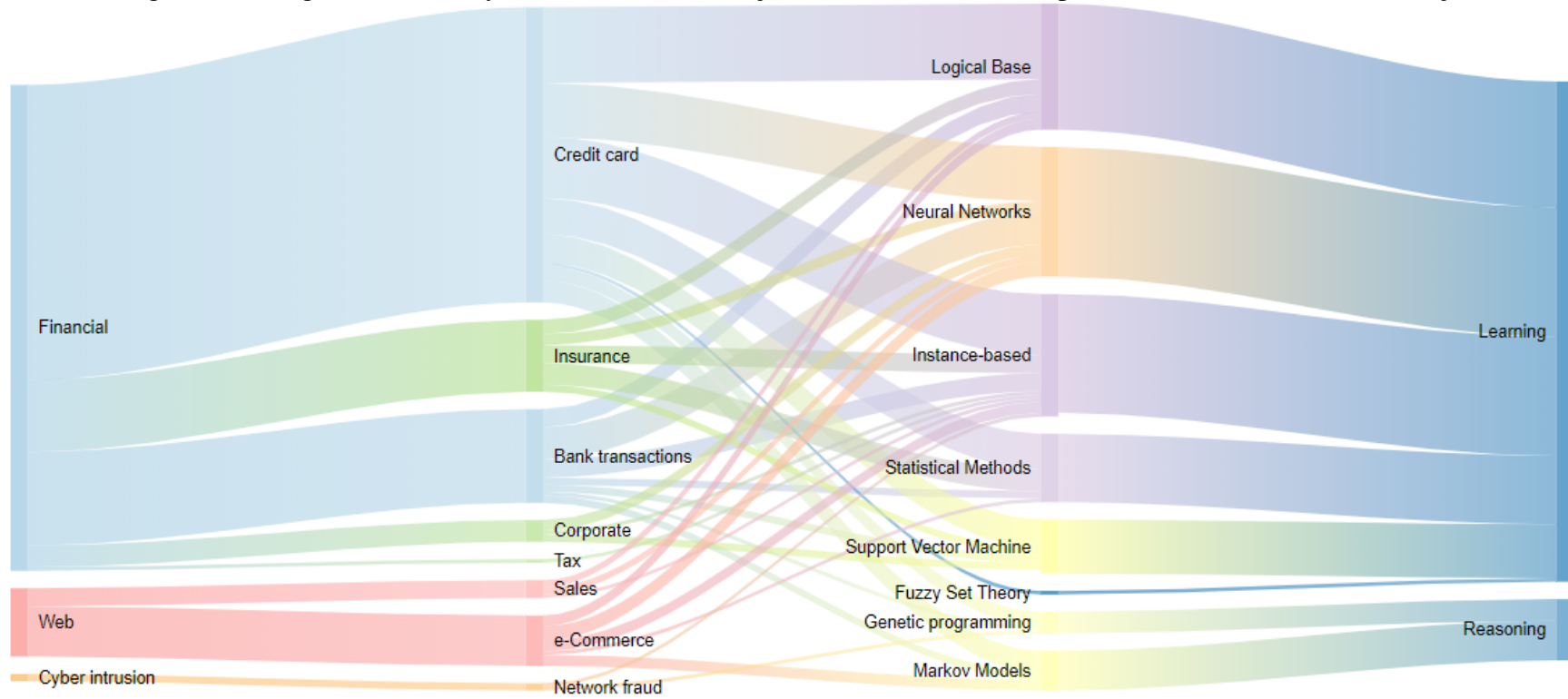
O segundo nível da taxonomia apresenta os 2 grandes grupos de abordagens (1) sistemas de aprendizagem e (2) sistemas de decisão (*reasoning*). Para o contexto desse exercício de classificação, as abordagens identificadas como sistemas de decisão (*reasoning*) são as que uma lógica de decisão seja criada de forma programática, como sistemas especialistas, lógica Fuzzy e sistemas de recomendação.

Já a categoria de sistemas de aprendizado é dividida em 5 subcategorias, são elas: Instance based, Redes Neurais, Logical base, Support Vector Machine e Aprendizado Estatístico.

Abaixo dos sistemas de decisão, existem 3 subgrupos, que correspondem a: Mar-

kov Models, Genetic Programming e modelos derivados de teoria dos jogos.

Figura 3.9: Diagrama de Sankey descrevendo interrelação entre taxonomias de tipos de fraude e técnicas de detecção



Fonte: Elaborado pelo autor

Como uma forma de mapear possíveis oportunidades de trabalhos utilizando abordagens diferentes em tipos de fraude poucos explorados, foi proposta uma visualização utilizando um diagrama de Sankey, veja Figura 3.9, para mapear como que os tipos de fraude se relacionam com as abordagens propostas em termos de número de trabalhos existentes para cada relação de tipo de fraude para abordagem de predição.

Esta visualização se baseou no número de abordagens por tipo de fraude extraídas da análise de referência cruzada que resultou em 150 registros em função dos trabalhos que propões múltiplas abordagens aplicados em um ou mais contextos de fraude.

3.4 Considerações finais

Esta revisão sistemática da literatura visa trazer luz à vasta gama de técnicas disponíveis para detectar fraudes sob métodos de aprendizagem e decisão (*reasoning*). Além do número significativo de algoritmos existentes, está se tornando uma prática comum combinar abordagens para alavancar o melhor que cada técnica tem a oferecer em uma solução híbrida ou *embedded*. O número de alternativas é tão alto que o primeiro desafio passa a ser definir o método que melhor se adapta às necessidades específicas do negócio em questão. Durante esta revisão, identificamos muitas pesquisas que definiram as técnicas a serem adotadas com base em requisitos específicos, que variam desde: proteger a privacidade do conjunto de dados, lidar com classes de dados altamente desequilibradas, lidar com o fluxo de dados em tempo real (ou *streaming*), até gerenciar o problema de “*concept drift*”².

Apesar da complexidade adicional que o número de alternativas pode trazer para os cientistas de dados, as técnicas emergentes tiveram sim uma contribuição efetiva para melhorar as métricas de previsão de fraudes em muitos aspectos. A principal delas diz respeito a sua capacidade de adaptação a novos padrões de fraude sem intervenção humana, reduzindo a quantidade de trabalho necessário para manter precisão, acurácia, sensibilidade (*recall*) entre outras métricas. Outro aspecto é a capacidade de derivar conhecimento de dados que, em alguns casos, mesmo um especialista com anos de experiência no domínio teria dificuldade para inferir. Outra contribuição relevante é a possibilidade de agregar múltiplos algoritmos em soluções combinadas, como sistemas de votação, me-

²Concept drift é um fenômeno em que a relação entre as entradas e saídas de um sistema muda ao longo do tempo. Em outras palavras, é quando o conceito ou padrão que o modelo de aprendizado de máquina foi treinado para reconhecer e prever começa a mudar ou evoluir, tornando o modelo menos preciso ou até mesmo inútil.

lhorando os resultados de modelos clássicos de prevenção de fraudes. Essas são sem dúvida contribuições-chave das técnicas emergentes neste domínio de combate a fraudes (QP1).

Os contextos de financiamento, cartão de crédito, transações on-line (como o comércio eletrônico) e até mesmo cripto-moedas são os que apresentam maior demanda por técnicas de controle de fraudes atualmente (QP2). E podemos dizer que, devido à falta de dados rotulados, há um interesse crescente por métodos não supervisionados, tornando assim algoritmos baseados em redes neurais, baseados em lógica e com base em instância (técnicas de clusterização) mais populares devido a sua capacidade de utilização em contextos onde dados rotulados não existem (QP3). Uma contribuição secundária deste levantamento bibliográfico está representada na taxonomia detalhada criada para estruturar e agrupar técnicas emergentes, disponível no apêndice Figura B.1.

4 TRABALHOS RELACIONADOS

Este Capítulo apresenta e compara os principais trabalhos relacionados ao contexto desta dissertação. Devido à profusão de trabalhos existentes no tópico de detecção de fraudes utilizando aprendizado de máquina e outras abordagens derivadas de IA, torna-se virtualmente impossível apresentar um comparativo geral de técnicas e resultados pesquisados. Desta forma, o escopo aqui se limita a destacar trabalhos correlatos que também tiveram foco em fraudes relacionadas a transações financeiras de crédito ou pagamento, distribuídos em 2 grupos: (1) abordagens que se destacam entre as propostas na sua classe de algoritmos e (2) trabalhos que compartilham ao menos um algoritmo utilizado nesse trabalho.

4.1 Abordagens que se destacam entre as propostas na sua classe de algoritmos

A lista de abordagens mencionadas nessa seção está sumarizada na Tabela 4.1. Em (SRIVASTAVA et al., 2008) os autores propuseram um novo método baseado em Hidden Markov Model (HMM) que representa o comportamento de um portador de cartão de crédito legítimo, a partir do qual discrepâncias (*outliers*) identificam fraude, adicionando ainda um componente de ajuste não supervisionado (K-means) a solução, para adaptar o modelo ao problema de mudança de comportamento gradual (*concept drift*).

Já em (SALAZAR et al., 2012), os autores propuseram uma combinação de técnicas que divide o processo de aprendizado sobre a autenticidade das transações em 4 estágios nomeados como: extração de *features*, treino e teste, fusão da decisão e apresentação do resultado. O algoritmo de aprendizado em questão é não-supervisionado e se baseia em um classificador *non-Gaussian mixture*.

A abordagem proposta em (SOHONY; PRATAP; NAMBIAR, 2018) combina dois métodos diferentes de aprendizado supervisionado, Random Forest e Redes Neurais para otimizar os resultados obtidos. Os autores utilizam cada classificador como um especialista, RF para indicar transações legítimas e RN para indicar transações fraudulentas calculando a predição como uma combinação dos dois valores.

Identificar fraudes em transações de cartão de crédito também é o foco do trabalho proposto em (ROY et al., 2018), o foco no caso é comparar os resultados obtidos por três classificadores baseados em redes neurais: (1) o primeiro baseado em redes neurais artificiais (ANN), o segundo baseado em redes neurais recorrentes (RNN) e finalmente

uma variação de RNN com um componente de longa memória de curto prazo. Este último modelo foi o que apresentou os melhores resultados, alcançando uma acurácia de 91.2%.

Em (SANTIAGO; PEREIRA; HIRATA, 2015), uma abordagem abrangente para endereçar o problema de fraude em transações de pagamento online é proposta utilizando Support Vector Machine (SVM). Os resultados obtidos foram validados a partir de experimentos em um conjunto de dados fornecido por uma grande empresa de pagamentos on-line da América Latina.

Finalmente, em (RANDHAWA et al., 2018) o método AdaBoost (*Adaptative Booster*) é utilizado como uma árvore com apenas dois níveis para melhorar o resultado do método de maioria dos votos usado na combinação de resultados de vários classificadores. Os experimentos são então executados sobre uma base de transações de cartão de crédito gerada de forma sintética para esse propósito.

Tabela 4.1: Trabalhos relacionados - Abordagens que se destacam entre as propostas na sua classe de algoritmos

Referência	Abordagem	Técnica	Tipo de Fraude
(SRIVASTAVA et al., 2008)	Reasoning	HMM	Cartão de crédito
(SALAZAR et al., 2012)	Embedded	non-Gaussian mixture	Cartão de Crédito
(SOHONY; PRATAP; NAMBIAR, 2018)	Embedded	Random Forest + Redes Neurais	Cartão de Crédito
(ROY et al., 2018)	Neural Networks	Recurrent Neural Networks (RNN)	Cartão de Crédito
(SANTIAGO; PEREIRA; HIRATA, 2015)	Learning	SVM	Cartão de Crédito
(RANDHAWA et al., 2018)	Learning	AdaBoost	Cartão de Crédito

4.2 Abordagens que compartilham ao menos um algoritmo utilizado nesse trabalho

A lista de abordagens mencionadas nessa seção está sumarizada na Tabela 4.2.

Em (XUAN et al., 2018) os autores adotam 2 tipos de algoritmo Random Forest para treinar o comportamento das *features* para transações normais e fraudulentas, permitindo traçar um paralelo entre os dois classificadores e estabelecer uma decisão de exceção para aquelas transações onde não houver um consenso entre as previsões para otimizar o seu desempenho.

Os autores de (KAVITHA; SURIKALA, 2018b) propuseram um meta classifi-

cador baseado em árvore que opera com predições em 2 níveis. Os resultados obtidos a partir do primeiro e segundo nível de predição são combinados para obter uma decisão final ponderando ambas. De forma geral, a decisão combinada apresenta maior acurácia do que as predições individuais.

Já os autores de (KAVITHA; SURIKALA, 2018a) propuseram um inovador modelo *ensemble* com múltiplos níveis chamado de Bagging Multiple Boosted Trees (BMBT) que remete ao conceito de um modelo XGB. Comparativos com modelos baseado em árvore padrão como RF e XGB indicam uma otimização na ordem de 1% para 5% nas métricas: AUC, BCR, BER e TNR.

No artigo (SHERLY; NEDUNCHEZHIAN, 2010), um sistema que combina técnicas de classificação supervisionada e clusterização é proposta pelos autores como um solução de dois estágios para detecção de fraudes, usando o algoritmo BOAT (Optimistic Decision Tree Construction) para identificar um fator de anomalia que indique potencial de fraude e então consultar em uma base de dados histórica pela ocorrência daquela combinação de valores como um passo de validação antes de uma predição de fraude ser feita.

Um *framework* de aprendizado *ensemble* é proposto pelos autores de (Wang et al., 2018) baseado no particionamento dos dados de teste e clusterização utilizando RF como componente de classificação principal. A efetividade do *framework* proposto é provado através do resultado de experimentos realizados em uma base de transações de cartão de crédito real.

Em (MELO-ACOSTA; DUITAMA-MUNOZ; ARIAS-LONDONO, 2017) uma metodologia é proposta para a automação do processo de detecção de traduções fraudulentas baseada em um algoritmo Random Forest Balanceado. Além disso, uma plataforma de alta escalabilidade utilizando Spark e Hadoop como repositório de dados permite que o volume de dados requerido seja processado em tempo real (ou, próximo a tempo real). Atingindo uma melhoria de cerca de 24% em termos de mediana geométrica se comparada a um algoritmo RF padrão.

No artigo (POZZOLO et al., 2015) os seus autores projetam dois sistemas de detecção de fraudes (FDSs) baseados em um método combinado (*ensemble*) de classificação com abordagem de movimentação de janela, a partir da qual foi possível demonstrar que a abordagem vitoriosa consiste em treinar dois classificadores separadamente (baseados em retro-alimentação e rótulos atrasados, respectivamente) para, ao final, agregar as decisões dos dois classificadores. Os experimentos indicaram uma substancial melhoria na métrica de ‘alertas de fraude’.

A proposta de (KHO; VEA, 2017) representa uma sugestão para o modelo de detecção capturar transações anômalas como um plano de contenção caso o sistema de prevenção de fraudes primário falhe. Diversos classificadores foram avaliados como candidatos para a criação do modelo, contudo, apenas RF obteve indicadores de acurácia acima de 93%.

O único representante de uma rede neural (MLP) nesta lista foi proposto por (MONTINI et al., 2013), onde os autores usam MLP para uma tarefa de análise de similaridade em transações bancárias partindo de padrões de dados de referência para uma transação legítima e uma transação fraudulenta.

Tabela 4.2: Trabalhos relacionados - Abordagens que compartilham ao menos um algoritmo utilizado nesse trabalho

Referência	Abordagem	Técnica	Tipo de Fraude
(XUAN et al., 2018)	Logical base	Parallel Random Forest	Cartão de Crédito
(KAVITHA; SURIKALA, 2018b)	Logical base	Tree Based Meta Classifier	Cartão de Crédito
(KAVITHA; SURIKALA, 2018a)	Embedded	Bagging Multiple Boosted Trees	Cartão de Crédito
(SHERLY; NEDUNCHEZHIAN, 2010)	Embedded	BOAT	Cartão de Crédito
(Wang et al., 2018)	Embedded	Random Forest	Cartão de Crédito
(MELO-ACOSTA; DUITAMA-MUNOZ; ARIAS-LONDONO, 2017)	Alta escalabilidade	Balanced Random Forest	Cartão de Crédito
(MONTINI et al., 2013)	Neural Networks	Multilayer Perceptron	Transações Bancárias
(ZAKARYAZAD; DUMAN, 2016)	Neural Networks	MLP / ANN	Cartão de crédito

4.3 Considerações finais

Este capítulo apresentou dois contextos de trabalhos relacionados: (1) abordagens que se destacam entre as propostas na sua classe de algoritmos e (2) trabalhos que compartilham ao menos um algoritmo utilizado na proposta de solução deste trabalho. Ao todo 14 trabalhos foram apresentados entre as duas categorias, cobrindo os seus detalhes e especificidades. É importante mencionar que no apêndice dessa dissertação estão listados

os 60 trabalhos de proposta que fizeram parte da revisão sistemática da literatura em tabelas que mostram os seus atributos mais relevantes, como categoria de abordagem a que pertence, algoritmo utilizado no cerne da solução e o tipo de fraude no qual a abordagem foi aplicada.

Muito embora não seja possível comparar os resultados em termos de métricas entre os trabalhos, pois, os dados que representam o contexto de fraude variam de um trabalho para o outro. É possível afirmar que a proposta apresentada nessa dissertação agora figura entre as abordagens aqui revisadas, servindo de referência para trabalhos futuros, principalmente nesse contexto de crédito financeiro corporativo, menos explorado, por outros trabalhos, pelos resultados obtidos durante a pesquisa conduzida na execução da SLR.

5 MODELO DE PONTUAÇÃO BASEADO EM REGRAS (MPBR)

O objetivo desse capítulo é apresentar a solução corporativa baseada em um motor de regras de decisão que representa o cerne do sistema de combate a fraude para o caso de negócio explorado por esse trabalho. Neste capítulo são apresentados ainda detalhes sobre o caso de negócio e como o processo de prevenção de fraudes é tratado atualmente. A seção é finalizada com uma análise de desafios e oportunidades identificados na solução de fraudes baseada em regras de decisão atualmente utilizada.

5.1 FICO Blaze Advisor

O FICO Blaze Advisor (ANDREESCU; MIRCEA et al., 2009) é uma plataforma de gerenciamento de regras de negócio, desenvolvida pela empresa FICO. Esta plataforma permite que os usuários criem e gerenciem regras de negócio em tempo real, para automatizar decisões empresariais complexas. Também pode ser estendido para contemplar funcionalidade de análise de dados e simulações, além de ser acoplável a outras ferramentas do pacote de gerenciamento de decisões FICO.

A solução corporativa de regras de decisão em questão foi desenvolvida sobre a plataforma FICO Blaze advisor que é considerada uma das principais opções para implementação de regras de decisão disponíveis no mercado, com foco em aprovação de crédito e avaliação de risco inerente, incluindo o contexto de fraude.

O FICO Blaze Advisor é um motor de regras de decisão configurado para tratar transações de acordo com um conjunto de estratégias, um modelo de classificação calculado com dados das transações no formato de variáveis que sumarizam os dados crus transformando os mesmos em informações mais relevantes para o processo de decisão. Para critério de comparação, estabelecemos os experimentos realizados no MPBR como *baseline*, considerando o sistema atual como ponto de partida para a análise dos resultados obtidos usando técnicas de aprendizado de máquina.

5.2 Customizações do framework implementadas para a necessidade do negócio

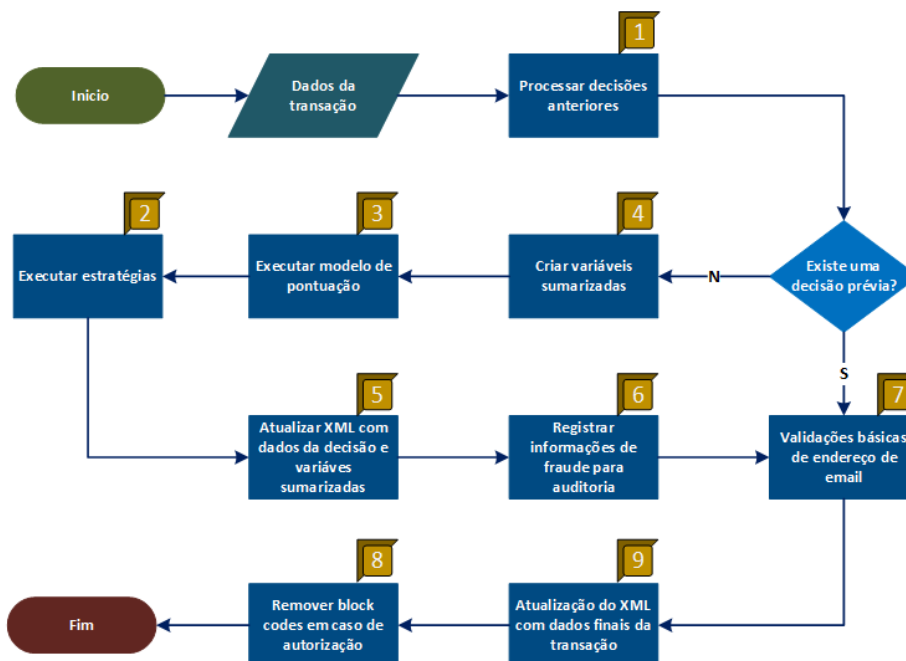
Através da adoção do *framework* de regras de decisão da FICO uma sofisticada solução de aprovação de transações foi desenvolvida na forma de um Modelo de Pontu-

ação Baseado em Regras. A solução proposta para a necessidade corporativa tem como pilar principal o pré-processamento de informações recebidas pelo fluxo de vendas sobre o cliente, produtos, contatos, endereços e outras fontes de dado que compõem a estrutura mestre de dados inerentes a uma transação. Dessa forma, os dados crus recebidos no momento do processamento das transações são transformados em variáveis sumarizadas que extraem conhecimento específico dos dados originais, que a partir desse momento são completamente desconsiderados. Podemos dizer que o *design* proposto para o serviço de decisão usa os dados brutos apenas para derivar as variáveis sumarizadas, também chamadas de calculadas, que representa o primeiro passo do fluxo de trabalho descrito na Figura 5.1. As variáveis sumarizadas se dividem em 3 grandes grupos, o primeiro deles diz respeito aos dados adquiridos no momento em que o cliente está submetendo a transação de compra (*Cart*), 46 variáveis são calculadas baseadas em dados submetidos pelo cliente/comprador. O segundo grupo de variáveis diz respeito a dados da transação (*Order*), ou pedido, incluem dados do cliente e produtos que estão sendo comprados, além de outras fontes de dados externas que são carregados a partir de parceiros especializados na detecção de fraudes ao todo 23 variáveis sumarizadas fazem parte dessa seção. O terceiro grande grupo é composto por informação dos componentes (*SKUs* - do inglês *stock keeping units* ou unidades de manutenção de estoque) que fazem parte da transação, é importante ressaltar aqui que o conceito de componente é bastante amplo e não se resume apenas as peças que compõem um produto como periféricos, por exemplo, mas pode se estender a aspectos da garantia dos produtos incluídos da compra e até mesmo aspectos relacionados a forma e condições de pagamento. Esta terceira e última seção de variáveis sumarizadas contempla 108 variáveis. Grupos de variáveis e suas respectivas quantidades:

- Variáveis relacionadas ao carrinho de compras (*Cart*) - Total: 46 variáveis.
- Variáveis relacionadas a ordem de compra (transação) - Total: 23 variáveis.
- Variáveis relacionadas a informações de componentes dos produtos (*SKUs*) - Total: 108 variáveis.
- Modelos de pontuação (*MPCN* e *MPCA*) - Total: 2 variáveis.
 - **MPCN**: Modelo de Pontuação de Conta Nova
 - **MPCA**: Modelo de Pontuação de Conta Antiga
- Total geral: 179 variáveis.

O passo seguinte no fluxo de decisão trata da execução do modelo de pontuação. O

Figura 5.1: Fluxo de trabalho do sistema de regras de decisão



Fonte: Elaborado pelo autor

modelo de pontuação são armazenados em duas variáveis sumarizadas que são calculadas com base nos valores das demais calculadas anteriormente para essa transação. Estes dois modelos são identificados pelas seguintes siglas: MPCN (Modelo de Pontuação de Conta Nova) e MPCA (Modelo de Pontuação de Conta Antiga). O MPCN é utilizado para avaliar transações relacionadas a contas de cliente que tenham sido criadas nos últimos 30 dias, enquanto o MPCA é utilizado para avaliar transações de contas de clientes que tenham sido criadas a mais de 30 dias. Essa diferenciação entre os modelos de pontuação é feita para focar em aspectos inerentes de diferentes tipos de fraude, para contas novas o tipo de fraude mais frequente é o de roubo de identidade (*identity theft*) que consiste do fraudster personificar um perfil que não condiz com as suas características. Já para contas consideradas antigas, o tipo de fraude mais frequente é comumente chamada de roubo de conta (*account takeover*), que se caracteriza pelo roubo de informações de uma conta existente que já teve transações aprovadas e pagas anteriormente. Os valores das variáveis sumarizadas nesses dois contextos são significativamente diferentes, justificando a criação de um modelo de pontuação específico para cada padrão. Estatisticamente falando, o tipo de fraude popularmente conhecida como roubo de identidade é muito mais frequente, contemplando mais de 95% do total de fraudes identificadas pelo sistema. Esse número pode ser explicado pela complexidade envolvida em um processo de roubo de conta. Em

compensação, fraudes baseadas no roubo de contas são muito mais difíceis de serem rastreadas, havendo casos em que os clientes levam meses para identificar o roubo e várias transações são efetivamente realizadas pelos fraudadores nesse meio tempo.

O passo seguinte do fluxo de decisão baseada no modelo de pontuação diz respeito a execução de estratégias. São denominadas estratégias as regras criadas para que identificar se alguma das variáveis sumarizadas atingiu ou excedeu um valor limite considerado pelo modelo atual. As estratégias são em geral expressões booleanas que retornam apenas um valor que pode ser verdadeiro ou falso.

Um exemplo de estratégia de fraude é utilizada para comparar endereços do cadastro do cliente com o endereço de entrega utilizado no pedido, essa comparação acontece tanto na rua quanto no CEP (*ZIP code*) do endereço. Se a diferença for constatada, a estratégia é automaticamente marcada como tendo sido atingida (dizemos que a estratégia teve um *hit*). Existem ao todo 107 estratégias no modelo de pontuação atual. As estratégias se dividem em 5 grupos:

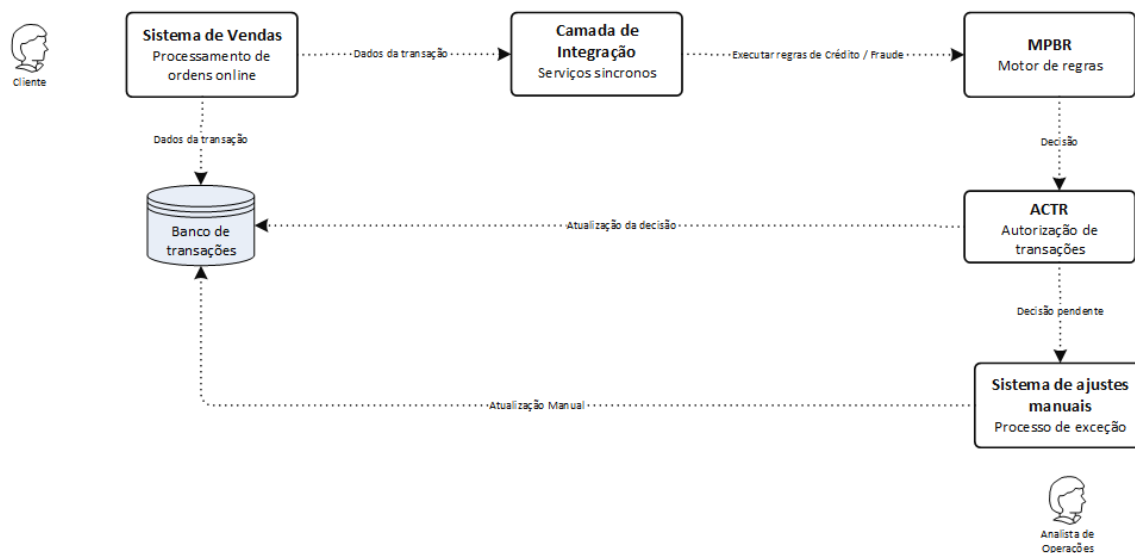
- Modelo de pontuação para conta nova (MPCN) - Total: 34 estratégias.
- Modelo de pontuação para conta antiga (MPCA) - Total: 10 estratégias.
- Estratégias aplicáveis tanto para MPCN quanto para MPCA - Total: 55 estratégias.
- Estratégias gerais - Total: 4 estratégias.
- Estratégias de relatório - Total: 4 estratégias.

As estratégias possuem um código utilizado para determinar a precedência quando múltiplas estratégias são ativadas em uma mesma validação. Essa priorização da estratégia a ser considerada pode ser útil para adicionar detalhes a decisão tomada pelo MPBR.

O passo seguinte do fluxo consiste em atualizar o XML utilizado na integração dos sistemas para incorporar o código da decisão, além dos detalhes sobre as variáveis sumarizadas e estratégia(s) pertinentes para essa decisão. Existem 2 decisões possíveis para regras de fraude, a transação pode ser aprovada automaticamente, ou pode ainda ser identificada como pendente. Uma transação pendente exige intervenção manual do time de retaguarda (também conhecido como time de operações). Uma transação pendente por fraude também pode ser chamada de *fraud referral*. Na sequência, o fluxo endereça questões de conformidade legal, como a decisão pode ter consequências legais caso um cliente exija entender as razões pelas quais uma transação em seu nome não foi aprovada imediatamente, o sistema precisa necessariamente registrar aspectos que permeiam a decisão tomada para informar autoridades, essa informação pode ser utilizada caso um cliente

se sinta lesado e processe a empresa na justiça. O passo seguinte do fluxo consiste em uma validação do endereço eletrônico do cliente, para garantir que o time de retaguarda tenha a informação correta para entrar em contato, caso necessário. É importante ressaltar que essa validação do email não tem nenhuma relação com a verificação da fraude propriamente dita. O último passo do fluxo trata da atualização da seção do XML que inclui informações sobre o crédito disponível do cliente para atualizar conforme a execução de regras de fraude que acabou de acontecer. Através dessa atualização as informações são propagadas para os sistemas responsáveis pela manutenção de dados de crédito. O diagrama descrito na imagem 5.2

Figura 5.2: Diagrama de integração dos sistemas



Fonte: Elaborado pelo autor

Nesta representação de alto nível da integração dos sistemas podemos ver os principais componentes que interagem com o MPBR para enviar a informação necessária para verificar a autenticidade de uma transação e posteriormente utilizar a decisão para que o fluxo possa ter sequencia de forma automatizada (*caso a transação seja legitima*) ou manual (*caso a transação seja fraudulenta*). Além do MPBR, estão representados nesse diagrama, o sistema de vendas, a camada de integração (serviços ou *SOA*), o sistema de autorização de compras em tempo real e a interface onde os usuários podem entrar decisões manuais baseado na resposta obtida dos clientes.

5.3 Caso de uso do negócio - predição de fraudes em transações comerciais

O processo de obtenção de crédito e o uso do crédito para efetuar uma transação acontece em duas etapas. Na primeira etapa o crédito precisa ser aprovado através de um processo chamado aplicação de crédito. Na grande maioria dos casos o processo de aprovação de crédito é instantâneo e logo após a requisição ser submetida o crédito já pode ser utilizado pelo cliente, exceções acontecem quando existe algum problema com o perfil do cliente. A segunda etapa consiste da utilização do crédito para efetuar alguma compra que pode ser paga através de uma fatura do cliente. Como o propósito principal é que o cliente possa usufruir do crédito para efetuar suas compras, é possível aplicar para crédito no meio da transação modificando o método de pagamento no momento que a transação é efetuada.

Essa informação é importante para o processo de identificação de fraude porque em ambas as etapas existem procedimentos para prevenir fraudes. No escopo desse trabalho, apenas as fraudes no âmbito da transação de compra estão sendo consideradas.

Uma das diretivas para determinar o valor limiar aceitável de transações, enviadas para revisão manual pelo time de operações, é o custo da fraude para o negócio em termos de retenção de clientes. Historicamente a quantidade de clientes perdidos no processo de aplicação de crédito gira em torno de 15% enquanto no âmbito de transações, quando os clientes já foram efetivados e tem uma conta de crédito, esse valor cai para cerca de 5%. Para minimizar a perda de clientes causada por decisões que bloqueiam transações como fraudulentas de forma indevida, as regras definidas pelo MPBR visa manter o número de *referrals* em torno de 5% do total de transações.

O monitoramento de fraudes não detectáveis pelas regras atualmente em vigor se dá de forma manual, exigindo que uma equipe de combate a fraude esteja constantemente monitorando picos de transações fraudulentas e seus respectivos padrões. Baseado nesse monitoramento, a equipe de monitoramento pode optar por criar uma regra específica que consiga bloquear o padrão identificado. Essas regras podem ser ativadas de forma temporária ou permanente, mas a sua existência e propósito mostram o quão artesanal o processo de combate a fraude acaba se tornando através do MPBR. Além dessas revisões esporádicas, existe uma equipe de especialistas do negócio que inclui pessoas com formação em estatística para atualizar as regras anualmente, principalmente as regras utilizadas para calcular o Modelo de Pontuação para Contas Novas e Antigas (MPCN e MPCA).

5.4 Considerações Finais

O sistema de decisões MPBR escolhido como *baseline* para validar os resultados obtidos pelo modelo de aprendizado de máquina por representar a solução atualmente utilizada no âmbito corporativo se baseia em regras definidas por especialistas no domínio do negócio.

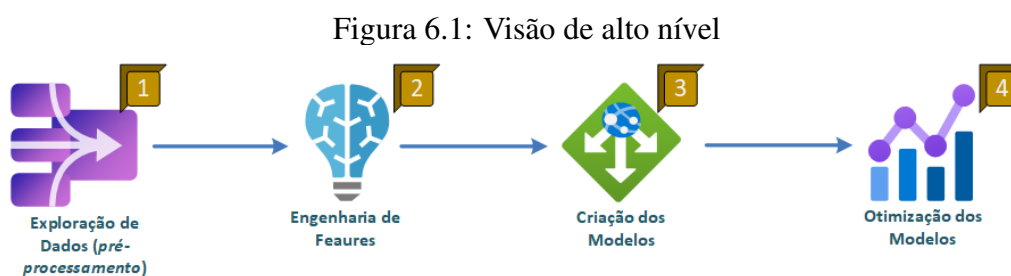
Se uma transação se enquadra em uma ou mais dessas regras, ela é rejeitada e rotulada como fraudulenta. Caso contrário, a transação é aprovada e concluída. O coração desse sistema baseado em regras está na lista de regras e mantidas meticulosamente por pessoas com vasto conhecimento sobre o domínio do negócio. Por outro lado, a desvantagem é a falta de flexibilidade, já que a estratégia de fraudadores o tempo todo para burlar as regras e obter crédito indevidamente. Em um sistema baseado em regras, há um atraso significativo na transação de fraude e na atualização das regras. Uma vez que um grupo de especialistas é responsável pela manutenção das mesmas, leva tempo para realizar a análise manual de dados fraudulentos para detectar e compreender novos padrões e para formular novas regras. Uma vez que as regras são ajustadas para contemplar novos padrões que definem o que é fraude, leva tempo para incorporar os ajustes as regras do MPBR.

6 UMA SOLUÇÃO DE PROCESSAMENTO DE TRANSAÇÕES DE CRÉDITO PARA PREDIÇÃO DE FRAUDES

Neste capítulo, é detalhada a proposta de solução desenvolvida para predição de fraudes em transações de compra financiadas utilizando um modelo desenvolvido com técnicas de “Aprendizado de Máquina”, mais especificamente por técnicas de classificação supervisionada baseada em lógica e redes neurais. O foco principal gira em torno das atividades desenvolvidas para obtenção e processamento dos dados de entrada, métodos utilizados para derivar conhecimento na forma de *features* e a pesquisa desenvolvida para identificar algoritmos apropriados para a modelagem e aplicá-los para gerar modelos eficazes na tarefa de predição de transações de crédito *online* fraudulentas.

6.1 Visão geral - (Metodologia)

O objetivo dessa seção é descrever as 4 etapas principais da solução de predição de fraudes em transação de crédito conforme a Figura 6.1. Esta solução é desenvolvida com técnicas (*Aprendizado de Máquina*), visando extrair conhecimento estatístico de dados de transações históricas para prever quando uma transação *online* deve ser considerada fraudulenta. É importante ressaltar que as etapas que compõem essa solução são demonstradas experimentalmente no Capítulo 7.



Fonte: Elaborado pelo autor

A solução proposta é estruturada em quatro etapas principais. A etapa de “Obtenção dos Dados” representada na Figura 6.1 (1) diz respeito ao mapeamento de todas as informações disponíveis sobre a transação no momento em que uma decisão sobre a autorização de uso de crédito acontece, para que essa informação possa então ser usada no processo de predição. Esses dados são então classificados e documentados, para que as atividades de pré-processamento possam ser conduzidas para transformar, integrar, criar uma estrutura de dados tabular, restringir acesso a dados restritos e rotular as transações.

A etapa de “Engenharia de *Features*”, representada na Figura 6.1 (2), trata das atividades relacionadas ao tratamento das *features* que efetivamente alimentam os modelos de Aprendizado de Máquina. Destaque para os métodos de “Seleção de *Features* [A]” e “Extração de *Features* [B]” que visam, respectivamente, [A] identificar as *features* mais relevantes para uma predição, descartando entradas com potencial para causar viés ou ruído ao modelo e [B] gerar novas *features* baseadas em composições e transformações dos dados originais da transação.

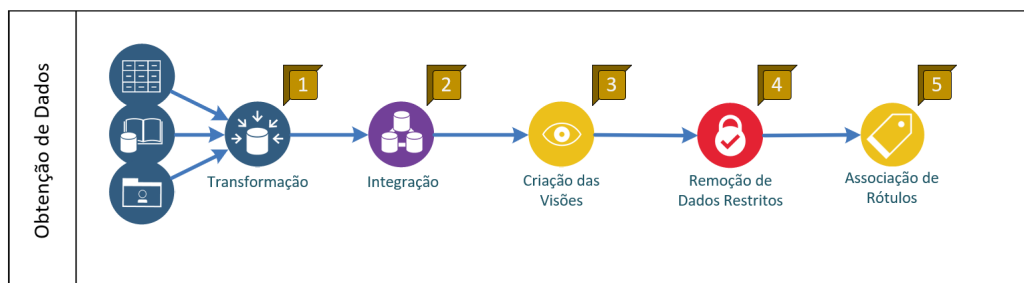
A etapa de “Criação dos Modelos”, representada na Figura 6.1 (3), é responsável por gerar os modelos, definindo os algoritmos utilizados e alimentando-os com os dados obtidos nas etapas de “Obtenção de Dados” Figura 6.1 (1) e “Engenharia de *Features*” Figura 6.1 (2). O resultado dessa etapa é um modelo treinado capaz de fazer predições.

Finalmente, a etapa de “Otimização dos Modelos”, representada na Figura 6.1 (4), tem como principal desafio a otimização de hiperparâmetros e utilização de diferentes intervalos de dados para treinamento e testes para melhorar os resultados dos diferentes modelos validados. O resultado dessa etapa é um modelo alvo completo, contemplando hiperparâmetros e contexto de dados para ser utilizado para predições em tempo real de transações fraudulentas.

6.2 Obtenção de Dados

O objetivo dessa seção é detalhar as fases que fazem parte da etapa de “Obtenção de Dados”, conforme ilustrada na Figura 6.2. As 5 fases que compõem a etapa de “Obtenção de Dados” são descritas a seguir.

Figura 6.2: Etapa 1 - Obtenção de dados



Fonte: Elaborado pelo autor

A primeira fase da etapa de “Obtenção de Dados” é denominada “Transformação” (Figura 6.2 (1)) sendo caracterizada pela transformação e centralização dos dados para simplificar a formatação dos dados e facilitar as suas análises. A partir de um repositório

centralizado, a fase de “Integração” (Figura 6.2 (2)) de dados tem início para mapear como os diferentes conjuntos de dados se relacionam entre si e criar as respectivas associações em um modelo de dados único.

A fase de “Criação das Visões” (Figura 6.2 (3)) propõe uma reorganização dos dados em uma visão tabular que permita combinar os atributos de tabelas com diferentes cardinalidades, gerando uma abstração da transação em um registro único. A fase de “Remoção de Dados Restritos” (Figura 6.2 (4)) garante que dados restritos ou sigilosos sejam removidos, ou mascarados para impedir que sejam expostos de forma indevida. São exemplos de dados restritos: informações pessoais (idade, raça, localização) ou ainda dados confidenciais como o número de seguridade social (para cidadãos americanos) e códigos de cartão de crédito.

A última fase da etapa de obtenção de dados é a “Atribuição de Rótulos” (Figura 6.2 (5)) que é utilizada para diferenciar as classes de registros que fazem parte do processo de predição. Para o problema em questão foi necessário identificar uma lógica para mapear os registros na visão tabular como “Fraudulentas” ou “Legítimas”.

6.3 Engenharia de *Features*

O objetivo dessa seção é detalhar as fases que fazem parte da etapa de “Engenharia de *Features*”, conforme ilustrada na Figura 6.3. As 6 fases que compõem a etapa de “Engenharia de *Features*” são descritas a seguir.

Figura 6.3: Etapa 2 - Engenharia de features



Fonte: Elaborado pelo autor

A etapa de “Engenharia de *Features*” diz respeito ao cerne da disciplina de ciência de dados, ilustrada na Figura 6.3. Essa atividade envolve a adoção de técnicas de redução de dimensionalidade como “Seleção e Extração de *features*” críticas para reduzir a complexidade do modelo de aprendizado de máquina e otimizar seus resultados. Muito embora estas técnicas possam ser usadas intercaladamente e em ciclos, para o propósito

dessa pesquisa, a “Seleção de *Features*” (Figura 6.3 (1)) foi priorizada como uma forma de reduzir a quantidade de dados para serem tratados nas fases seguintes.

Após uma pré-seleção baseada principalmente em disponibilidade, homogeneidade e propósito das *features*, a fase seguinte foca na “Classificação” (Figura 6.3 (2)) das mesmas pelo tipo de dados (numérico, categórico e suas variações). As *features* numéricas são prioritárias para as fases de “Análise de Correlação” (Figura 6.3 (3)), “Medição de Relevância” ((Figura 6.3 (4))) e “Normalização” (Figura 6.3 (5)) dos dados. Estas três fases, embora envolvam técnicas distintas, foram conduzidas de forma coordenada em função da sua interdependência.

Embora o resultado da “Análise de Correlação” (Figura 6.3 (3)) e “Medição de Relevância” ((Figura 6.3 (4))) mapeie as *features* com contribuição significativa para predição de fraudes, entradas menos relevantes devem ser mantidas para análise futura sobre sua influência na identificação de novos padrões de fraude.

Na fase de “Extração de *Features*” (Figura 6.3 (6)), técnicas de análise de simetria e valores discrepantes são adotadas para reduzir o conjunto, enquanto *features* do tipo categórico (principalmente: datas, booleanas e nominais) podem ser exploradas para extração de novas *features* usando técnicas de codificação como *Onehot*¹ e combinações de correspondência.

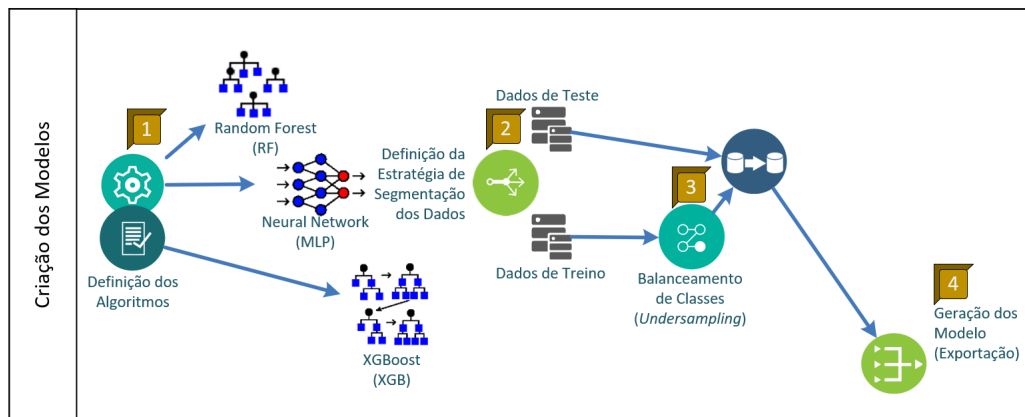
6.4 Criação dos Modelos

O objetivo dessa seção é detalhar as fases que fazem parte da etapa de “Criação dos Modelos”, conforme ilustrada na Figura 6.4. As 4 fases que compõem a etapa de “Criação dos Modelos” são descritas a seguir.

Na etapa de “Criação dos Modelos”, o primeiro passo deve ser a “Definição de Algoritmos” (Figura 6.4 (1)) que serão utilizados para conceber os modelos primários. É interessante que mais de um algoritmo seja selecionado, possibilitando a comparação dos resultados obtidos por múltiplos modelos. Para os experimentos descritos no Capítulo 7 foram escolhidos 3 algoritmos: *Random Forest* - **RF**, *Extreme Gradient Boost* - **XGB** e *Multilayer Perceptron* - **MLP**, considerados adequados ao problema da predição de fraude no contexto de transações financeiras.

¹Codificação Onehot pode ser definida como uma forma de conversão de variáveis categóricas em binárias multiplicando o número de *features* geradas pelo número de entradas que se deseja representar. Possibilitando alimentar algoritmos de aprendizagem de máquina que, em muitos casos, não trabalham bem com dados categóricos.

Figura 6.4: Etapa 3 - Criação dos Modelos



Fonte: Elaborado pelo autor

A fase seguinte diz respeito a “Definição de Estratégia de Segmentação dos Dados” (Figura 6.4 (2)) através da qual o conjunto de dados disponível é dividido em fatias que possibilitem treinar e testar o modelo para garantir resultados confiáveis e melhorar as métricas. Essa divisão lógica dos dados deve considerar, pelo menos, critérios como: intervalos de tempo, volume de dados e amostragem das classes de predição (*transações fraudulentas e legítimas*).

Em virtude da natureza altamente desbalanceada dos dados (cerca de 2% dos dados representam fraudes), inerente ao problema da predição de fraudes, na qual o volume de transações fraudulentas é muito inferior ao volume das transações legítimas, é válido dedicar uma fase ao “Balanceamento dos Dados” (Figura 6.4 (3)). Existem inúmeras técnicas disponíveis para tratar esse problema, em sua maioria derivações ou combinações de técnicas de super-amostragem e sub-amostragem. Para os experimentos descritos no Capítulo 7, a técnica utilizada foi a de sub-amostragem (*undersampling*). Para ser mais específico, uma técnica de subamostragem com remoção aleatória de registros da população da classe predominante para balancear os dados. Essa técnica é simples de ser implementada, abstraindo toda a complexidade de geração de dados e preocupações inerentes a encontrar uma fronteira entre as classes. Além de não afetar os resultados em cenários onde a população da classe dominante não apresenta muita variação (população homogênea). A quantidade de registros de transações legítimas foi reduzida para equalizar com a quantidade de registros de transações fraudulentas na escala de 1 (fraude) para 100 (legítimas).

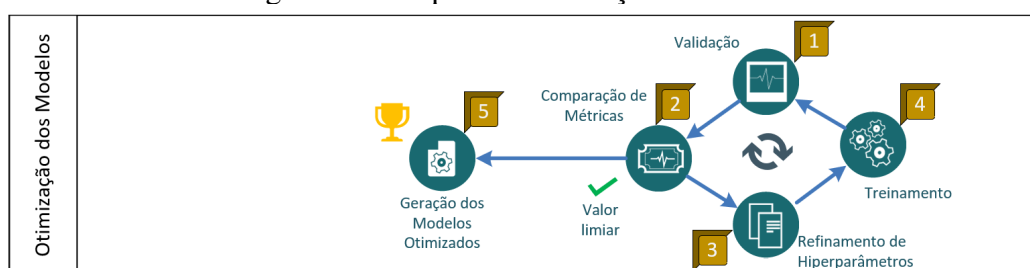
Na fase de “Geração dos Modelos” (Figura 6.4 (4)) os algoritmos escolhidos são alimentados com dados e parâmetros que permitam a criação de um modelo de referência para cada abordagem, permitindo comparar os seus resultados e analisar as suas métricas.

O produto dessa fase é um conjunto com um modelo para cada algoritmo, seus respectivos contextos (*hiperparâmetros e conjunto de dados de entrada*) e métricas obtidas.

6.5 Otimização dos Modelos

O objetivo dessa seção é detalhar as fases que fazem parte da etapa de “Otimização dos Modelos”, conforme ilustrada na Figura 6.5. As 5 fases que compõem a etapa de “Otimização de Modelos” são descritas a seguir.

Figura 6.5: Etapa 4 - Otimização dos Modelos



Fonte: Elaborado pelo autor

A etapa de “Otimização dos Modelos” tem início com um ciclo de 4 fases que se repetem por um número indefinido de vezes, até que o resultado alcançado pelo modelo em questão atinja um patamar definido ao longo dos experimentos, quando as métricas alcançam um valor limiar (*threshold*). O ponto de início desse ciclo pode variar sem afetar o seu resultado. Para o contexto deste trabalho, define-se a “Validação” (Figura 6.5 (1)) do modelo como ponto de entrada, considerando que um modelo de referência (*baseline*) acabou de ser gerado como produto da etapa de “Criação dos Modelos”.

A fase de “Comparação de Métricas” (Figura 6.5 (2)) tem como premissa calcular as métricas desejadas para o modelo em questão, compará-las com os resultados de outros modelos dos experimentos e verificar se o valor limiar proposto foi alcançado. O seguinte conjunto de métricas deve fazer parte dos resultados coletados:

6.5.1 Métricas Utilizadas para Avaliar os Resultados Obtidos

A matriz de confusão é a base tipicamente usada para medir o desempenho de qualquer classificador (SOKOLOVA; LAPALME, 2009). Especialmente efetiva para previsões baseadas em 2 classes. Este método é usado para avaliar o desempenho de um modelo pela comparação da decisão do classificador e rótulo da transação. Verdadeiro

Positivo (VP) representa uma transação fraudulenta corretamente predita pelo modelo, (FP) representa uma transação legítima predita como fraudulenta pelo modelo, (VN) representa uma transação legítima corretamente predita pelo modelo e (FN) representa uma transação fraudulenta predita como legítima pelo modelo.

Várias métricas são derivadas da matriz de confusão. A **acurácia**, por exemplo, é frequentemente apresentada como uma medida de desempenho, mas é sabido que não é uma métrica confiável quando o conjunto de dados é desequilibrado como nos conjuntos de dados de fraude o são (PROVOST; FAWCETT; KOHAVI, 1998). Uma boa alternativa à acurácia é utilizar a **acurácia balanceada**, que não é influenciada pelo desbalanceamento das classes, porque os cálculos ocorrem em cima da taxa de verdadeiros positivos e verdadeiros negativos. Chegando dessa forma a um valor mais correto na relação dos acertos do modelo pelas classes de predição.

A **precisão** das transações de fraude é o número de fraudes transações corretamente identificadas a partir do número total de transações identificadas transações fraudulentas. A taxa de falso-positivo é o número de falsos positivos, entre todas as transações genuínas.

A **pontuação F1** é uma métrica única que indica para quantas fraudes as transações foram classificadas corretamente e em quantos casos as fraudes passaram despercebidas. A pontuação F1 é especialmente tendenciosa quando há um grande desequilíbrio de classe e não proporciona uma métrica de comparação confiável para o contexto desse trabalho.

Uma medida melhorada é provida pelo **Coefficiente de Correlação Matthews** (MCC) (MATTHEWS, 1975), que é uma métrica única que pode ser utilizado em dados altamente desbalanceados, pois leva em conta todos os quadrantes da matriz de confusão. O MCC é um coeficiente de correlação entre a classe binária observada e a prevista com um valor que varia de -1 à +1. Um coeficiente positivo de +1 representa uma previsão perfeita, zero condiz com o desempenho de um classificador baseado em sorte (*50% de chance*) e menor que zero indica um desempenho pior do que um classificador baseado em 50% de chance. O MCC foi calculado para todos os experimentos realizados nesse capítulo.

Mesmo as métricas que apresentam uma distorção nos seus resultados como “**Acurácia**” e “**Pontuação F1**” foram mantidas na tabela de resultados final para permitir que o resultado dos experimentos aqui relatados possa ser comparado com outras publicações que utilizam essas métricas pela sua popularidade.

A lista abaixo relaciona as métricas utilizadas para avaliar os resultados dos experimentos ordenadas pelo seu grau de relevância:

- (1) Verdadeiro Negativo (*VN*).
- (2) Verdadeiro Positivo (*VP*).
- (3) Falso Negativo (*FN*).
- (4) Falso Positivo (*FP*).
- (5) Acurácia Balanceada.
- (6) Correlação de Matthews.
- (7) Precisão - PPV (*Positive Predictive Value*).
- (8) Sensitividade - TPR (*True Positive Rate*) - *Recall*.
- (9) Especificidade - TNR (*True Negative Rate*).
- (10) Acurácia.
- (11) Pontuação F1.

A fase de “Refinamento de Hiperparâmetros” (Figura 6.5 (3)) pode ser considerada uma das mais críticas da etapa de “Otimização de Modelos”, podendo consumir muito tempo em função da sua complexidade, principalmente se considerarmos que estamos trabalhando com múltiplos algoritmos e cada algoritmo possui o seu conjunto próprio de hiperparâmetros. Algumas técnicas, no entanto, se propõem a reduzir o esforço dessa atividade, nos experimentos descritos no Capítulo 7, foi utilizada uma técnica de *grid-search* através da qual centenas (ou até milhares) de experimentos são gerados variando valores de hiperparâmetros em um intervalo pré-definido e comparando os seus resultados para identificar uma combinação otimizada.

Na fase de “Treinamento” (Figura 6.5 (4)), um novo modelo é alimentado com base em um novo conjunto de hiperparâmetros, um novo contexto de dados ou ambos. Este treinamento pode também ser visto como um re-treinamento caso alguma característica do modelo anterior esteja sendo utilizada como ponto de partida para gerar o novo modelo.

Finalmente, na fase “Geração dos Modelos Otimizados” (Figura 6.5 (5)), define-se o nosso modelo alvo para cada algoritmo e exporta-se o modelo em um formato que nos permita integrá-lo ao fluxo de transações para realizar transações de forma *online*. O produto dessa fase é um conjunto com um modelo *otimizado* para cada algoritmo, seus respectivos contextos (*hiperparâmetros e conjunto de dados de entrada*) e métricas obtidas.

O pseudo código ilustrado no Algoritmo 1 apresenta uma visão lógica de como as iterações de treinamento são estruturadas para o modelo MLP. Este algoritmo descreve o passo a passo utilizado para treinar e testar o modelo com novos conjuntos de dados, passando por ciclos evolutivos, chamados de épocas. Digamos que, para um determinado problema, identificamos que precisaremos de dez épocas. Os dados de entrada passarão pela rede neural dez vezes e a cada vez, os pesos serão atualizados na direção do ideal, o mais próximo das características que queremos que a nossa rede neural aprenda. Para isso temos o número de épocas como um hiperparâmetro para configurar um limite para os ciclos de treinamento e teste. Ao final do processo as métricas do modelo gerado são calculadas e salvas, junto como o modelo e a lista de hiperparâmetros utilizados.

Algoritmo 1: LÓGICA DE TREINAMENTO PARA MODELOS BASEADOS EM REDES NEURAI

Entrada: *dados_de_treino, dados_de_teste_processados*

Saída: *modelo_treinado, hiperparametros_do_modelo, metricas_de_desempenho*

```

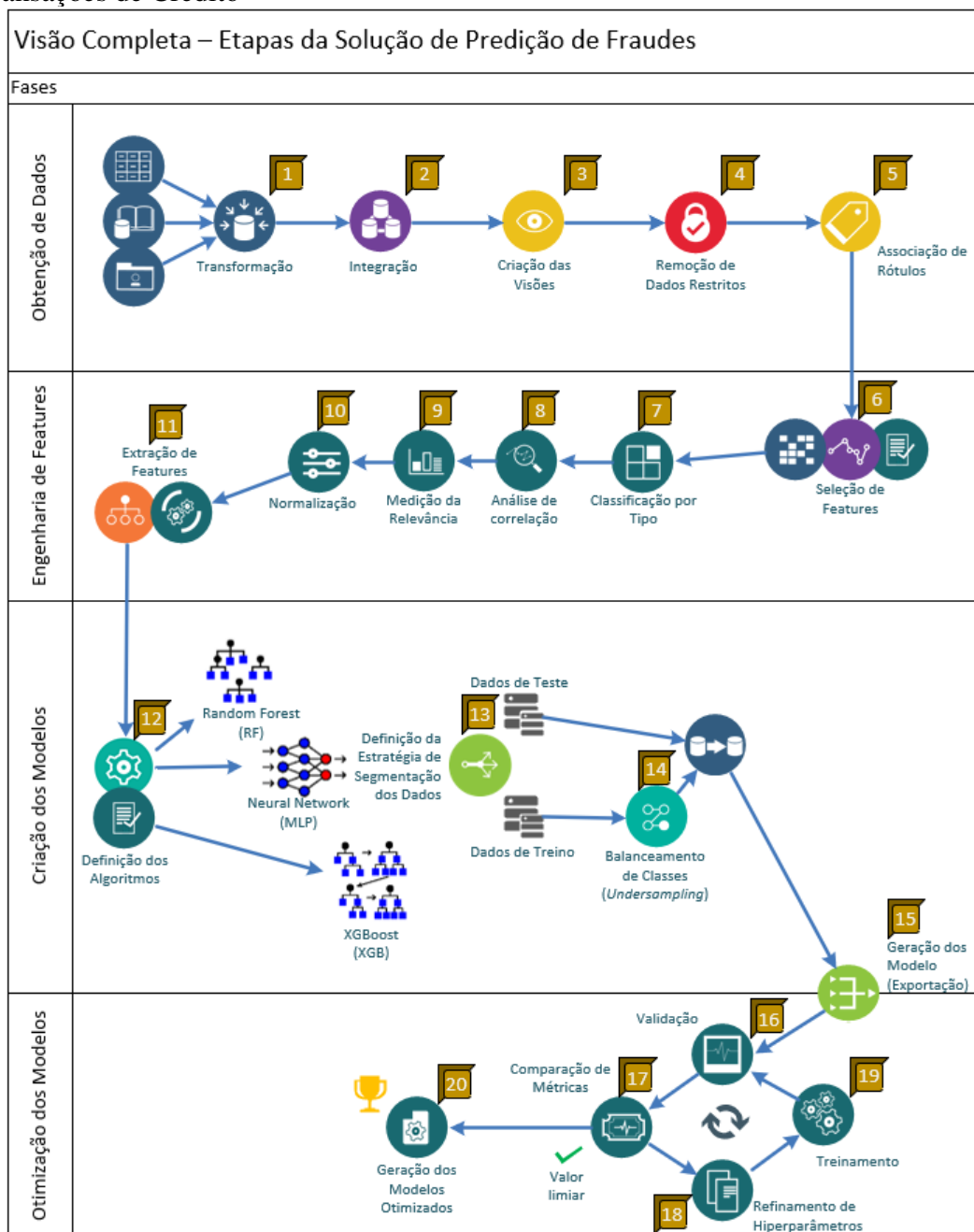
1 início
2   carregar dados_de_treino
3   carregar dados_de_teste
4   definir hiperparametros                                ▷ Arquivo de config
5   modelo ← parametros_iniciais                            ▷ Parametros internos do modelo
6   epoca ← 0
7   enquanto epoca ≤ epocas faça
8     | parametros_do_modelo ← atualizar_modelo(lote_de_dados)
9   fim
10  calcular metricas_de_desempenho
11  salvar modelo_treinado
12  salvar hiperparametros_do_modelo
13  salvar metricas_de_desempenho
14 fim

```

6.6 Fluxo Completo

O objetivo dessa seção é mostrar como as fases que integram as diferentes etapas da “Solução de Predição de Fraudes em Transações de Crédito” se conectam entre si, conforme ilustrado na Figura 6.6.

Figura 6.6: Visão Completa das Etapas da Solução Proposta para Predição de Fraudes em Transações de Crédito



Fonte: Elaborado pelo autor

Esta visão sumarizada das 4 etapas: “Obtenção de Dados”, “Engenharia de *Features*”, “Criação dos Modelos” e “Otimização dos Modelos” nos permite entender como

cada uma das 20 fases que compõem o fluxo como um todo se correlacionam, e interagem entre si.

7 IMPLEMENTAÇÃO DA SOLUÇÃO PROPOSTA

Neste capítulo é detalhada a implementação da proposta utilizada para conduzir os experimentos. As seções estão estruturadas conforme as etapas da proposta de solução, conectando o fluxo apresentado anteriormente aos experimentos realizados para validar a solução e analisar os resultados obtidos.

7.1 Obtenção de Dados

A etapa de “Obtenção dos Dados” diz respeito ao mapeamento das informações disponíveis sobre a transação no momento da decisão de sua autorização, esses dados são então classificados e documentados. Um dos grandes desafios da etapa de “Obtenção de Dados” foi avaliar o grande conjunto de atributos disponível no momento que o processo de autorização de uma transação de crédito ocorre por parte do **MPBR**.

Os dados de entrada para a etapa de “Obtenção de Dados” foram suportados por um dicionário de dados criado pelo time de negócios, descrevendo sucintamente cada fontes de dados, disponível, seu propósito e seus atributos.

Na Tabela 7.1 podemos ver uma sumarização do volume de atributos existentes, relacionados a cada uma das fontes de dados disponíveis. A primeira coluna na esquerda da tabela define o seu “Tipo” dividindo as fontes de dados em dois grandes grupos: fontes externas (*providas por parceiros/fornecedores*) e fontes internas disponibilizadas por sistemas integrados ao fluxo de transações. A segunda coluna chamada de “Grupo” descreve a fonte específica daquele conjunto de dados, seja o nome do fornecedor ou do sistema em questão. A terceira coluna define um “Subgrupo” daquela fonte de dados, utilizado para identificar quando os fornecedores ou sistemas internos possuem e fornecem dois ou mais conjuntos de dados distintos a respeito da transação. As três colunas seguintes são quantitativas e descrevem respectivamente:

- **Colunas:** número de campos disponíveis para aquela fonte de dados;
- **Seleção:** do conjunto inicial de colunas quantas foram selecionadas para gerar features;
- **Features:** quantas features foram derivadas a partir dos campos selecionados (*seleção*);

Os atributos relacionados a fontes externas correspondem ao maior volume do

conjunto (58%) com 714 campos, mas apenas 140 *features* foram derivadas desses campos, enquanto as *features* derivadas de campos de sistemas internos corresponderam a ampla maioria, com 354 *features* sendo efetivamente utilizadas nos modelos finais (72% de todas as *features*).

Tabela 7.1: Tabela com resumo dos conjuntos de dados avaliados como parte da etapa de obtenção de dados

Tipo	Grupo	Subgrupo	Colunas	Seleção	Features	
Externo	EmailAge		122	25	25	
	Experian	Crédito	248	0	0	
		NFD	16	10	10	
		PreciseID	189	73	73	
	Iovation		10	0	0	
	Lexis Nexis	Fone	27	0	0	
		Identificador	27	11	11	
	Neustar		54	8	8	
Produto		21	13	13		
Interno	Casos de Fraude	SFDC	158	0	0	
	SGCV (Conta)	Datawarehouse	17	13	13	
	ACTR (Transação)	Cliente		36	15	15
		Datawarehouse		37	11	11
		FISERV		59	32	32
		Iovation		10	7	69
		Discrepâncias		6	4	4
		SKU		38	20	65
		Calculados		6	3	105
				118	40	40
MDCU (Crédito)		38	0	0		
Total			1237	285	494	

7.1.1 Transformação

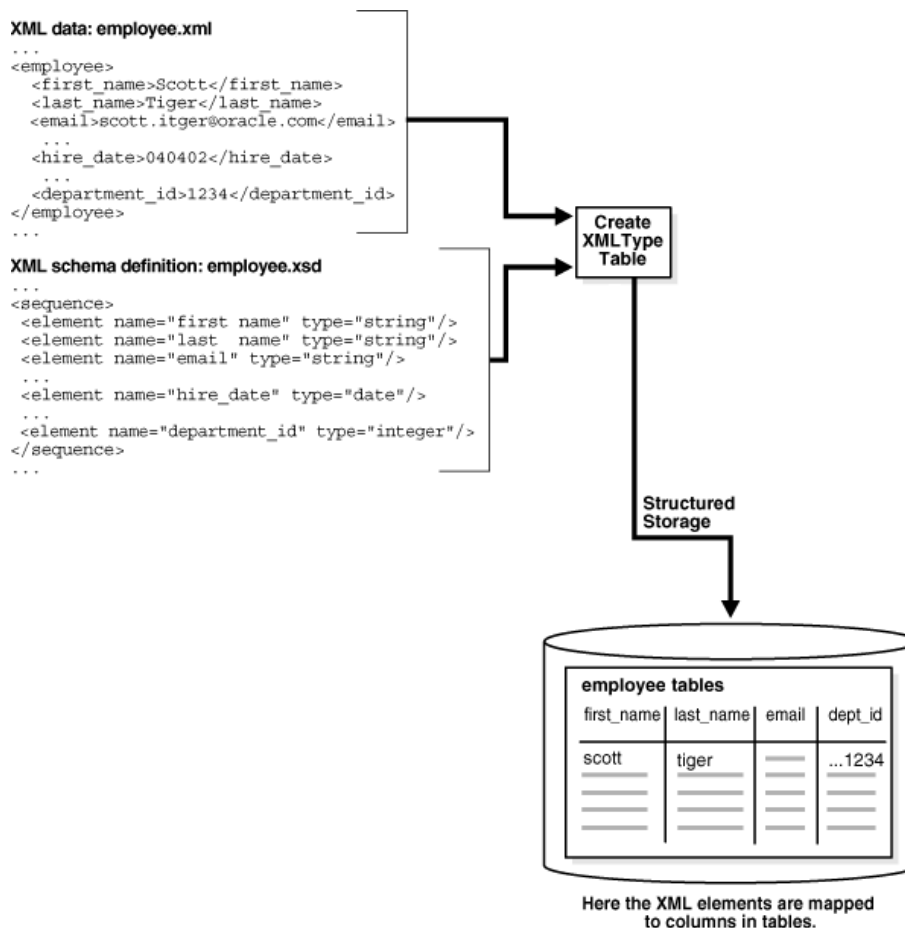
A fase de “Transformação” se caracteriza por consolidar as fontes de dados em um repositório centralizado, padronizando a sua formatação. A informação referente a uma transação processada pelo **MPBR** é nativamente armazenada no formato de um XML (*semi-estruturado*) que inclui todos os atributos enviados para que o sistema de regras de decisão possa gerar uma decisão.

Em função do volume de transações sendo exploradas, 1,75 milhões de registros representando 3 anos (2020, 2021 e 2022) de transações e aspectos relacionados a complexidade do *parsing* do XML e manutenibilidade desses dados para experimentos futuros,

a opção foi migrar esses dados para um banco de dados relacional. Em um primeiro momento Greenplum e em um segundo momento um banco de dados Oracle dedicado para os experimentos.

Para implementar essa migração de dados, foi criado um processo para exportar o conteúdo do XML e carregá-lo em colunas específicas das tabelas do banco de dados. Esse processo gerou múltiplas tabelas no banco com relações de cardinalidade variando conforme a estrutura do XML e os dados disponíveis em cada transação. Podemos ver uma representação simplificada desse processo para uma subseção do XML na Figura, 7.1.

Figura 7.1: Processo de Explosão de Atributos de XML em Tabelas de um banco relacional



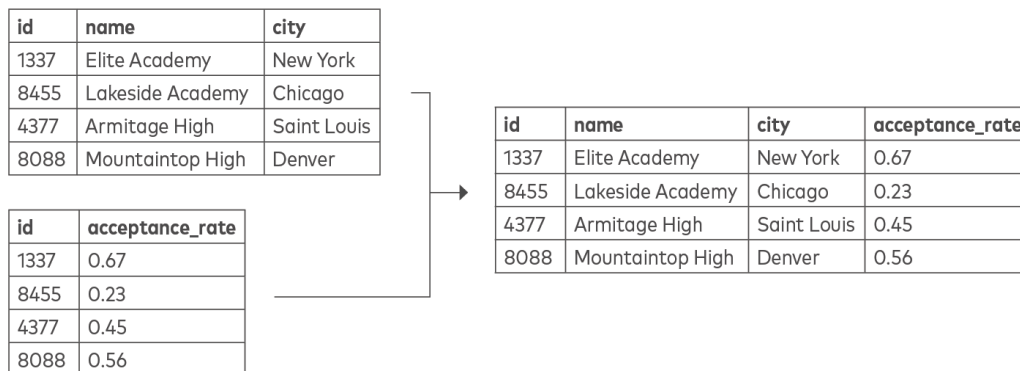
Fonte: Elaborado pelo autor

7.1.2 Integração

A fase de “Integração” unifica os dados do banco relacional em um número reduzido de tabelas, removendo tabelas de relacionamento e replicando os dados em um

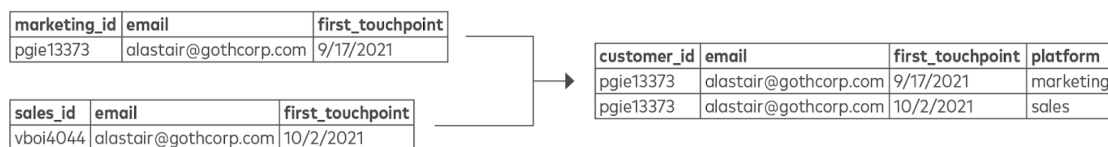
processo de *desnormalização*. Esse passo foi importante para reduzir a complexidade dos relacionamentos no banco de dados, encaminhando a criação de uma visão tabular. Algumas operações foram realizadas para combinar (*join*) e mesclar (*merge*) os dados, como representado respectivamente nas Figuras 7.2 e 7.3.

Figura 7.2: Exemplo de processo de combinação (*join*) de registros para reduzir a cardinalidade das relações



Fonte: Elaborado pelo autor

Figura 7.3: Exemplo de processo de mescla (*merge*) de tabelas para reduzir quantidade de objetos de banco necessários para representar os dados.



Fonte: Elaborado pelo autor

Por fim, campos de chave única ou relacional, identificadores, colunas vazias e colunas sem variação de valor foram descartadas, reduzindo a quantidade total de campos a serem considerados na etapa de “Engenharia de *Features*”.

7.1.3 Criação das Visões

A fase de “Criação das Visões” representa a sumarização dos dados para gerar uma visão tabular onde cada coluna representa um atributo da transação e cada linha representa uma transação única. Como resultado da fase de “Integração” os objetos do banco já haviam sido reduzidos para um número mínimo para representar as transações, o passo seguinte foi criar um processo ETL (*Extract Transform Load*) para traduzir os

dados aninhados em tabelas filhas ao registro da transação de crédito principal (exemplos de tabelas filhas: itens da transação, produtos comprados, endereços do cliente, histórico de compras anteriores do cliente).

Neste processo, foi utilizada uma combinação de técnicas de agrupamento, onde o valor carregado para o registro transacional variou de uma simples contagem do número de registros com uma dada característica, o valor máximo (*ou mais recente, no caso de datas*) ou ainda um somatório dos valores do atributo (*como, por exemplo, no caso dos valores dos itens da compra*). Esta atividade acabou se tornando um primeiro passo em direção a criação das *features* que seriam eventualmente utilizadas para alimentar os modelos.

A execução desse processo teve um custo computacional elevado (o tempo de processamento chegou a mais de 24 horas em um dado momento, devido ao volume de dados e colunas sumarizadas), em função do volume de dados sendo processados. Somente a tabela de itens das transações possuía mais de 6 milhões de registros, relacionados aos 1,75 milhões de transações disponibilizados para os experimentos. Por esse motivo, foi necessário segmentar o processamento da visão tabular em 5 outras tabelas. Eventualmente, o próprio conceito de uma *visão* tabular teve que ser ajustado para uma tabela persistida em banco para otimizar o desempenho durante o processamento dos dados.

O número final de campos na tabela que materializa a visão tabular é de 279 atributos únicos que incluem representações sumarizadas de seções como, histórico de compras anteriores, endereços do cliente, produtos comprados e seus componentes.

O produto final dessa fase foi a própria visão tabular, utilizada como entrada para alimentar os modelos escolhidos nas fases posteriores desse experimento e possibilitando que uma gama maior de algoritmos de aprendizado de máquina pudessem ser utilizados.

7.1.4 Remoção de Dados Restritos

Durante a fase de “Remoção de Dados Restritos” foram conduzidas atividades para identificação e mascaramento de dados com algum tipo de restrição de acesso. Em alguns casos, o mascaramento consistiu em apenas remover os dados, porém quando o campo em questão se mostrou necessário para alguma análise futura os dados foram transformados para que o valor original não pudesse ser recuperado.

No caso de informações pessoais sujeitas a análise de impacto a privacidade (do inglês: *PIA*) a abordagem foi simplesmente suprimir os dados, principalmente no contexto

de documentos como número de seguridade social (*SSN*), carteira de motorista e passaporte. Em função das leis referentes a concessão de crédito (*The Equal Credit Opportunity Act [ECOA], 15 U.S.C. 1691 [Estados Unidos]*) campos que possam estar amarrados a um contexto discriminatório também foram sumariamente descartados. Os campos que caíram nessa categoria foram: idade, língua preferida e país de origem.

Outros campos precisaram ser mantidos no conjunto de dados apesar de aspectos de confidencialidade. Um exemplo é a informação de endereço que se encaixa no conceito de *PIA*, mas teve os detalhes mascarados através da transformação em *features* que descrevam a latitude e longitude do *ZIP code* e um código Hash composto derivado dos textos de rua, número, CEP, cidade, estado e país. O motivo pelo qual a informação de endereço foi preservada diz respeito as diferentes informações de endereço que relacionadas a uma transação: endereço pessoal, endereço profissional, endereço de cobrança e endereço de entrega. Comparar variações do endereço se mostrou uma abordagem importante para detecção de fraudes no **MPBR**.

Outro campo que precisou de um tratamento especial, para lidar com a seu caráter confidencial, foi o número da conta de crédito do cliente, que nesse caso equivale a um número de cartão de crédito. Esse campo é peça fundamental do processo de associação dos rótulos, visto a seguir na fase de “Rotular Transações”. A abordagem utilizada para mascarar o código da conta do cliente foi a utilização de um algoritmo de criptografia com chave composta para impedir que o valor original pudesse ser recuperado.

7.1.5 Associação de Rótulos

A fase de “Associação de Rótulos” é a última da etapa de “Obtenção de Dados” e tem relação com o processo de identificação dos registros para diferenciar transações legítimas de fraudulentas.

O Algoritmo 2 descreve a lógica utilizada para mapear transações fraudulentas de forma automatizada. O bloqueio de uma conta de cliente pode estar relacionado a vários fatores, mas apenas dois códigos são relevantes para os rótulos de Fraude:

- **L** (*lock*): indica que a conta está sendo investigada por suspeita de fraude (estado transitório).
- **F** (*fraud*) : indica que a conta está bloqueada por confirmação de fraude (estado final).

A lógica do algoritmo de “Associação de Rótulos” estabelece que após 45 dias da data da transação, caso não exista nenhum registro de bloqueio por Fraude, a transação é considerada efetivamente legítima. Nesse meio tempo, a transação é considerada interinamente legítima. O bloqueio por fraude (*código ‘F’*) de uma conta faz com que a transação imediatamente anterior aquele evento de bloqueio, para aquela conta, seja marcada como fraudulenta. Se houverem outras transações relacionadas a conta de cliente em questão, elas serão descartadas do conjunto de dados de treino e teste, a partir deste momento. Bloqueios por fraude do tipo temporário (**L**) são apenas utilizados para descartar transações daquela conta de forma interina até que o bloqueio (**L**) evolua para (**F**) ou que a conta seja eventualmente desbloqueada.

Algoritmo 2: ROTULAR TRANSAÇÕES FRAUDULENTAS

Entrada: Conta C, Última Transação UT, Transações Anteriores TA

Saída: Rótulo R: (F) - fraude/ (L) - legítimo

```

1 início
2   se C = 'bloqueada por fraude (Tipo F)' então
3     UT.Rotulo ← 'F';
4     se Existe TA então
5       TA.Ignorear ← VERDADEIRO;
6     fim
7     R ← 'F';
8   fim
9   senão
10    se C = 'bloqueada por suspeita de fraude (Tipo L)' então
11      UT.Ignorear ← VERDADEIRO;
12    fim
13    se Existe TA então
14      TA.Ignorear ← VERDADEIRO;
15    fim
16    R ← 'L';
17  fim
18  retorna R
19 fim

```

É importante destacar que o sistema atual não armazena a informação sobre a confirmação da fraude (*ground-truth*) da transação. Sendo que a confirmação de fraude só é efetivamente disponibilizada no sistema responsável pela manutenção das contas de clientes e pode levar semanas para acontecer.

7.2 Engenharia de *Features*

Esta seção descreve como as atividades de “Engenharia de *Features*” foram desenvolvidas para implementar a proposta a partir de técnicas específicas de seleção e extração de *features*. Foi durante essa etapa que o maior número de campos da visão tabular foram descartados, abrindo espaço para a extração de um conjunto significativo de novas *features* derivadas dos atributos apontados como mais relevantes para predições dos modelos iniciais.

Ao longo dessa etapa, uma decisão foi firmada por manter *features* não tão relevantes no conjunto final foi tomada. Visando assim criar modelos mais robustos e capazes de serem retreinados para identificar novos padrões de comportamento fraudulento.

7.2.1 Seleção de *Features*

O objetivo dessa seção é mostrar como foi implementada a “Seleção de *Features*” no desenvolvimento de um estudo de caso da proposta e execução dos experimentos. Considerando a quantidade de dados disponíveis para a tarefa de predição originalmente, conduzir um processo de seleção de *features* efetivo se tornou crucial, principalmente na identificação de dados mais relevantes para a predição. A seguinte listagem apresenta as técnicas de “Seleção de *Features*” adotadas durante essa fase:

- Técnica de relevância de *features* (modelo baseado em árvore): principais *features* são identificadas por seu valor limiar.
- Correlação com a variável alvo: identificação das *features* com maior correlação estabelecendo um valor limiar de correlação.
- Informação mútua com a variável alvo: identificação das *features* com maior informação mútua.
- ANOVA - análise de variância.
- *L1 regularization*: regressão Lasso (também conhecido como Valor Absoluto de Magnitude) calculado por meio de um modelo SVM.
- Eliminação recursiva de *features*: baseada no respectivo modelo **XGB** ou **RF**.

A prioridade inicial das atividades de “Seleção de *Features*” é mapear entradas que não adicionem valor ao modelo, essa ausência de relevância pode ser causada por

diversos fatores, a análise mais básica diz respeito ao conteúdo de um campo que pode ser completamente vazio, ser constante apresentando sempre o mesmo valor ou repetir os mesmos valores de outra *feature*. Muito embora parte dessa análise tenha começado ainda durante a etapa de “Obtenção de Dados”, principalmente para colunas de tabelas vazias.

A remoção mais relevante de *features* do conjunto de dados por esse critério aconteceu durante a fase de seleção, quando ao todo, 12 *features* foram eliminadas por não apresentar uma grande variação de valores, conforme listado abaixo:

- Lista negativa de CEP do endereço de cobrança - Critério: vazio
- Lista negativa do telefone principal do cliente - Critério: vazio
- Produto da categoria 4 (*PDA, portátil*) - Critério: repetido
- Memória flash do dispositivo habilitada [IOV] - Critério: repetido
- Linha de negócio do componente igual à nuvem (*cloud*) - Critério: constante
- Identificador do endereço de cobrança do item - Critério: constante
- Correspondência na lista registros de vítimas [EXPR] - Critério: constante
- Correspondência no campo de vítimas [EXPR] - Critério: constante
- Contagem de devoluções do endereço de negócios [EXPR] - Critério: constante
- Contagem de ocorrências do telefone de negócios [EXPR] - Critério: constante
- Contagem de devoluções do telefone de negócios [EXPR] - Critério: constante
- Pontuação FPDS (Sistema de Dados de Compra Federal) [EXPR] - Critério: constante

Após a eliminação de um primeiro conjunto de *features*, técnicas de análise de variância e relevância foram utilizadas para detectar ruído, viés, correlação com a variável alvo, posteriormente identificada na forma de campos do conjunto de dados atualizadas em função de uma decisão do **MPBR**. Ao todo, 26 *features* e 5 tabelas inteiras foram removidas do conjunto de dados durante esse processo.

Para a fase seguinte de análise de correlação segregar as *features* por tipo e em alguns casos normalizar os seus valores para otimizar os resultados. As técnicas de análise de correlação adotadas foram o coeficiente de Pearson e o ranking de Spearman, ambas apresentaram resultados similares. Para melhorar a resolução e legibilidade das imagens, o resultado da análise foi dividido em duas Figuras 7.4 e 7.5 onde podemos ver uma simplificação da análise de Pearson para as *features* impactadas por problemas de correlação. A seguinte informação deve ser considerada para analisar o índice de correlação segundo o código de cores para esses 2 gráficos (*OBS: para a análise em questão os valores vari-*

aram de -0,197 à 1).

- Índice de correlação menor que zero - Ação: manter *features*.
- Índice de correlação entre 0 e 0,2 - Ação: manter *features*;
- Índice de correlação entre 0,2 e 0,8 - Ação: manter *features*;
- Índice de correlação acima de 0,8 - Ação: eliminar uma das *features*;

Para a análise de correlação de Pearson tanto valores próximos a +1 quanto -1 são elegíveis a serem removidos. Valores próximos a -1 indicam alta correlação inversa. Não houve nenhum caso de alta correlação inversa para a análise em questão.

As imagens das Figuras 7.4 e 7.5 representam apenas o subconjunto das *features* que apresentaram alta correlação e seus respectivos pares. A análise original de correlação incluiu todas as *features* (mais de 500 linhas e colunas) em uma única imagem, infelizmente a imagem original ficaria ilegível nessa dimensão de página. Optamos então por representar apenas essa amostra.

Figura 7.4: Análise de correlação de *features* - Ilustração do coeficiente de Pearson para pares de *features* do conjunto de dados - primeiro conjunto de *features*

	summ_prd_handheld_thr	summ_orderamount	order_totalfinancedamount	iovt_ipaddress_loc_lng	order_calculatedtotalfinancedamount	skus_itemdesc_notebooks	emagr_ipdistancemil	summ_prd_handheld_thr_tamnt	skus_lineofbusiness_monitors	iovt_realipaddress_loc_lng	iovt_ipaddress_loc_lat	fiser_days_delinquent
summ_prd_handheld_thr	1.000	-0.005	-0.005	0.000	-0.005	-0.007	-0.001	0.988	-0.003	0.000	0.001	0.004
summ_orderamount	-0.005	1.000	1.000	-0.036	1.000	0.325	0.026	-0.005	-0.019	-0.036	-0.012	-0.017
order_totalfinancedamount	-0.005	1.000	1.000	-0.036	1.000	0.325	0.026	-0.005	-0.019	-0.036	-0.012	-0.017
iovt_ipaddress_loc_lng	0.000	-0.036	-0.036	1.000	-0.036	0.018	0.116	0.000	-0.019	0.992	0.019	0.002
order_calculatedtotalfinancedamount	-0.005	1.000	1.000	-0.036	1.000	0.325	0.026	-0.005	-0.019	-0.036	-0.012	-0.017
skus_itemdesc_notebooks	-0.007	0.325	0.325	0.018	0.325	1.000	0.021	-0.007	-0.197	0.018	-0.004	-0.013
emagr_ipdistancemil	-0.001	0.026	0.026	0.116	0.026	0.021	1.000	-0.001	-0.006	0.111	-0.046	0.000
summ_prd_handheld_thr_tamnt	0.988	-0.005	-0.005	0.000	-0.005	-0.007	-0.001	1.000	-0.003	0.000	0.001	0.003
skus_lineofbusiness_monitors	-0.003	-0.019	-0.019	-0.019	-0.019	-0.197	-0.006	-0.003	1.000	-0.019	-0.004	0.002
iovt_realipaddress_loc_lng	0.000	-0.036	-0.036	0.992	-0.036	0.018	0.111	0.000	-0.019	1.000	0.019	0.002
iovt_ipaddress_loc_lat	0.001	-0.012	-0.012	0.019	-0.012	-0.004	-0.046	0.001	-0.004	0.019	1.000	0.001
fiser_days_delinquent	0.004	-0.017	-0.017	0.002	-0.017	-0.013	0.000	0.003	0.002	0.002	0.001	1.000

Fonte: Elaborado pelo autor

O resultado da análise de correlação nos permitiu eliminar 17 *features* do conjunto de dados de entrada por possuírem um grau de similaridade acima de 90% com uma ou mais *features* disponíveis. A seguinte listagem mostra o conjunto de *features* removidas por análise de correlação:

- Correlação de distância entre IP e endereço em milhas [EmAge]
- Probabilidade de ser uma requisição de um robô (bot) baseado no IP [IOV]
- Latitude do endereço IP real [IOV]
- Longitude do endereço IP real [IOV]

Figura 7.5: Análise de correlação de *features* - Ilustração do coeficiente de Pearson para pares de *features* do conjunto de dados - segundo conjunto de *features*

	iovt_realipaddress_loc_lat	summ_prd_notebk_xps	summ_prd_projctr_tamt	skus_lineofbusiness_desktops	iovt_realipaddress_botnet_score	summ_prd_projctr	skus_itemdesc_desktops	iovt_ipaddress_botnet_score	order_totalamount	summ_prd_notebk_xps_tamt
iovt_realipaddress_loc_lat	1.000	-0.005	0.000	-0.004	-0.031	0.000	-0.003	-0.025	-0.011	-0.003
summ_prd_notebk_xps	-0.005	1.000	-0.001	-0.118	-0.018	-0.001	-0.127	-0.033	0.294	0.918
summ_prd_projctr_tamt	0.000	-0.001	1.000	-0.001	0.000	0.000	0.911	-0.001	0.006	-0.001
skus_lineofbusiness_desktops	-0.004	-0.118	-0.001	1.000	0.025	-0.001	0.929	0.041	0.401	-0.110
iovt_realipaddress_botnet_score	-0.031	-0.018	0.000	0.025	1.000	0.000	0.015	0.996	-0.009	-0.031
summ_prd_projctr	0.000	-0.001	0.911	-0.001	0.000	1.000	-0.001	0.000	0.005	-0.001
skus_itemdesc_desktops	-0.003	-0.127	-0.001	0.929	0.015	-0.001	1.000	0.040	0.392	-0.118
iovt_ipaddress_botnet_score	-0.025	-0.033	0.000	0.041	0.996	0.000	0.040	1.000	0.000	-0.042
order_totalamount	-0.011	0.294	0.006	0.401	-0.009	0.005	0.392	0.000	1.000	0.348
summ_prd_notebk_xps_tamt	-0.003	0.918	-0.001	-0.110	-0.031	-0.001	-0.118	-0.042	0.348	1.000

Fonte: Elaborado pelo autor

- Identificador do endereço de entrega do item
- Valor financiado total da transação
- Linha de negócios do componente igual a desktop
- Linha de negócios do componente igual a monitor
- Variável calculada - dias de atraso no pagamento da fatura
- Variável calculada - distância entre os endereços de entrega e cobrança do cliente
- Variável calculada - idade do dispositivo de onde a transação foi iniciada [IOV]
- Variável calculada - valor total da transação
- Variável calculada - valor total de itens com componente dispositivo móvel ou PDA
- Variável calculada - quantidade de itens com componente XPS
- Variável calculada - valor total de itens com componente XPS
- Variável calculada - quantidade de itens com componente projetor
- Variável calculada - valor total de itens com componente projetor

Muito embora um número considerável de *features* tenha sido descartado durante a fase de seleção para evitar que essas impactassem o resultado da predição, é importante destacar que um número maior de *features* poderia ser desconsiderado se não houvesse uma prerrogativa de manter um conjunto de dados substancial para análise de variações de padrão de fraudes no futuro.

7.2.2 Extração de *Features*

Esta seção visa descrever como a “Extração de *Features*” foi implementada a partir da proposta de solução de predição de fraudes, para a realização dos experimentos. A seguinte listagem apresenta as técnicas de “Extração de *Features*” adotadas durante essa fase:

- Tratamento de valores discrepantes (anomalias) : IQR e IQR desconsiderando o valor da modal.
- Correção de assimetria: transformação logarítmica e potência quadrada.
- Codificação de variáveis categóricas: técnica “Onehot” e “Top to Onehot” para derivar novas *features*.
- Geração de *features* baseada na diferença de datas. (*Por exemplo: tempo de vida de uma conta de cliente*).
- Normalização: transformação da distribuição para ajustar a média para zero e desvio padrão igual a um.
- Composição de novas *features* a partir da fração de *features* existentes normalizadas.
- Geração de *features* a partir da contabilização do número de correspondências de um dado valor.

Ao longo das etapas de obtenção de dados, engenharia de *features* e criação dos modelos 27 novas *features* foram propostas e mantidas para o modelo final. Após a análise de relevância das *features* no contexto de predição, 5 das *features* geradas como parte deste trabalho se destacaram entre as 20 mais importantes, a Tabela 7.2 lista essas *features* e descreve o seu significado.

Tabela 7.2: Descrição das *features* mais relevantes criadas pelo processo de extração.

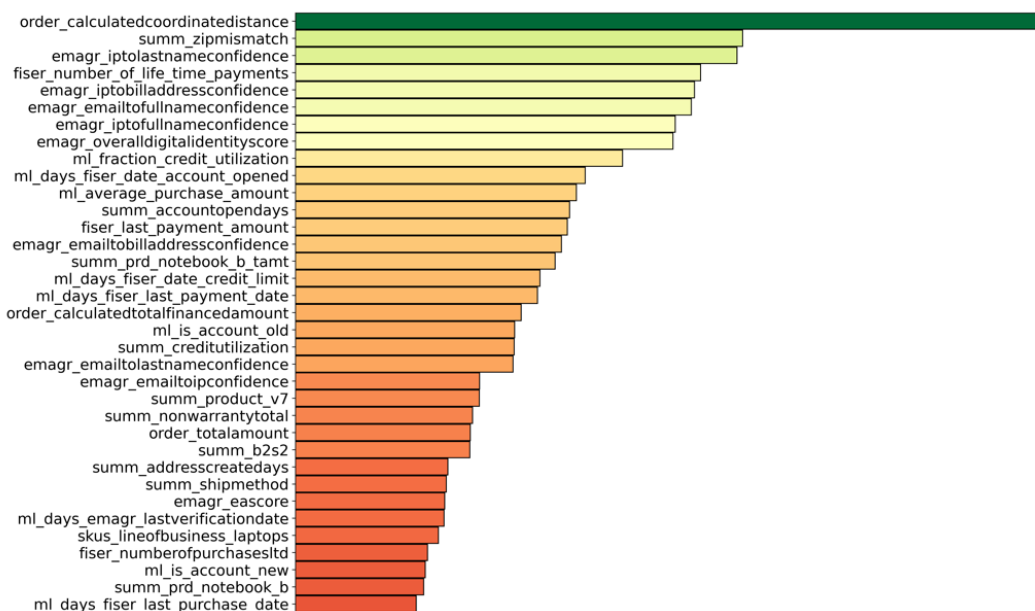
Posição	Nome	Documentação
8	Fração do crédito utilizado	Percentual do crédito total atualmente usado pelo cliente.
9	Dias desde a abertura da conta [FSV]	Quantidade de dias que se passaram desde que a conta de crédito foi aberta.
10	Perfil de gastos	Valor médio das compras desse cliente.
16	Dias desde a atualização de limite [FSV]	Número de dias desde que o valor do limite de crédito do cliente mudou.
17	Dias desde o último pagamento de fatura	Número de dias desde que o cliente efetuou seu último pagamento.

7.2.3 Features Mais Relevantes

Uma técnica aplicada de forma recorrente durante vários momentos da etapa de “Engenharia de *Features*”, foi a análise de relevância baseada no modelo RF (*Random Forest*). Essa técnica nos permite comparar a contribuição que as diversas *features* estão trazendo para o modelo e ordena-las em uma lista de prioridade.

Na Figura 7.6 podemos ver a lista priorizada relativa ao modelo mais atual gerado em janeiro de 2023 que possui 273 *features*. Na Tabela 7.3 temos uma pequena descrição para as 20 *features* mais relevantes dessa lista, é interessante descartar que 5 dessas 20 *features* foram geradas pela modelagem conduzida durante esse estudo.

Figura 7.6: Lista de *features* mais relevantes para decisões do modelo RF pela métrica: média decrescente de impureza (*mean decrease in impurity*).



Fonte: Elaborado pelo autor

Entre as fontes de dados originais que se destacam na listagem de *features* mais relevantes descritas na Tabela 7.3 se destacam pela quantidade de *features* derivadas as seguintes fontes:

(1) **EmailAge** (fonte *Externa*) com 7 *features* ao todo. A EmailAge é um desenvolvedor de tecnologia de prevenção contra fraude e verificação de identidade utilizada para tornar as transações mais fáceis e seguras. A empresa fornece soluções preditivas multifacetadas com e-mail no centro através da combinação de chaves, dados de propriedade e tecnologia de aprendizagem por máquina. A EmailAge possui algoritmos proprietários

Tabela 7.3: Descrição das *features* que tiveram maior relevância para decisões.

Pos	Nome	Documentação
1	Distancia Calculada Entre Coordenadas [ACTR Cliente]	Valor calculado da distância entre o endereço de entrega e o endereço de cobrança (Valores especiais: 0 se iguais, -1 se alguma coordenada não estiver disponível).
2	Discrepância de CEP [ACTR Calculado]	Valor booleano que indica se o código postal de cobrança e entrega são iguais (Verdadeiro) ou diferentes (Falso).
3	Confiança Correlação IP e Sobrenome [EmAge]	Nível de confiança calculado pelo provedor para combinação de IP e sobrenome do cliente.
4	Número de Pagamentos [ACTR - FiSv]	Quantidade de pagamentos efetuados para uma conta de cliente desde a sua abertura.
5	Confiança Correlação IP e Endereço Cobrança [EmAge]	Nível de confiança calculado pelo provedor para essa combinação de IP e endereço de cobrança.
6	Confiança Correlação Email e Nome Completo [EmAge]	Nível de confiança calculado pelo provedor para essa combinação de endereço de email e nome completo do cliente.
7	Confiança Correlação IP e Nome Completo [EmAge]	Nível de confiança calculado pelo provedor para essa combinação de IP e nome completo.
8	Fração da Utilização do Crédito [ML - FiSv]	Percentual do crédito total atualmente usado pelo cliente.
9	Dias Desde Abertura Conta [ML - FiSv]	Quantidade de dias que se passaram desde que a conta de crédito foi aberta.
10	Valor Médio Compras [ML - ACTR]	Valor médio das compras desse cliente.
11	Pontuação da Identidade Digital [EmAge]	Pontuação de risco da identidade digital
12	Diferença em Dias entre Datas de Abertura Conta e Transação [ACTR - Calculados]	Diferença em dias desde que a conta de crédito do cliente foi registrada no FISERV e a data dessa compra.
13	Valor do Último Pagamento [ACTR - FiSv]	Valor do último pagamento feito pelo cliente (fatura).
14	Confiança Correlação Email e Endereço de Cobrança [EmAge]	Nível de confiança calculado pelo provedor para essa combinação de endereço de email e endereço de cobrança.
15	Valor Total dos Itens com Componente "NOTEBOOK" [ACTR - Calculado]	Valor total da compra relacionados a produtos de linha de notebooks.
16	Dias Atualização Limite Crédito [ML - FiSv]	Número de dias desde que o valor do limite de crédito do cliente mudou.
17	Dias Último Pagamento [ML - FiSv]	Número de dias desde que o cliente efetuou seu último pagamento.
18	Valor Financiado Total da Transação [ACTR]	Valor total da compra efetivamente financiado usando o crédito do cliente.
19	Utilização do Crédito [ACTR - Calculado]	Valor percentual do limite de crédito atualmente utilizado pelo cliente.
20	Confiança Correlação Email e Sobrenome [EmAge]	Nível de confiança calculado pelo provedor para essa combinação de endereço de email e sobrenome do cliente.

a fim de fornecer alerta em tempo real de transações de risco e fornece uma pontuação de risco e permite que as empresas realizem economias significativas na identificação e interrupção de transações fraudulentas e melhorem a experiência do cliente.

(2) **Fiserv** (fonte *Interna*) com 6 *features* ao todo. A Fiserv é uma multinacional Americana que disponibiliza serviços de tecnologia na área de financiamento para clientes nos setores de serviços financeiros, incluindo: bancos, poupanças, uniões de crédito, corretores de seguros, hipotecas, seguradoras, financeiras, empresas especializadas em arrendamento (leasing) e varejistas. A Fiserv possui um conjunto completo de sistema para gestão de carteira de crédito de clientes, possibilitando a terceirização completa desses serviços.

(3) **Variáveis Calculadas** (fonte *Interna*) com 4 *features* ao todo. As variáveis calculadas são os campos pré-processados pelo **MPBR**. Como representam a única fonte utilizada pelo sistema atual para tomar decisões, a sua relevância para os modelos de aprendizado de máquina já era esperado. Estes campos foram detalhados no Capítulo 5 (*baseline*).

(4) **ACTR** (fonte *Interna*) com 3 *features* ao todo. O sistema ACTR é responsável por processar as requisições de autorização de transações de crédito, coletando dados de todas as fontes necessárias para enviar ao **MPBR**. No caso das *features* com alta relevância, a fonte é o próprio sistema de vendas que repassa as informações-base da transação para que o processo de autorização tenha início.

Para finalizar essa seção apresentamos a Tabela 7.4 que descreve as mudanças que ocorreram entre um modelo e outro e a quantidade de *features* utilizadas em cada versão dos modelos.

7.3 Criação dos Modelos

O objetivo dessa seção é mostrar como aconteceu a concepção dos modelos de aprendizado de máquina, para implementação da proposta. Desde a escolha dos algoritmos, passando pelos protótipos iniciais até evoluir para modelos refinados, passando por atividades específicas de otimização de hiperparâmetros.

Tabela 7.4: Quantidade de *Features* Utilizadas no Treinamento de Diferentes Versões dos Modelos.

Data	Versão	<i>Features</i>	Acurácia
Jun/22	RF 1.0 Modelo inicial <i>features</i> numéricas e subconjunto das categóricas)	212	0,5
Jul/22	RF 2.0 Primeiro experimento com treinamento balanceado	216 212	0,93
Ago/22	RF 3.0 Migração de repositório (Oracle) + <i>features</i> visão tabular	618	0,95
Ago/22	RF 4.0 XGB 1.0 Remoção de <i>features</i> com informação da variável alvo	316 316	0,87 0,97
Ago/22	MLP 1.0 Novo conjunto de <i>features</i> extraídos	486	0,84
Ago/22	RF 5.0 XGB 2.0 MLP 2.0 Número de <i>features</i> variando entre experimentos Processo de explorar novas fontes de dados foi interrompido	574 372 414	0,90 0,97 0,84
Ago/22	RF 6.0 XGB 3.0 MLP 3.0 Experimento com um <i>baseline</i> de <i>features</i> para os 3 modelos	369 369 369	0,97 0,97 0,96
Set/22	RF 7.0 XGB 4.0 MLP 4.0 <i>Feature</i> com problemas de conformidade removida	368 368 368	0,97 0,97 0,89
Out/22	RF 8.0 XGB 5.0 MLP 5.0 Último conjunto de <i>features</i> categóricas adicionadas	467 467 467	0,82 0,87 0,86
Nov/22	RF 9.0 XGB 6.0 MLP 6.0 Otimização de hiperparâmetros	429 429 429	0,89 0,86 0,90
Dez/22	MLP 7.0 MLP definido como modelo para produção	438	0,86
Jan/23	MLP 8.0 Simplificação para integrar API em produção	237	0,87

7.3.1 Definição dos Algoritmos

A metodologia para escolha dos modelos a serem utilizados na implementação da proposta considerou os seguintes fatores: (1) existência de dados rotulados, (2) quantidade e tipos de dados disponíveis para alimentar os modelos, (3) aderência dos algoritmos ao problema de predição em questão e (4) a flexibilidade desses modelos para serem retreinados e expandidos para um novo conjunto de *features*. Baseado nesses aspectos os seguintes algoritmos foram selecionados para geração de modelos e comparação dos seus resultados: *Random Forest* - **RF**, *Extreme Gradient Boost* - **XGB** e *Multilayer Perceptron* - **MLP**.

Após a definição dos algoritmos a serem adotados, ciclos iterativos foram conduzidos para abordar problemas como o desbalanceamento das classes de predição e opções foram exploradas para otimizar a segmentação do conjunto de dados para treino e testes do modelo.

7.3.2 Modelos Iniciais

O principal produto dessa etapa são os modelos iniciais gerados com cada um dos algoritmos, chamados em um primeiro momento de *baseline* para futuras comparações e avaliações dos modelos refinados.

A Tabela 7.5 nos mostra as métricas dos modelos iniciais gerados com o algoritmo *Random Forest*, o primeiro método utilizado para gerar modelos. Como podemos ver em sua primeira versão, a acurácia era de 50% em função do quão imaturo estava o processamento das *features* utilizadas. Ao longo dos dois primeiros meses apenas modelos **RF** foram gerados, até que o processo de pré-processamento das *features* estivesse mais estável. Podemos ver como as métricas variam em uma amplitude muito grande entre esses 5 modelos, demonstrando descobertas referentes a *features* geradas a partir da decisão do MPBR e outras que apenas adicionavam ruído a predição.

Durante o mês de agosto resultados mais estáveis foram obtidos, sendo iniciados experimentos com outros algoritmos, começando com *XGBoost*, como representado na tabela 7.6. Os experimentos iniciais utilizando **XGB** apresentaram resultados melhores em função do processo de engenharia de *features* estar mais avançado e existir um conjunto de *features* maior pronto para alimentar o modelo. De fato, os resultados obtidos nesses primeiros experimentos com **XGB** ainda incluíam dados que inferiam o valor da

Tabela 7.5: Modelos iniciais gerados utilizando Random Forest (RF).

Métricas ML	Random Forest				
	RF 1.0	RF 2.0	RF 3.0	RF 4.0	RF 5.0
Acurácia	0,5	0,933	0,946	0,87	0,904
Precisão	0	0,926	0,019	0,821	0,018
Sensitividade	0	0,902	0,983	0,741	0,896
Especificidade	1	N/D	0,908	0,999	0,911
Pontuação F1	0	0,914	0,038	0,779	0,036
TN	N/D	N/D	N/D	296818	259624
FP	N/D	N/D	N/D	374	25339
FN	N/D	N/D	N/D	601	55
TP	N/D	N/D	N/D	1717	474
<i>Features</i>	212	216	618	316	574
Hiperparâm.	N	N	N	N	N
Treino B.	F	F	F	V	V
Teste B.	F	F	F	F	F
<i>Undersampling</i>	F	F	F	V	V
Período	Jun/22	Jul/22	Ago/22	Ago/22	Ago/22

variável alvo que tiveram de ser suprimidos posteriormente por não existirem no momento da predição.

Finalmente, ainda em agosto, tiveram início os experimentos com o algoritmo *Multilayer Perceptron*, os resultados iniciais desses experimentos são descritos na tabela 7.7. Podemos ver uma maior consistência nos resultados obtidos durante essas duas rodadas de experimentos, a quantidade de *features* testadas nesse primeiro momento para gerar modelos **MLP** era maior, pois o processo de engenharia de *features* estava em seu auge.

Muito embora a quantidade de *features* fosse maior os resultados obtidos não foram tão expressivos quanto com o modelo **XGB**. Dois fatores foram chave para que o desempenho não fosse tão bom, nesse primeiro momento, (1) o ruído causado por *features* recentemente adicionadas ao conjunto de dados para o qual o **MLP** pareceu ser mais suscetível e (2) a complexidade de parametrização do algoritmo que demandou mais tempo.

7.4 Otimização dos Modelos

A etapa de “Otimização dos Modelos” foi o momento em que o maior número de experimentos ocorreram, pegando como ponto de partida os resultados obtidos com

Tabela 7.6: Modelos iniciais gerados utilizando XGBoost (XGB).

Métricas ML	XGBoost	
	XGB 1.0	XGB 2.0
Acurácia	0,965	0,973
Precisão	0,304	0,309
Sensitividade	0,947	0,964
Especificidade	0,983	0,983
Pontuação F1	0,461	0,468
TN	292169	291975
FP	5023	4989
FN	122	84
TP	2196	2234
<i>Features</i>	316	372
Hiperparâm.	N	N
Treino B.	V	V
Teste B.	F	F
Undersampling	V	V
Período	Ago/22	Ago/22

Tabela 7.7: Modelos iniciais gerados utilizando Multilayer Perceptron (MLP).

Métricas ML	Multilayer Perceptron	
	MLP 1.0	MLP 2.0
Acurácia	0.838	0.839
Precisão	0.044	0.101
Sensitividade	0.841	0.748
Especificidade	0.836	0.931
Pontuação F1	0.084	0.178
TN	251119	237118
FP	49427	17693
FN	435	673
TP	2301	1993
<i>Features</i>	486	414
Hyperparâm.	N	N
Treino B.	V	V
Teste B.	F	F
<i>Undersampling</i>	V	V
Período	Ago/22	Ago/22

os modelos iniciais de cada algoritmo, mudanças foram propostas para otimizar o seu desempenho. Estas ações consistiram da remoção de *features* que estivessem adicionando ruído, adição de novas *features* derivadas de dados promissores segundo a análise de relevância e a otimização de hiperparâmetros, que até então não haviam sido tocados.

Une-se a isso demandas de remoção de *features* por questões de conformidade de negócio e disponibilidade em tempo de predição, que também afetaram os resultados ao longo do tempo, como veremos no decorrer deste capítulo, começando pela otimização dos hiperparâmetros.

7.4.1 Abordagem de Otimização de Hiperparâmetros

A abordagem utilizada nessa etapa do processo para otimizar os hiperparâmetros e maximizar os resultados se baseou em um *Grid Search* através do qual cerca de 2000 modelos foram treinados e comparados entre si, variando os valores dos hiperparâmetros linearmente para encontrar combinações onde os resultados para uma ou mais métricas fossem melhoradas.

Cada algoritmo possui seu próprio conjunto de hiperparâmetros, inerentes ao funcionamento do método e os fatores que podem influenciar o seu desempenho. Por esse motivo o processo de otimização dos hiperparâmetros aconteceu individualmente para cada modelo, muito embora, para cenários de experimentos correlatos, os resultados obtidos foram comparados entre os modelos.

7.4.2 Hiperparâmetros *Random Forest* - RF

Nesta seção serão apresentados os hiperparâmetros ajustados e otimizados no contexto do algoritmo *Random Forest*. A Tabela 7.8 mostra o resultado do processo de otimização de hiperparâmetros para o algoritmo *Random Forest*. Nessa tabela os hiperparâmetros modificados ao longo dos experimentos estão listados em uma ordenação decrescente pela sua importância individual relativa para a melhoria da acurácia balanceada ao longo dos testes.

Este valor de importância foi calculado pela soma de (1) e (2), (1) sendo o resultado da subtração da acurácia balanceada mais alta obtida com o valor otimizado pela acurácia balanceada mais alta obtida com os demais parâmetros e (2) o resultado da sub-

tração da acurácia balanceada mais baixa obtida com o valor otimizado pela acurácia balanceada mais baixa obtida com os demais parâmetros. Para valores negativos ou iguais a zero foi considerado o valor 0,001 representando o ganho coletivo que aquela otimização teve para o modelo. A fórmula de cálculo da importância é descrito no item abaixo:

- **Importância:** $ABM_{maiorOpt} - ABM_{maiorDem} + ABM_{menorOpt} - ABM_{menorDem}$

O resultado do processo de otimização dos valores de hiperparâmetros do modelo *Random Forest* está descrito na Tabela 7.8. Por essa análise os dois parâmetros mais importantes para a otimização dos resultados do algoritmo de *Random Forest* foram: “*max features*” e “*min samples split*”. Estes dois hiperparâmetros podem ser descritos como:

- *max features* :: número máximo de *features* estanciado em cada árvore em uma *random forest*
- *min samples split* :: indica o número mínimo de amostras necessárias para dividir um nodo interno da árvore de decisão

Tabela 7.8: Tabela como hiperparâmetros ajustados como parte dos experimentos com o algoritmo Random Forest ordenados pela sua importância relativa.

Hiperparâmetro	Importância	Intervalo	Valor Otimizado
max features	0,023	sqrt - 467	93
min samples split	0,002	2 - 500	2
n estimators	0,001	1 - 400	40
max depth	0,001	1 - 100	20
min samples leaf	0,001	1 - 1000	1
max leaf nodes	0,001	2 - 2048 - None	None
bootstrap	N/T	False - False	False
min impurity decrease	N/T	0 - 0	0
min weight fraction leaf	N/T	0 - 0	0
oob score	N/T	False - False	False
random state	N/T	47 - 47	47
warm start	N/T	False - False	False
use keras	N/T	False - False	False
verbose	N/T	Blank - Blank	Blank
folds	N/T	Blank - Blank	Blank

7.4.3 Hiperparâmetros *Extreme Gradient Boosting* - XGB

Para os hiperparâmetros aplicados ao contexto do **XGB** uma rodada de experimentos foi conduzida a partir dos quais os resultados descritos na Tabela 7.9 foram obtidos. A análise de otimização de hiperparâmetros do algoritmo **XGB** englobou 25 hiperparâmetro e os resultados obtidos estão ordenados pela sua importância.

Se destacaram pela sua influência em melhorar os resultados obtidos com esse método os parâmetros “n estimators” e “max depth”, descritos a seguir:

- *n estimators* :: número de árvores de decisão criadas em cada floresta
- *max depth* :: indica o número máximo de nodos filhos que podem crescer até que a árvore de decisão seja cortada

7.4.4 Hiperparâmetros *Multilayer Perceptron* - MLP

No contexto de hiperparâmetros aplicáveis ao algoritmo *Multilayer Perceptron* foi onde o maior número de experimentos aconteceram em diferentes rodadas de tentativas de otimização. Este foi o algoritmo para o qual as mudanças nos hiperparâmetros fizeram mais diferença nos resultados obtidos como descrito na Tabela 7.10. A análise de otimização de hiperparâmetros do algoritmo **MLP** englobou 23 hiperparâmetro e os resultados obtidos estão ordenados pela sua importância. E foram realizados experimentos com variação de valores para 19 desses parâmetros.

Ao final dos experimentos se destacaram pela sua influência em melhorar os resultados obtidos com esse método os parâmetros “epochs”, “continue training”, “learning rate init”, “metrics str”, “early stopping”, “monitor”, “n iter change”, “tol”, “hidden layer sizes”, “batch size”, “max iter” e “learning rate”. O propósito desses parâmetros está descrito na seguinte listagem:

- *epochs* :: define o número de vezes que o algoritmo de aprendizado vai processar e iterar pelo conjunto de dados completo de treino
- *learning rate init* :: controla a taxa ou velocidade de aprendizado *inicial* do modelo
- *early stopping* :: interrompe o processo de treinamento instanciado pela otimização de hiperparâmetros quando as métricas não estão melhorando efetivamente
- *n iter no change* :: define o número de iterações que o processo de otimização deve

Tabela 7.9: Tabela como hiperparâmetros ajustados como parte dos experimentos com o algoritmo XGBoost ordenados pela sua importância relativa.

Hiperparâmetro	Importância	Intervalo	Valor Otimizado
n estimators	0.052	5 - 500	5
max depth	0.018	1 - 20	1
colsample bytree	0.001	0.1 - 1	0.1
learning rate	0.001	0.01 - 0.5	0.01
min child weight	0.001	1 - 50	50
subsample	0.001	0.5 - 1	0.5
base score	0	0.5 - 0.5	0.5
booster	0	Blank - Blank	Blank
colsample bylevel	0	1 - 1	1
colsample bynode	0	1 - 1	1
min split loss	0	Blank - Blank	Blank
max delta step	0	0 - 0	0
scale pos weight	0	1 - 1	1
seed	0	47 - 47	47
use keras	0	Blank - Blank	Blank
verbose	0	Blank - Blank	Blank
Folds	0	Blank - Blank	Blank
gamma	0	0 - 0	0
missing	0	Blank - Blank	Blank
objective	0	binary:logistic	Blank
random state	0	47 - 47	47
reg alpha	0	0 - 0	0
alpha	0	Blank - Blank	Blank
reg lambda	0	1 - 1	1
lambda	0	Blank - Blank	Blank

aguardar sem melhoria efetiva das métricas antes de ser interrompido

- *tol* :: tolerância definida como critério para interrupção do processo de otimização
- *hidden layer sizes* :: representa o número de camadas escondidas adicionadas a rede neural
- *batch size* :: representa o número de amostras de treino que devem ser utilizadas durante uma iteração (época)
- *max iter* :: número máximo de iterações para serem conduzidas antes que o processo de otimização seja interrompido
- *learning rate* :: controla a taxa ou velocidade de aprendizado do modelo após as primeiras iterações

Tabela 7.10: Tabela como hiperparâmetros ajustados como parte dos experimentos com o algoritmo Multilayer Perceptron ordenados pela sua importância relativa.

Hiperparâmetro	Importância	Intervalo	Valor Otimizado
epochs	0,053	Blank - 200 - 1000	200
learning rate init	0,009	Blank, 0,001, 0,005, 0,01	0,005
early stopping	0,008	Blank - 1	Blank
n iter no change	0,008	Blank - 5	Blank
tol	0,008	Blank - 0	Blank
hidden layer sizes	0,006	[50 - 300] 1 - 4*	[100, 100]
batch size	0,005	50 - 200	100
max iter	0,005	200 - 1000	200
learning rate	0,004	0 - 0,1	0,1
blocs	0,001	0 - 5	0
Features Count	0,001	429, 438, 467	429
Folds	0,001	Blank - 3	3
loss	0,001	Blank - bin cross	bin cross
optimizer	0,001	Blank - adam	adam
use batch norm	0,001	False - True	FALSE
use dropout	0,001	Blank - False - True	TRUE
use keras	0,001	True - False	TRUE
use skip connection	0,001	False - True	FALSE
verbose	0,001	Blank - 0	0
activation	0	relu - relu	relu
alpha	0	0-0	0
shuffle	0	Blank - Blank	Blank
solver	0	adam - adam	adam

7.4.5 Linha do Tempo dos Modelos Gerados

Esta seção visa descrever os eventos que influenciaram a evolução dos modelos ao longo do tempo para que o leitor entenda as oscilações de desempenho, em alguns casos inesperadas. A Tabela 7.11 foi criada com o propósito de servir como um guia para sumarizar as diferentes nuances e o impacto que cada decisão, proposital ou arbitrária, teve nos resultados obtidos.

Tabela 7.11: Eventos que influenciaram a evolução dos modelos de aprendizado de máquina ao longo do tempo.

Data	Evento	Features	Acurácia
Jun/22	Modelo inicial <i>features</i> numéricas + amostra de categóricas) - Apenas um algoritmo: Random Forest (RF) - Experimento conduzido apenas como <i>baseline</i>	216	0,5
Jul/22	Progresso na atividade de engenharia de <i>features</i> - Diversas rodadas de treino e testes foram conduzidas, melhorando o resultado geral do modelo (RF) - <i>Features</i> que não existem no momento da predição são adicionadas ao modelo melhorando os resultado <i>artificialmente</i>	212	0,93
Ago/22	A fase de engenharia de <i>features</i> chega ao seu ápice e um modelo é gerado com 618 features	618	0,946
Ago/22	Primeiro experimento com treino balanceado (<i>undersampling</i>) - Um grande número de <i>features</i> são mapeadas como não podendo ser utilizadas por não estarem presentes no momento da predição ou mesmo por questões de conformidade. Enquanto essas discussões amadurecem voltamos ao estágio anterior incluindo apenas as <i>features</i> do <i>baseline</i> e um novo conjunto de <i>features</i> numéricas (experimento descrito na Figura 7.7)	316	0,87
	Primeiro modelo XGB gerado	316	0,965
	Primeiro modelo MLP gerado	316	0,838
Ago/22	Modelo treinado com todas as transações - Primeiros testes realizados com intervalos de dados pré-definidos	574	0,9
Set/22	Redução do número de <i>features</i> para remover viés indesejado e eliminar problemas de compliance Este modelo ainda tinha <i>features</i> correlacionadas a variável alvo.	368	0,97
Out/22	MLP selecionado para gerar API Mudança na lógica de rótulos das transações Features que refletem a decisão do MPBR são identificadas e removidas do conjunto de dados	467	0,86
Nov/22	Por diretiva corporativa todo um conjunto de <i>features</i> é suprimido do conjunto de dados	429	0,89
Dez/22	Variáveis sumarizadas do tipo stag são removidas Features com informação de crédito do cliente são removidas Desenvolvimento da API (pipeline)	273	0,88
Jan/23	Mudança na lógica de seleção de registros para treinamento (descarte)	273	0,88
	API integrada em produção	237	0,87

A relação de *features* utilizada para alimentar o modelo de predição de fraudes

passou por diversos ajustes ao longo das diferentes etapas deste trabalho. Nesta seção é proposta uma análise das diferentes versões dos modelos gerados ao longo do tempo, linha de tempo essa sumarizada na tabela 7.11.

Em um primeiro momento, um modelo baseado em Random Forest (RF) foi utilizado para avaliar a relevância e desenvolver um modelo que serviria de *baseline* para as melhorias a serem desenvolvidas a seguir. Para esse modelo, o conjunto de dados foi separado em treinamento e testes de forma aleatória. Com um número de 216 *features*, o modelo atingiu uma precisão balanceada superior a 90%. Uma precisão melhor do que as obtidas pelos modelos mais recentes. No entanto, o conjunto de dados de treinamento não foi dividido por intervalos de tempo, tendendo a maximizar as métricas de desempenho do modelo. Outro fator que influenciou esses resultados foi a utilização de *features* derivadas da decisão do MPBR com alta correlação com o rótulo utilizado variável alvo. Essas *features* acabaram sendo removidas posteriormente para eliminar o viés de uma entrada que não existe no momento da predição.

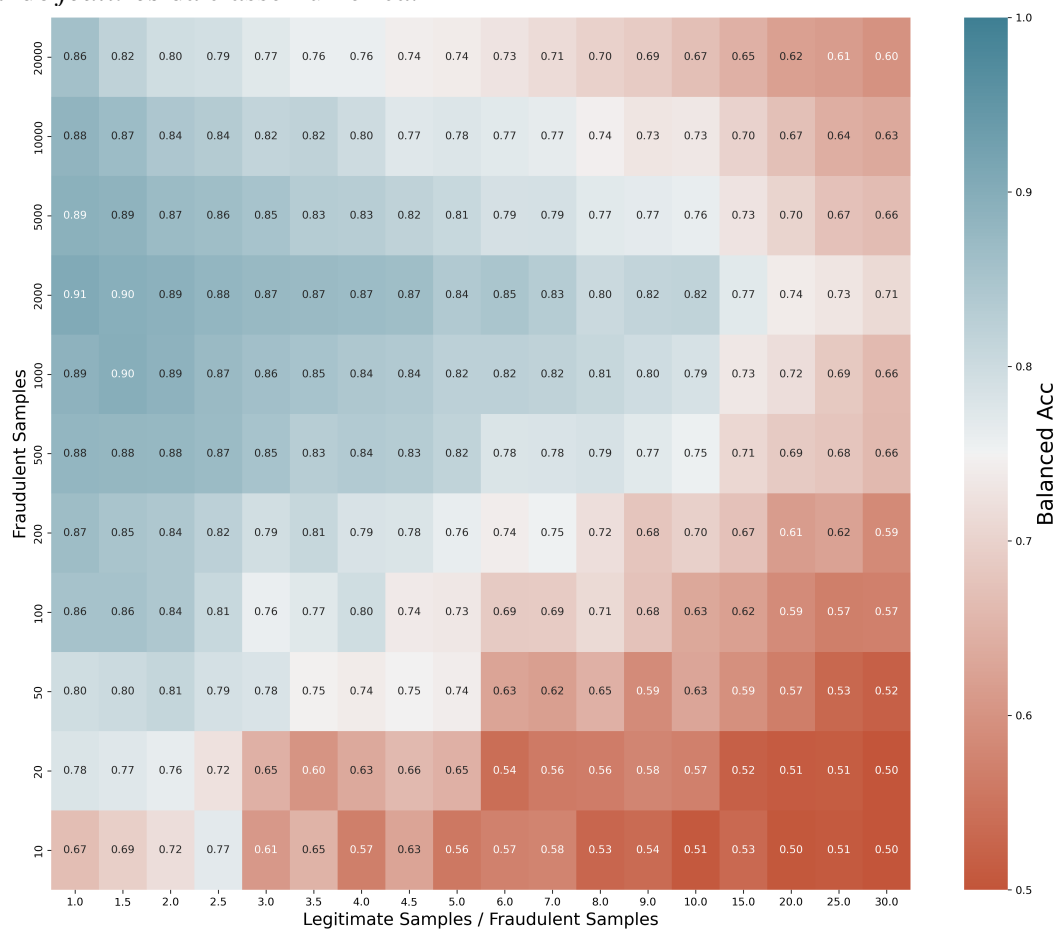
Durante a fase de “Engenharia de *Features*”, foi gerada uma versão inicial para cada modelo (RF, XGB e MLP) treinadas com o número máximo de *features* extraídas do conjunto de dados (618). Após a primeira fase de seleção de *features*, o número total de entradas para o modelo caiu para 574 com uma precisão balanceada em torno de 90%. As *features* removidas nessa fase foram as descartadas por características dos seus valores, fossem eles contantes, vazios ou aleatórios ao alvo, sendo assim considerados inadequados para o treinamento e aprendizado de máquina.

Ainda durante as atividades do mês de agosto, um novo conjunto de *features* numéricas foi pré-processado e testado como candidato para serem adicionadas ao conjunto de dados, algumas delas passaram a fazer parte dos modelos a partir desse exercício.

O resultado da análise da contribuição das *features* numéricas pode ser visto na Figura 7.7, onde o eixo horizontal (X) mostra a relação entre as classes legítima e a fraudulenta no treinamento. O eixo vertical (Y) mostra o número de fraudes no conjunto de dados. Observe que o número total de fraudes na amostra é de 2000, portanto valores acima de desse valor caracterizam excessos de falsos positivos.

Ao final dessa limpeza no conjunto de dados, que removeu algumas *features* atualizadas com informação disponível apenas após a decisão do **MPBR**, a taxa de precisão média caiu abaixo de 85%. Foi quando o exercício de otimização de hiperparâmetros iniciou usando uma técnica de *grid-search* e o valor da precisão média voltou para a faixa dos 88%.

Figura 7.7: Acurácia balanceada para diferentes conjuntos de dados de treinamento utilizando *features* da classe numérica.



Fonte: Elaborado pelo autor

Em um estágio seguinte do processo de engenharia de *features* algumas fontes de dados tiveram que ser cortadas em função de fatores externos. Primeiramente recebemos a confirmação de que o pacote de dados de “Hierarquia dos Produtos” originalmente parte do escopo de dados, teria que ser descartada. Em seguida, outra fonte relevante de *features* teve que ser descartada em função de questões de conformidade legal por se tratarem de dados referentes a aplicação de crédito dos clientes, ao final desse processo chegamos a um modelo baseado em um conjunto de dados com 467 *features* que manteve uma precisão média de 88%.

No momento de integrar o modelo com o fluxo de transações em produção, foram identificados novos campos que não eram enviados para o modelo no momento da predição ou que simplesmente não eram preenchidos. Para evitar que estes campos causassem ruído no processo de predição, todos os campos foram removidos para evitar impacto no desempenho das predições. O modelo final chegou a produção com 237 *features* e 87% de acurácia.

A Tabela 7.12 mostra o histórico de todos os modelos *Random Forest* gerados ao longo do tempo com suas respectivas métricas. Foram ao todo 9 versões diferentes de modelos RF treinados e testados com diferentes estágios de maturidade do conjunto de dados disponível entre junho e novembro de 2022. A tabela também mostra características do modelo quanto a implementação ou não de balanceamento de dados, otimização de hiperparâmetros e a quantidade de *features* utilizadas.

Tabela 7.12: Listagem de modelos Random Forest (RF) gerados durante os experimentos.

Métricas ML	Random Forest								
	RF 1.0	RF 2.0	RF 3.0	RF 4.0	RF 5.0	RF 6.0	RF 7.0	RF 8.0	RF 9.0
Acurácia	0,5	0,933	0,946	0,87	0,904	0,971	0,973	0,819	0,896
Precisão	0	0,926	0,019	0,821	0,018	0,312	0,135	0,076	0,034
Sensitividade	0	0,902	0,983	0,741	0,896	0,959	0,966	0,731	0,869
Especificidade	1	N/D	0,908	0,999	0,911	0,983	0,981	0,908	0,922
Pontuação F1	0	0,914	0,038	0,779	0,036	0,47	0,237	0,138	0,065
TN	N/D	N/D	N/D	296818	259624	292058	45137	231292	42450
FP	N/D	N/D	N/D	374	25339	4906	895	23519	3582
FN	N/D	N/D	N/D	601	55	96	5	718	19
TP	N/D	N/D	N/D	1717	474	2222	140	1948	126
<i>Features</i>	212	216	618	316	574	369	368	467	429
Hiperparâm.	N	N	N	N	N	N	N	S	S
<i>Undersampling</i>	F	F	F	V	V	V	V	V	V
Período	Jun/22	Jul/22	Ago/22	Ago/22	Ago/22	Ago/22	Set/22	Out/22	Nov/22

A Tabela 7.13 mostra o histórico de todos os modelos *XGBoost* gerados ao longo do tempo com suas respectivas métricas. Foram ao todo 6 versões diferentes de modelos XGB treinados e testados com diferentes estágios de maturidade do conjunto de dados disponível entre agosto e novembro de 2022. A tabela também mostra características do modelo quanto a implementação ou não de balanceamento de dados, otimização de hiperparâmetros e a quantidade de *features* utilizadas.

Tabela 7.13: Listagem de modelos Extreme Gradient Boost (XGB) gerados durante os experimentos.

Métricas ML	XGBoost					
	XGB 1.0	XGB 2.0	XGB 3.0	XGB 4.0	XGB 5.0	XGB 6.0
Acurácia	0,965	0,973	0,971	0,97	0,871	0,865
Precisão	0,304	0,309	0,339	0,139	0,08	0,042
Sensitividade	0,947	0,964	0,956	0,959	0,843	0
Especificidade	0,983	0,983	0,985	0,981	0,899	0
Pontuação F1	0,461	0,468	0,5	0,243	0,147	0,079
TN	292169	291975	292638	45174	229130	43418
FP	5023	4989	4326	858	25681	2614
FN	122	84	102	6	418	31
TP	2196	2234	2216	139	2248	114
<i>Features</i>	316	372	369	368	467	429
Hiperparâm.	N	N	N	N	S	S
<i>Undersampling</i>	V	V	V	V	V	V
Período	Ago/22	Ago/22	Ago/22	Set/22	Out/22	Nov/22

A Tabela 7.14 mostra o histórico de todos os modelos *Multilayer Perceptron* gerados ao longo do tempo com suas respectivas métricas. Foram ao todo 8 versões diferentes de modelos MLP treinados e testados entre agosto de 2022 e novembro de 2023. A tabela também mostra características do modelo quanto a implementação de otimização de hiperparâmetros e a quantidade de *features* utilizadas. Todas as versões de MLP tiveram balanceamento de classes de dados.

As Tabelas com informações sobre os modelos apresentadas ao longo dessa seção, se propõem a oferecer uma visão geral dos resultados obtidos com cada algoritmo. As considerações sobre desempenho dos modelos, bem como análise comparativa dos seus resultados, serão apresentadas no Capítulo 8 dos “Resultados dos Experimentos”.

Tabela 7.14: Listagem de modelos Multilayer Perceptron (MLP) gerados durante os experimentos.

Métricas ML	Multilayer Perceptron							
	MLP 1.0	MLP 2.0	MLP 3.0	MLP 4.0	MLP 5.0	MLP 6.0	MLP 7.0	MLP 8.0
Acurácia	0.838	0.839	0.968	0.899	0.856	0.905	0.864	0.895
Precisão	0.044	0.101	0.224	0.045	0.156	0.04	0.067	0.076
Sensitividade	0.841	0.748	0.963	0.855	0.755	0.876	0.809	0.844
Especificidade	0.836	0.931	0.974	0.943	0.957	0.933	0.919	0.895
Pontuação F1	0.084	0.178	0.363	0.086	0.258	0.076	0.123	0.139
TN	251119	237118	289222	43411	243905	42955	231659	228196
FP	49427	17693	7742	2621	10906	3077	20522	26684
FN	435	673	86	21	654	18	346	406
TP	2301	1993	2232	124	2012	127	1468	2191
<i>Features</i>	486	414	369	368	467	429	438	237
Hyperparâm.	N	N	N	N	S	S	S	S
<i>Undersampling</i>	V	V	V	V	V	V	V	V
Período	Ago/22	Ago/22	Ago/22	Set/22	Out/22	Nov/22	Dez/22	Jan/23

8 RESULTADOS DOS EXPERIMENTOS

Neste capítulo são apresentados os resultados dos experimentos realizados com cada um dos modelos de aprendizado de máquina testados e comparados com o nosso *baseline* que é o sistema de regras de decisão **MPBR**. Os resultados obtidos são então analisados e as principais conquistas e realizações serão debatidas em uma seção de considerações finais.

8.1 Configuração dos Experimentos

O objetivo dos experimentos conduzidos ao longo da pesquisa foi avaliar se um modelo de aprendizado de máquina pode ser tão efetivo na tarefa de predição de transações fraudulentas quanto um sistema de regras de decisão configurado ao longo de anos para executar apenas essa tarefa. Em nossos experimentos o nosso *baseline* é representado por esse sistema de regras de decisão (**MPBR**).

A Figura 8.1 descreve uma linha de tempo através do qual todos os experimentos foram realizados com ênfase para os 4 estudos de caso e os fatores que os diferenciam entre si:

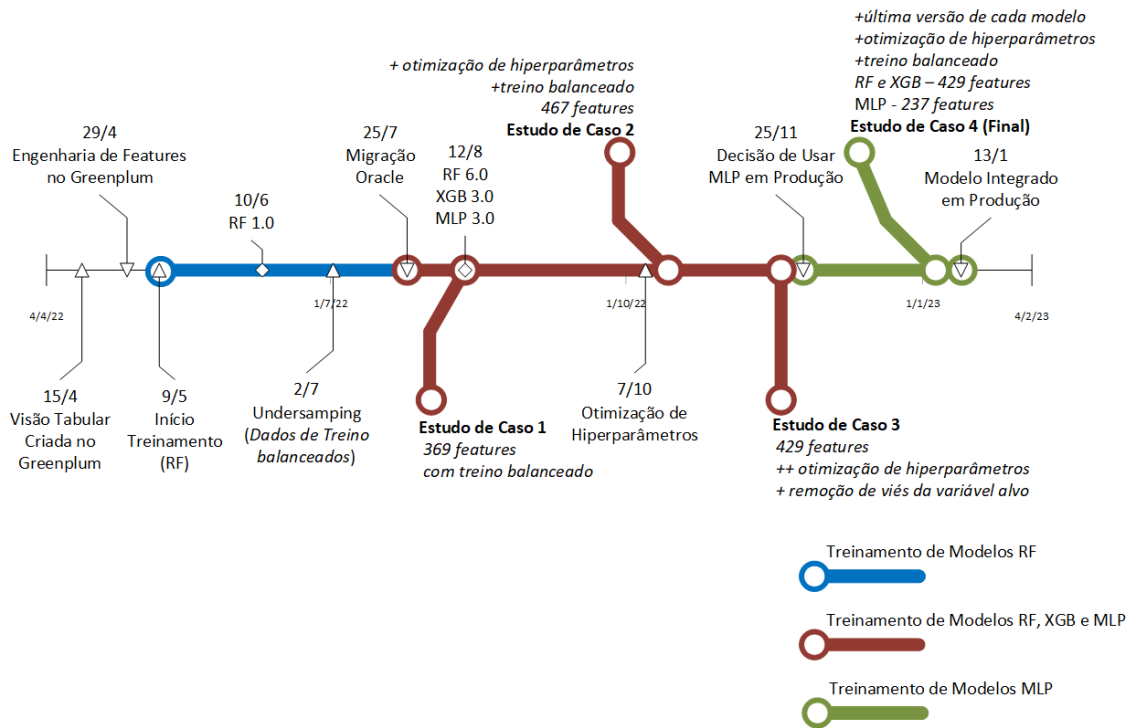
- Quantidade de *features* incluídas em cada modelo
- Estágio do processo de engenharia de *features*
- Conjunto de dados usados para os testes
- Utilização de uma abordagem de balanceamento de classes dos dados (*Undersampling*) - Treino balanceado
- Otimização de hiperparâmetros (em estágio final ou em estágio final)
- Remoção de *features* derivadas de alguma forma da variável alvo e da decisão do sistema **MPBR**.

Após uma tentativa de usar um *data lake* Greenplum para realizar os experimentos iniciais da iniciativa, que acabou esbarrando em problemas de latência de rede entre os *data centers* da base das aplicações e o *data lake* propriamente dito causando que o tempo de carga de dados de uma transação XML levasse aproximadamente 10 segundos, a decisão foi provisionar uma base dedicada na mesma localização das aplicações de origem.

O banco de dados dedicado para os experimentos é um banco Oracle 19c rodando

Figura 8.1: Metodologia dos Experimentos - Linha do tempo descrevendo todos os eventos com relevância para afetar o resultado dos experimentos destacando os momentos em que cada estudo de caso foi executado.

Metodologia dos Experimentos – Linha do Tempo



Fonte: Elaborado pelo autor

em um servidor Oracle Linux 9 com 64GB de RAM, com 150 TB de disco e 3TB de temp table para execução de processos em memória. A quantidade de memória na temp table se mostrou necessária em função do processo utilizado para geração da visão tabular.

O volume de dados carregados no banco Oracle representa cerca de 30% da capacidade física do banco (~50 TB) comportando 1.75 milhões de registros que representam 3 anos de transações (2020, 2021 e 2023). É importante salientar que os dados do primeiro semestre de 2020 foram desconsiderados para treino e teste por possuírem um viés em função do comportamento das transações no início da pandemia e as decisões de negócio relacionadas a como lidar com aquele comportamento anômalo.

8.2 Execução dos Experimentos

Nesta seção, são descritos todos os experimentos realizados para provar que a solução proposta nesta dissertação de mestrado pode ser implementada em um cenário de detecção de fraudes da vida real.

8.2.1 Execução do Baseline

Nesta seção, são debatidos aspectos do desempenho do sistema de regras de decisão MPBR em efetivamente identificar transações fraudulentas e prevenir impacto monetário para o negócio. Os números considerados para essa análise preliminar dizem respeito ao período de janeiro de 2021. A Tabela 8.1 descreve a matriz de confusão para as transações desse período que chegaram a um total de 46.177 transações, cerca de 1.500 transações por dia, sempre lembrando que os finais de semana apresentam uma demanda menor de transações em períodos normais.

Alguns aspectos para se considerar para avaliar o MPBR:

- Trata-se de um sistema corporativo de regras de decisão com quase 15 anos de existência que recebe atualizações nas suas regras, anualmente, para se manter efetivo.
- Foi implementado de forma a priorizar erros do tipo Falso Negativo (quando uma transação fraudulenta é aprovada) para reduzir o desgaste e privilegiar a manutenção da relação com o cliente.
 - *Racional*: toda vez que um cliente verdadeiro tem uma transação declinada

por fraude isso gera um desgaste.

- Por apresentar essa característica torna-se um sistema extremamente competitivo para as métricas de Verdadeiros Negativos, Falsos Positivos, Especificidade e Acurácia.
- Os modelos de aprendizado de máquina só foram superiores ao **MPBR** nessas quatro métricas quando utilizaram alguma informação derivada da variável alvo indevidamente: Verdadeiros Negativos, Falsos Positivos, Especificidade e Acurácia.
- A métrica da acurácia balanceada foi adotada para equilibrar os resultados dos modelos, já que nesse caso o valor da métrica é calculado pela média da soma dos acertos e erros (positivos e negativos).

Tabela 8.1: Matriz de confusão gerada a partir dos resultados de um mês de transações processadas para pelo sistema de regras de decisão corporativo (janeiro de 2021).

Predição	Sistema de Regras de Decisão	
	Positivo Fraude	Negativo Legítimo
Positivo	81	1676
Negativo	64	44356

Na Tabela 8.2 podemos ver as principais métricas calculadas para o sistema de decisão baseado nos números gerais da matriz de confusão do MPBR.

Tabela 8.2: Tabela de métricas calculadas para o sistema de regras de decisão referente ao período de janeiro de 2021.

Métrica	Sistema de Regras de Decisão (MPBR)
Acurácia Balanceada	0,5273
Coefficiente de Correlação de Matthews	0.0581
Sensitividade TPR	0,0578
Especificidade TNR	0,9968
Precisão PPV	0,0654
Acurácia	0,9932
Pontuação F1	0,0613

8.2.2 Execução dos Experimentos com os Modelos de Aprendizado de Máquina

Para comparar os resultados obtidos pelo *baseline* (sistema **MPBR**) com os resultados apresentados pelos modelos de aprendizado de máquina foram identificados 3

momentos em que um mesmo conjunto de *features* foi usado para treinar os 3 modelos **RF**, **XGB** e **MLP**. Para poder efetivamente comparar os resultados as métricas para as transações equivalentes ao conjunto de testes foram extraídas tb para o **MPBR** equiparando assim os valores disponíveis para a análise das métricas.

Nas próximas seções estão descritos as particularidades de cada um destes estudos de caso:

- Estudo de caso 1 - Modelos com 369 *features* [Ago/22]
- Estudo de caso 2 - Modelos com 467 *features* [Out/22]
- Estudo de caso 3 - Modelos com 429 *features* [Nov/22]

8.2.3 Estudo de caso 1 - Modelos com 369 *features*













































Objetivo: comparar o melhor modelo *Random Forest* criado até então, contra versões refinadas dos modelos *XGBoost* e *MLP* para entender qual modelo apresenta o melhor desempenho para predição de fraudes com o conjunto de *features* disponível (369 *features*).

Suposição: os modelos de aprendizado de máquina gerados como parte desse estudo de caso são capazes de predizer fraudes de forma tão efetiva quanto o sistema de regras de decisão *MPBR*.

Alguns aspectos para se levar em consideração para avaliar o estudo de caso 1:

- Os resultados apresentados pelos modelos de aprendizado de máquina estão entre os melhores obtidos ao longo de todo o processo de otimização.
- Posteriormente foi descoberto que os resultados só atingiram esse patamar em função de *features* derivadas da variável alvo que ainda estavam no conjunto de dados.
- O modelo **XGB** foi o que melhor performou com esse contexto de *features* e conjuntos de teste e treino.
- A melhor acurácia balanceada foi obtida pelo modelo *Random Forest*, sendo o único critério em que esse modelo foi superior aos demais.
- Esta versão dos modelos ainda não contava com nenhuma otimização de hiperparâmetros.
- A técnica de balanceamento de dados por sub-amostragem já estava sendo aplicada nessa versão do modelo.

Tabela 8.3: Comparação dos Resultados do Baseline e Experimentos para os Modelos com 369 *features* [Agosto de 2022]

Dados →	Jan - Ago 2022			
Modelos Métricas	MPBR	MLP3.0	XGB3.0	RF6.0
Verdadeiros Positivos (TP)	1550 - 	2232 -  44%	2216 -  43%	2222 -  43%
Verdadeiros Negativos (TN)	283409 - 	289222 -  2%	292638 -  3%	292058 -  3%
Falsos Positivos (FP)	13555 - 	7742 -  43%	4326 -  68%	4906 -  64%
Falsos Negativos (FN)	768 - 	86 -  89%	102 -  87%	96 -  88%
Acurácia Balanceada	0.8119 - 	0.9684 -  19%	0.9707 -  20%	0.9711 -  20%
Acurácia	0.9531 - 	0.9738 -  2%	0.9852 -  3%	0.9833 -  3%
Sensitividade (TPR)	0.6687 - 	0.9629 -  44%	0.95600 -  43%	0.95860 -  43%
Especificidade (TNR)	0.9552 - 	0.9739 -  2%	0.9854 -  3%	0.9835 -  3%
Precisão (PPV)	0.1201 - 	0.2238 -  86%	0.3387 -  182%	0.3117 -  160%
Pontuação F1	0.2012 - 	0.3632 -  81%	0.5002 -  149%	0.4705 -  134%
Correlação de Matthews	0.2578 - 	0.4576 -  78%	0.5644 -  119%	0.5416 -  110%

8.2.4 Estudo de caso 2 - Modelos com 467 *features*

Objetivo: verificar o quanto a otimização de hiperparâmetros beneficiou os modelos de aprendizado de máquina se comparados com suas versões anteriores, entre si e principalmente em comparação com o *baseline* (**MPBR**).

Suposição: o processo de otimização de hiperparâmetros fez com que os modelos de aprendizado de máquina atingissem o mesmo patamar de eficácia do sistema de regras de decisão **MPBR**.

Alguns aspectos para se considerar para avaliar o estudo de caso 2:

- Muito embora os resultados obtidos com essa versão dos modelos estejam muito mais próximos dos produzidos pelo próprio sistema de regras de decisão (**MPBR**), é correto afirmar que ele ainda possuía *features* geradas a partir de logs da decisão do **MPBR** que influenciaram positivamente os resultados dos modelos.
- O modelo **MLP** foi o que melhor performou com esse contexto de *features* e conjuntos de teste e treino. Tendo apresentado os melhores valores em 6 das métricas analisadas.
- A melhor acurácia balanceada, entretanto, foi obtida pelo modelo *XGBoost*. Que também foi o melhor modelo em termos de sensibilidade (TPR).
- Esta foi a primeira versão dos modelos a contar com otimização de hiperparâmetros, que ainda não havia explorada em profundidade quando essa versão dos modelos foi gerada.
- Este modelo já incorporou uma mudança feita na lógica utilizada para rotular transações fraudulentas para desconsiderar transações antigas da conta bloqueada por fraude.

Tabela 8.4: Comparação dos Resultados do Baseline e Experimentos para os Modelos com 467 *features* [Outubro de 2022]

Dados →	Mar - Set 2022			
Modelos Métricas	MPBR	MLP5.0	XGB5.0	RF8.0
Verdadeiros Positivos (TP)	1893 -	2012 - 6%	2248 - 19%	1948 - 3%
Verdadeiros Negativos (TN)	243916 -	243905 - 0,005%	229130 - 6%	231292 - 5%
Falsos Positivos (FP)	10895 -	10906 - 0,1%	25681 - 136%	23519 - 116%
Falsos Negativos (FN)	773 -	654 - 15%	418 - 46%	718 - 7%
Acurácia Balanceada	0,8336 -	0,8559 - 3%	0,8712 - 5%	0,8192 - 2%
Acurácia	0,9547 -	0,9551 - 0,04%	0,8986 - 6%	0,9059 - 5%
Sensitividade (TPR)	0,7101 -	0,7547 - 6%	0,8432 - 19%	0,7307 - 3%
Especificidade (TNR)	0,9572 -	0,9571 - 0,01%	0,8992 - 6%	0,9077 - 5%
Precisão (PPV)	0,1480 -	0,1558 - 5%	0,0805 - 46%	0,0765 - 48%
Pontuação F1	0,2450 -	0,2582 - 5%	0,1470 - 40%	0,1385 - 43%
Correlação de Matthews	0,3109 -	0,3301 - 6%	0,2417 - 22%	0,2165 - 30%

8.2.5 Estudo de caso 3 - Modelos com 429 *features*

Objetivo: mensurar o impacto para os modelos de aprendizado de máquina da remoção de *features* que traziam uma informação implícita sobre a variável alvo e a decisão do sistema **MPBR**.

Suposição: mesmo sem as *features* que introduziam um viés positivo para a predição de fraudes, os modelos de aprendizado de máquina ainda apresentam um desempenho competitivo com o sistema de regras de decisão **MPBR**.

Alguns aspectos para se levar em consideração para avaliar o estudo de caso 3:

- Os resultados apresentados pelos modelos de aprendizado de máquina são compatíveis com os apresentados pelos modelos finais, indicando que não temos nenhum fator externo desequilibrando os resultados para melhor ou pior nesse caso.
- O sistema de regras de decisão **MPBR** obteve os melhores resultados para 7 de 11 métricas avaliadas.
- Seguido de perto pelo modelo **MLP** o melhor nas outras 4 métricas, incluindo acurácia balanceada.
- O processo de otimização de hiperparâmetros estava finalizado quando essa bateria de testes foi realizada.
- Quando essa bateria de testes foi conduzida já existia uma decisão pela escolha do modelo MLP para integração em produção. A partir desse momento o modelo MLP passou a ser o único a sofrer manutenção e participar de novas rodadas de teste.

Tabela 8.5: Comparação dos Resultados do Baseline e Experimentos para os Modelos com 429 *features* [Novembro de 2022]

Dados →	Ago - Out 2022			
Modelos Métricas	MPBR	MLP6.0	XGB6.0	RF9.0
Verdadeiros Positivos (TP)	99 -	127 - 28%	114 - 15%	126 - 27%
Verdadeiros Negativos (TN)	44255 -	42955 - 3%	43418 - 2%	42450 - 4%
Falsos Positivos (FP)	1777 -	3077 - 73%	2614 - 47%	3582 - 102%
Falsos Negativos (FN)	46 -	18 - 61%	31 - 33%	19 - 59%
Acurácia Balanceada	0.8221 -	0.9046 - 10%	0.8647 - 5%	0.8956 - 9%
Acurácia	0.9605 -	0.9330 - 3%	0.9427 - 2%	0.9220 - 4%
Sensitividade (TPR)	0.6828 -	0.8759 - 28%	0.7862 - 15%	0.8690 - 27%
Especificidade (TNR)	0.9614 -	0.9332 - 3%	0.9432 - 2%	0.9222 - 4%
Precisão (PPV)	0.0528 -	0.0396 - 25%	0.0418 - 21%	0.034 - 36%
Pontuação F1	0.098 -	0.0758 - 23%	0.0794 - 19%	0.0654 - 33%
Correlação de Matthews	0.1825 -	0.1781 - 2%	0.1731 - 5%	0.1629 - 11%

8.2.6 Estudo de caso com a versão final de cada modelo

Objetivo: comparar as métricas do modelo *MLP* utilizado em produção contra as versões mais atuais dos modelos *RF* e *XGB* para garantir que não houve uma perda de desempenho em função da redução do número de features que o processo de integração com o fluxo *online* de transações demandou.

Suposição: a versão final do modelo de aprendizado de máquina *MLP* é competitivo com as métricas apresentadas pelo sistema de regras de decisão/*baseline* **MPBR**.

Alguns aspectos para se levar em consideração para avaliar os resultados obtidos pela versão final de cada modelo:

- Esta é a única comparação de resultados em que os modelos não compartilham as mesmas características de treinamento. O modelo *MLP* testado foi o último gerado com *237 features* enquanto os modelos *MLP* e *XGB* contam com *429 features*. Essa diferença acontece pois os outros modelos deixaram de ser mantidos após Nov/2022.
- O sistema de regras de decisão **MPBR** manteve os melhores resultados para 7 de 11 métricas avaliadas.
- As outras 4 métricas, no entanto foram divididas entre *MLP* e *XGB* que obteve melhor resultado em Verdadeiros Positivos, Falsos Negativos e Sensitividade. O modelo *MLP* manteve a melhor Acurácia Balanceada, sendo um ótimo indicador.
- O modelo **RF** mesmo não tendo sido o melhor em nenhuma métrica apresentou o melhor resultado entre os 3 modelos em todas as métricas em que o **MPBR** apresentou melhores resultados.

8.3 Considerações sobre os resultados obtidos

Lista de considerações sobre os resultados obtidos durante os quatro estudos de casos apresentados nessa dissertação:

- A escolha do modelo *MLP* para ser usado ao final do processo considerou o seu equilíbrio entre as métricas indicado pelo “Acurácia Balanceada”, mas os principais fatores foram aspectos qualitativos associados, por exemplo, a capacidade do *MLP* de tratar dados e maior flexibilidade para conduzir retreinamento parcial e total.

Tabela 8.6: Comparação dos Resultados do Baseline e Experimentos

Dados →	Ago - Dez 2021			
Modelos Métricas	MPBR	MLP	XGB	RF
Verdadeiros Positivos (TP)	1607 -	2191 -	2291 -	2114 -
		36%	43%	32%
Verdadeiros Negativos (TN)	243108 -	228196 -	222071 -	232098 -
		-6%	-9%	-5%
Falsos Positivos (FP)	11772 -	26684 -	32740 -	22713 -
		127%	178%	93%
Falsos Negativos (FN)	990 -	406 -	375 -	552 -
		59%	62%	44%
Acurácia Balanceada	0,7863 -	0,8695 -	0,8654 -	0,8519 -
		11%	10%	8%
Acurácia	0,9504 -	0,8948 -	0,8714 -	0,9096 -
		-6%	-8%	-4%
Sensitividade (TPR)	0,6188 -	0,8437 -	0,8593 -	0,7929 -
		36%	39%	28%
Especificidade (TNR)	0,9538 -	0,8953 -	0,8715 -	0,9109 -
		-6%	-9%	-4%
Precisão (PPV)	0,1201 -	0,0759 -	0,0654 -	0,0851 -
		-37%	-46%	-29%
Pontuação F1	0,2012 -	0,1392 -	0,1215 -	0,1538 -
		-31%	-40%	-24%
Correlação de Matthews	0,2578 -	0,2340 -	0,2158 -	0,2414 -
		-9%	-16%	-6%

- Se comparado com o XGB, o MLP pode ser considerado um algoritmo mais flexível.
- No que diz respeito a superioridade do sistema de decisão **MBPR** aos modelos no quesito redução de falsos negativos, esse pode ser um aspecto positivo se aliarmos o modelo **MLP** como uma fonte de dados adicional para o **MBPR**.
- Outra opção, associada a possíveis trabalhos futuros é parametrizar esse inclinação no balanço das decisões estabelecendo que a probabilidade de fraude mínima para uma transação ser identificada como fraudulenta pelo modelo **MLP** passe a ser 70% ao invés do 51% valor atual, e que corresponde ao comportamento padrão do algoritmo.

Principais resultados positivos: a acurácia balanceada do modelo MLP foi superior a apresentada pelo **MPBR** tanto no experimento final quanto nos resultados em produção, o que mostra que os modelos de aprendizado de máquina podem ser competitivos.

Solução ideal: a implementação de um ciclo de retreino permitirá que o modelo MLP evolua de forma automatizada, melhorando o desempenho obtido durante os experimentos aqui apresentados.

Casos de falha: principalmente em intervalos de tempo onde o número de fraudes cai, as métricas do modelo MLP tendem a se distanciar negativamente do **MPBR**. Esse fenômeno ocorre em função da natureza do sistema de decisões que prioriza o erro pelo aspecto de falsos negativos para melhorar a relação com os clientes.

9 CONCLUSÃO

Esta dissertação apresentou uma revisão sistemática sobre o impacto das “Técnicas Emergentes” para combater o crescimento desenfreado da quantidade de fraudes que vem se intensificando ao longo dos anos. Uma taxonomia de “Técnicas Emergentes” é apresentada para estruturar os resultados da revisão sistemática, e colocada lado-a-lado com uma taxonomia de Tipos de Fraude. Por fim, a contribuição principal: uma solução de predição de fraudes baseada em algoritmos de aprendizado de máquina aplicada a um contexto de aprovação de crédito em transações de compra.

O cerne deste trabalho compreende uma proposta de solução para predição de fraudes em transações de crédito baseada em aprendizado de máquina, onde suas 4 etapas são detalhadas e decompostas em um total de 20 fases que são descritas em detalhes.

Esta proposta é então colocada em prática em um conjunto de dados real que representa transações de financiamento de compras processadas entre 2020 e 2022. O processo de implementação da solução de predição de fraudes, nesse contexto, é descrito em detalhes, bem como o conjunto de dados utilizado para desenvolver os modelos de aprendizado de máquina e o pré-processamento realizado para preparar os dados e alimentar os modelos efetivamente.

O *baseline* adotado para mensurar os resultados obtidos durante os experimentos representa uma solução corporativa de combate a fraude que utiliza um sistema de decisão composto por um modelo de pontuação baseado em regras (MPBR). Esta solução corporativa tem suas características e métricas apresentadas com riqueza de detalhes.

Durante o processo de implementação da proposta 3 algoritmos diferentes de aprendizado de máquina foram selecionados e utilizados na geração de modelos: (1) Random Forest [RF], (2) Extreme Gradient Boost [XGB] e (3) Redes Neurais Multilayer Perceptron [MLP]. Ao todo, 23 versões de modelo diferentes foram geradas ao longo do processo evolutivo das *features* e implementação de otimizações em aspectos de balanceamento de dados e hiperparâmetros.

Os experimentos desenvolvidos ao longo da implementação da proposta envolvem comparação dos resultados dos 3 algoritmos entre eles e com o *baseline* (MPBR). Para possibilitar uma comparação mais fiel e confiável entre os resultados dos modelos de aprendizado de máquina, 4 estudos de casos foram propostos para os quais os modelos foram treinados e testados com um mesmo conjunto de dados, incluindo o mesmo conjunto de *features*. Os resultados dos experimentos apontaram o modelo MLP como o

mais equilibrado na comparação de volume de fraudes reais detectadas contra o volume de alertas falsos (falsos positivos) e fraudes que passaram despercebidas (falsos negativos), esta informação é apontada pela métrica de “Acurácia Balanceada” que ficou em 86,95% no resultado do modelo final gerado e testado com dados de agosto a dezembro de 2021. Outro aspecto positivo associado ao MLP é sua flexibilidade para retreino, muito superior ao XGB, por exemplo.

Uma observação que pode ser feita sobre os resultados dos experimentos é a superioridade do *baseline* (MPBR) no quesito menor número de alertas falsos, este fator se deve a uma decisão de design do sistema de decisão que prioriza o bom relacionamento com os clientes. No estudo de caso final (4) o modelo que mais se aproximou do MPBR nessa métrica foi o RF com 93% a mais de falsos alertas, ou seja, quase o dobro, para aquele experimento. O modelo XGB se destacou como o melhor em métricas importantes como *Recall* (TPR) com 85,9% e volume total de verdadeiros positivos, ficando quase 43% acima do MPBR nesse indicador. O modelo RF por sua vez não foi superior a todos os concorrentes em nenhuma métrica, mas ficou logo atrás do MPBR em todas as métricas vencidas pelo sistema de decisão (7 ao todo), mostrando uma maior proximidade de comportamento entre o sistema de decisão (MPBR) e o modelo RF.

A evolução natural da solução de predição baseada em aprendizado de máquina vai ao encontro a implementação do ciclo de retreino, permitindo a evolução do seu desempenho obtido durante os experimentos de forma automatizada e recorrente. Uma limitação identificada diz respeito ao aumento da distância nas métricas do MPBR para os modelos ML quando o número de fraudes no período é menor. Esse fenômeno ocorre em função da natureza do sistema de decisões que prioriza o erro pelo aspecto de falsos negativos para melhorar a relação com os clientes. Essa característica do MPBR pode ser um aspecto positivo se aliarmos o modelo ML como uma fonte de dados adicional para as decisões do **MBPR**.

Os resultados específicos dos experimentos demonstram que as técnicas emergentes representadas nesse trabalho por algoritmos de aprendizado de máquina agregam valor ao contexto de detecção de fraudes, melhorando métricas importantes dos sistemas de decisão clássicos e podendo atuar em parceria com eles para reduzir custo operacional tanto na atualização periódicas das regras e equações de um MPBR quanto no volume de transações que demandam uma revisão manual por parte da equipe de operações.

9.1 Direções futuras

Em continuidade a este trabalho, algumas direções futuras foram identificadas referentes a expansão da solução de predição de fraudes e a sua otimização. Os seguintes aspectos são considerados melhorias importantes para o trabalho desenvolvido até aqui:

- Desenvolver funcionalidade de explicabilidade que permita ao usuário do sistema entender porque uma transação foi predita como “fraudulenta” ou “legítima” pelo modelo. Esse item pode ser considerado uma questão de conformidade da solução, se pensarmos que um cliente tem suporte jurídico para solicitar que a empresa explique por que uma compra específica não foi autorizada.
- Implementar o retreinamento baseado no conjunto de *features* atualmente incorporadas ao modelo.
- Implementar uma extensão do retreinamento que permita que novos atributos (*features*) sejam incorporados ao modelo a medida que novos padrões de fraude vão sendo descobertos sem exigir que o modelo seja re-criado do zero.
- Implementar uma interface de usuário que permita que os usuários possam ver informações sobre as predições e gráficos com estatísticas que possibilitem o usuário navegar através do histórico de transações.
- Adicionar o conceito de “*human-in-the-loop*” que permita que os usuários interajam com as predições do modelo e também possam parametrizar valores limiares para definir, por exemplo, a partir de que % de risco uma transação deve ser considerada fraudulenta.
- Aplicar a solução de predição de fraudes aqui proposta em outros contextos de dados.

REFERÊNCIAS

- ANDREESCU, A.; MIRCEA, M. et al. Managing knowledge as business rules. **Informatica Economică**, v. 13, n. 4, p. 63–74, 2009.
- ASSIS, C. A. et al. A genetic programming approach for fraud detection in electronic transactions. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2014. ISBN 9781479945221.
- BADRIYAH, T.; RAHMANIAH, L.; SYARIF, I. Nearest neighbour and statistics method based for detecting fraud in auto insurance. In: . [S.l.: s.n.], 2018.
- BALASUPRAMANIAN, N.; EPHREM, B. G.; AL-BARWANI, I. S. User pattern based online fraud detection and prevention using big data analytics and self organizing maps. In: . [S.l.: s.n.], 2018. v. 2018-January.
- BAUDER, R. A.; KHOSHGOFTAAR, T. M. Medicare fraud detection using machine learning methods. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. v. 2017-December, p. 858–865. ISBN 9781538614174.
- BAUDER, R. A. et al. Predicting medical provider specialties to detect anomalous insurance claims. **2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)**, 2016.
- BEALS, M.; DELIEMA, M.; DEEVY, M. Framework for a taxonomy of fraud. 7 2015.
- BEHDAD, M.; FRENCH, T. Online learning classifiers in dynamic environments with incomplete feedback. In: . [S.l.: s.n.], 2013. p. 1786–1793. ISBN 9781479904549.
- BEHERA, T. K.; PANIGRAHI, S. Credit card fraud detection: A hybrid approach using fuzzy clustering & neural network. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2015. p. 494–499. ISBN 9781479917341.
- BHUSARI, V.; PATIL, S. Study of hidden markov model in credit card fraudulent detection. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2016. ISBN 9781467392143.
- BOTTOU, L. From machine learning to machine reasoning: An essay. **Machine Learning**, v. 94, 2014. ISSN 08856125.
- BREIMAN, L. Random forests. **Mach. Learn.**, Kluwer Academic Publishers, USA, v. 45, n. 1, p. 5–32, oct 2001. ISSN 0885-6125. Available from Internet: <<https://doi.org/10.1023/A:1010933404324>>.
- CALDEIRA, E.; BRANDAO, G.; PEREIRA, A. C. Fraud analysis and prevention in e-commerce transactions. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2014. p. 42–49. ISBN 9781479969531.
- CHANG, W. H.; CHANG, J. S. A multiple-phased modeling method to identify potential fraudsters in online auctions. In: . [S.l.: s.n.], 2010. p. 186–190. ISBN 9780769540436.

CHARLEONNAN, A. Credit card fraud detection using rus and mrn algorithms. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. p. MIT73–MIT76. ISBN 9781509041053.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 785–794. ISBN 9781450342322. Available from Internet: <<https://doi.org/10.1145/2939672.2939785>>.

CHEN, Y. J.; WU, C. H. On big data-based fraud detection method for financial statements of business groups. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. p. 986–987. ISBN 9781538606216.

GERON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. 2nd. ed. [S.l.]: O'Reilly Media, Inc., 2019. ISBN 1492032646.

GUPTA, P.; MUNDRA, A. Online in-auction fraud detection using online hybrid model. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2015. p. 901–907. ISBN 9781479988907.

GYAMFI, N. K.; ABDULAI, J. D. Bank fraud detection using support vector machine. In: . [S.l.: s.n.], 2019.

HUANG, R.; TAWFIK, H.; NAGAR, A. K. On the use of innate and adaptive parts of artificial immune systems for online fraud detection. In: . [S.l.: s.n.], 2010. p. 1669–1676. ISBN 9781424464388.

INDRAJANI et al. Recognizing debit card fraud transaction using chaid and k-nearest neighbor: Indonesian bank case. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. ISBN 9781509051304.

IYER, D. et al. Credit card fraud detection using hidden markov model. In: . [S.l.: s.n.], 2011.

JIAN, L.; RUICHENG, Y.; RONGRONG, G. Self-organizing map method for fraudulent financial data detection. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2016. p. 607–610. ISBN 9781509025350.

JIANG, C. et al. Credit card fraud detection: A novel approach using aggregation strategy and feedback mechanism. **IEEE Internet of Things Journal**, Institute of Electrical and Electronics Engineers Inc., v. 5, p. 3637–3647, 10 2018. ISSN 23274662.

KAREEM, S.; AHMAD, R. B.; SARLAN, A. B. Framework for the identification of fraudulent health insurance claims using association rule mining. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. v. 2018-January, p. 99–104. ISBN 9781538607909.

KAVITHA, M.; SURIKALA, M. Hybrid multi-level credit card fraud detection system by bagging multiple boosted trees (bmbt). In: . [S.l.: s.n.], 2018.

- KAVITHA, M.; SURIAKALA, M. Real time credit card fraud detection on huge imbalanced data using meta-classifiers. In: . [S.l.: s.n.], 2018.
- KHAN, A.; SINGH, T.; SINHAL, A. Implement credit card fraudulent detection system using observation probabilistic in hidden markov model. In: . [S.l.: s.n.], 2012. ISBN 9781467317207.
- KHO, J. R. D.; VEA, L. A. Credit card fraud detection based on transaction behavior. In: . [S.l.: s.n.], 2017. v. 2017-December. ISSN 21593450.
- LI, X. et al. Transaction fraud detection using gru-centered sandwich-structured model. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. p. 761–766. ISBN 9781538614822.
- LI, Y. et al. Research and application of random forest model in mining automobile insurance fraud. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2016. p. 1756–1761. ISBN 9781509040933.
- Li, Z. et al. Credit card fraud detection via kernel-based supervised hashing. In: **2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBD-Com/IOP/SCI)**. [S.l.: s.n.], 2018. p. 1249–1254.
- LIU, J. M. et al. A hybrid semi-supervised approach for financial fraud detection. In: . [S.l.: s.n.], 2017. v. 1.
- MAREESWARI, V.; GUNASEKARAN, G. Prevention of credit card fraud detection based on hsvm. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2016. ISBN 9781509013524.
- MATTHEWS, B. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. **Biochimica et Biophysica Acta (BBA) - Protein Structure**, v. 405, n. 2, p. 442–451, 1975. ISSN 0005-2795. Available from Internet: <<https://www.sciencedirect.com/science/article/pii/0005279575901099>>.
- MELO-ACOSTA, G. E.; DUITAMA-MUNOZ, F.; ARIAS-LONDONO, J. D. Fraud detection in big data using supervised and semi-supervised learning techniques. In: . [S.l.: s.n.], 2017.
- MONAMO, P.; MARIVATE, V.; TWALA, B. Unsupervised learning for robust bitcoin fraud detection. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2016. p. 129–134. ISBN 9781509024735.
- MONTINI, D. A. et al. A sampling diagnostics model for neural system training optimization. In: . [S.l.: s.n.], 2013. p. 510–518. ISBN 9780769549675.
- MURTAGH, F. Multilayer perceptrons for classification and regression. **Neurocomputing**, v. 2, n. 5, p. 183–197, 1991. ISSN 0925-2312. Available from Internet: <<https://www.sciencedirect.com/science/article/pii/0925231291900235>>.
- PATIL, P. S.; DHARWADKAR, N. V. Analysis of banking data using machine learning. In: . [S.l.: s.n.], 2017.

PAULA, E. L. et al. Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. p. 954–960. ISBN 9781509061662.

PENG, H.; YOU, M. The health care fraud detection using the pharmacopoeia spectrum tree and neural network analytic contribution hierarchy process. 2016.

POZZOLO, A. D. et al. Credit card fraud detection and concept-drift adaptation with delayed supervised information. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2015. v. 2015-September. ISBN 9781479919604.

PROVOST, F. J.; FAWCETT, T.; KOHAVI, R. The case against accuracy estimation for comparing induction algorithms. In: **Proceedings of the Fifteenth International Conference on Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. (ICML '98), p. 445–453. ISBN 1558605568.

RAHMAWATI, D. et al. Fraud detection on event log of bank financial credit business process using hidden markov model algorithm. In: . [S.l.: s.n.], 2017. v. 2018-January.

RANDHAWA, K. et al. Credit card fraud detection using adaboost and majority voting. **IEEE Access**, Institute of Electrical and Electronics Engineers Inc., v. 6, p. 14277–14284, 2 2018. ISSN 21693536.

RAWTE, V.; ANURADHA, G. Fraud detection in health insurance using data mining techniques. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2015. ISBN 9781479955220.

RIZKI, A. A.; SURJANDARI, I.; WAYASTI, R. A. Data mining application to detect financial fraud in indonesia's public companies. In: . [S.l.: s.n.], 2017. v. 2018-January.

ROUX, D. D. et al. Tax fraud detection for under-reporting declarations using an unsupervised machine learning approach. In: . [S.l.]: Association for Computing Machinery, 2018. p. 215–222. ISBN 9781450355520.

ROY, A. et al. Deep learning detecting fraud in credit card transactions. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. p. 129–134. ISBN 9781538663431.

ROY, R.; GEORGE, K. T. Detecting insurance claims fraud using machine learning techniques. In: . [S.l.: s.n.], 2017.

SALAZAR, A. et al. Combination of multiple detectors for credit card fraud detection. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. p. 138–143. ISBN 9781509058440.

SALAZAR, A. et al. Automatic credit card fraud detection based on non-linear signal processing. In: . [S.l.: s.n.], 2012. p. 207–212. ISBN 9781467348072. ISSN 10716572.

SANTIAGO, G. P.; PEREIRA, A. C.; HIRATA, R. A modeling approach for credit card fraud detection in electronic payment services. In: . [S.l.]: Association for Computing Machinery, 2015. v. 13-17-April-2015, p. 2328–2331. ISBN 9781450331968.

SEO, J.; MENDELEVITCH, O. Identifying frauds and anomalies in medicare-b dataset. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. p. 3664–3667. ISBN 9781509028092. ISSN 1557170X.

SHERLY, K. K.; NEDUNCHEZHIAN, R. Boat adaptive credit card fraud detection system. In: . [S.l.: s.n.], 2010. p. 503–509. ISBN 9781424459674.

SOHONY, I.; PRATAP, R.; NAMBIAR, U. Ensemble learning for credit card fraud detection. In: **Proceedings of the ACM India Joint International Conference on Data Science and Management of Data**. New York, NY, USA: Association for Computing Machinery, 2018. (CoDS-COMAD '18), p. 289–294. ISBN 9781450363419. Available from Internet: <<https://doi.org/10.1145/3152494.3156815>>.

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. **Information Processing Management**, v. 45, n. 4, p. 427–437, 2009. ISSN 0306-4573. Available from Internet: <<https://www.sciencedirect.com/science/article/pii/S0306457309000259>>.

SRIVASTAVA, A. et al. Credit card fraud detection using hidden markov model. **IEEE Transactions on Dependable and Secure Computing**, v. 5, n. 1, p. 37–48, 2008.

TAHA, A. A.; MALEBARY, S. J. An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. **IEEE Access**, v. 8, p. 25579–25587, 2020.

VERMA, A.; TANEJA, A.; ARORA, A. Fraud detection and frequent pattern matching in insurance claims using data mining techniques. In: . [S.l.: s.n.], 2018. v. 2018-January.

Wang, H. et al. An ensemble learning framework for credit card fraud detection based on training set partitioning and clustering. In: **2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)**. [S.l.: s.n.], 2018. p. 94–98.

WANG, X.; WU, H.; YI, Z. Research on bank anti-fraud model based on k-means and hidden markov model. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. p. 780–784. ISBN 9781538649916.

XU, W. et al. Random rough subspace based neural network ensemble for insurance fraud detection. In: . [S.l.: s.n.], 2011. p. 1276–1280. ISBN 9780769543352.

XUAN, S. et al. Random forest for credit card fraud detection. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. p. 1–6. ISBN 9781538650530.

YARAM, S. Machine learning algorithms for document clustering and fraud detection. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. ISBN 9781509012800.

ZAKARYAZAD, A.; DUMAN, E. A profit-driven artificial neural network (ann) with applications to fraud detection and direct marketing. **Neurocomputing**, v. 175, p. 121–131, 2016. ISSN 0925-2312. Available from Internet: <<https://www.sciencedirect.com/science/article/pii/S0925231215015015>>.

ZEAGER, M. F. et al. Adversarial learning in credit card fraud detection. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2017. p. 112–116. ISBN 9781538618486.

ZHENG, L. et al. A new credit card fraud detecting method based on behavior certificate. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. p. 1–6. ISBN 9781538650530.

ZHENG, M. et al. Fraudne: A joint embedding approach for fraud detection. In: . [S.l.: s.n.], 2018. v. 2018-July.

ZHU, S.; WANG, Y.; WU, Y. Health care fraud detection using nonnegative matrix factorization. In: . [S.l.: s.n.], 2011. p. 499–503. ISBN 9781424497188.

ZHU, X. et al. Intelligent financial fraud detection practices in post-pandemic era. **The Innovation**, 2021.

ÖZÇELİK, M. H. et al. Improving a credit card fraud detection system using genetic algorithm. In: . [S.l.: s.n.], 2010. p. 436–440. ISBN 9781424475773.

**APÊNDICE A — LISTA COMPLETA DE ARTIGOS DE PROPOSTA DE
ABORDAGEM COBERTOS NA SLR**

Tabela A.1: Trabalhos relacionados - Lista de artigos propondo uma abordagem centrada em Instance-Based learning

Referência	Abordagem	Técnica	Tipo de Fraude
(IYER et al., 2011)	Clustering	K-Means	Cartão de Crédito
(SALAZAR et al., 2012)	Embedded	non-Gaussian mixture	Cartão de Crédito
(ZEAGER et al., 2017)	Embedded	Gaussian mixture	Cartão de Crédito
(SALAZAR et al., 2017)	Embedded	non-Gaussian mixture	Cartão de Crédito
(INDRAJANI et al., 2017)	Clustering	kNN	Cartão de Débito (banco)
(BEHDAD; FRENCH, 2013)	Embedded	XCSR	Cartão de Crédito
(BADRIYAH; RAHMANIAH; SYARIF, 2018)	Clustering	kNN	Pedido de Crédito de Seguro
(ZHU; WANG; WU, 2011)	Embedded	Fatoração de matriz não-negativa	Seguros de carros
(SEO; MENDELEVITCH, 2017)	Clustering	K-Means	Seguro saúde
(PAULA et al., 2017)	Outliers	kNN	Lavagem de Dinheiro
(MONAMO; MARIVATE; TWALA, 2016)	Clustering	trimmed K-Means	Bitcoin frude
(LIU et al., 2017)	Clustering	CBiForest	Fraude financeira
(ROUX et al., 2018)	Clustering	K-Means	Fraude em impostos
(CHEN; WU, 2017)	Embedded	kNN	Corporativo (livros fiscais)

Tabela A.2: Trabalhos relacionados - Lista de artigos propondo uma abordagem centrada em Logical Base learning

Referência	Abordagem	Técnica	Tipo de Fraude
(SOHONY; PRATAP; NAMBIAR, 2018)	Embedded	Random Forest (+NN)	Cartão de Crédito
(XUAN et al., 2018)	Logical base	Parallel Random Forest	Cartão de Crédito
(KAVITHA; SURIAKALA, 2018b)	Logical base	Tree Based Meta Classifier	Cartão de Crédito
(KAVITHA; SURIAKALA, 2018a)	Embedded	Bagging Multiple Boosted Trees	Cartão de Crédito
(SHERLY; NEDUNCHEZHIAN, 2010)	Embedded	BOAT	Cartão de Crédito
(Wang et al., 2018)	Embedded	Random Forest	Cartão de Crédito
(MELO-ACOSTA; DUITAMA-MUNOZ; ARIAS-LONDONO, 2017)	High-Scalability	Balanced Random Forest	Cartão de Crédito
(POZZOLO et al., 2015)	Embedded	FDSs	Cartão de Crédito
(KHO; VEA, 2017)	Logical base	Random Tree + J48	Cartão de Crédito
(ROY; GEORGE, 2017)	Embedded	Decision Tree + Random Forest	Pedido de Crédito Seguro
(LI et al., 2016)	Logical base	Random Forest	Seguro de Carro
(YARAM, 2017)	Embedded	Random Forest (+NB)	Fraude de Seguros
(BAUDER; KHOSHGOFTAAR, 2017)	Multiple techniques	Random Forest	Seguro Saúde
(CHANG; CHANG, 2010)	Logical base	Decision trees	e-Commerce (loopholes)

Tabela A.3: Trabalhos relacionados - Lista de artigos propondo uma abordagem centrada em Neural Networks

Referência	Abordagem	Técnica	Tipo de Fraude
(ROY et al., 2018)	Neural Networks	Recurrent Neural Networks (RNN)	Cartão de Crédito
(CALDEIRA; BRANDAO; PEREIRA, 2014)	Multiple	Neural Networks	Cartão de Crédito
(Li et al., 2018)	Multiple	Artificial Neural Networks (ANN)	Cartão de Crédito
(BEHERA; PANIGRAHI, 2015)	Embedded	Core: Neural Network learner	Cartão de Crédito
(BALASUPRAMANIAN; EPHREM; AL-BARWANI, 2018)	NN Unsupervised	Self-Organizing Map	Transações bancárias
(PATIL; DHARWADKAR, 2017)	Neural Networks	ANN	Transações bancárias eletrônicas
(MONTINI et al., 2013)	Neural Networks	Multilayer Perceptron	Transações Bancárias
(LI et al., 2018)	Embedded	deep sequential learning model	e-Commerce fraud
(ZHENG et al., 2018b)	Embedded	FraudNE	e-Commerce fraud
(HUANG; TAWFIK; NAGAR, 2010)	NN Unsupervised	Growing Hierarchical Self-Organizing Map	Fraude corporativa
(JIAN; RUICHENG; RONGRONG, 2016)	NN Unsupervised	SOM	Fraude financeira corporativa
(PENG; YOU, 2016)	Neural Networks	Neural Network Analytic Contribution Hierarchy Process	Fraude de Seguros
(XU et al., 2011)	Embedded	Random rough subspace-based neural network	Seguro de carro

Tabela A.4: Trabalhos relacionados - Lista de artigos propondo uma abordagem centrada em Support Vector Machine

Referência	Abordagem	Técnica	Tipo de Fraude
(SANTIAGO; PEREIRA; HIRATA, 2015)	Learning	SVM	Cartão de Crédito
(MAREESWARI; GUNASEKARAN, 2016)	Learning	Híbrido Support Vector Machine	Cartão de Crédito
(GYAMFI; ABDULAI, 2019)	Learning	SVM + Spark	Cartão de Crédito
(RAWTE; ANURADHA, 2015)	Embedded	SVM (+cluster)	Fraude no seguro de saúde
(KAREEM; AHMAD; SARLAN, 2018)	Learning	SVM	Fraude no seguro de saúde
(RIZKI; SURJANDARI; WAYASTI, 2017)	Embedded	SVM (+ANN)	Fraude corporativa

Tabela A.5: Trabalhos relacionados - Lista de artigos propondo uma abordagem centrada em Statistical learning

Referência	Abordagem	Técnica	Tipo de Fraude
(RANDHAWA et al., 2018)	Learning	AdaBoost	Cartão de Crédito
(CHARLEONNAN, 2017)	Embedded	Naïve Bayes (+MLP e RBF)	Cartão de Crédito
(JIANG et al., 2018)	Embedded	Logistic Regression with raw data (+RF e HMM)	Cartão de Crédito
(BAUDER et al., 2016)	Learning	Naïve Bayes	Fraude no seguro de saúde
(VERMA; TANEJA; ARORA, 2018)	Embedded	Statistical Decision Tree (+cluster)	Fraude no seguro de saúde

Tabela A.6: Trabalhos relacionados - Lista de artigos propondo uma abordagem centrada em Hidden Markov Models e outros sistemas de decisão (*reasoning*)

Referência	Abordagem	Técnica	Tipo de Fraude
(KHAN; SINGH; SINHAL, 2012)	Reasoning	Hidden Markov Models	Cartão de Crédito
(BHUSARI; PATIL, 2016)	Reasoning	Hidden Markov Models	Cartão de Crédito
(ZHENG et al., 2018a)	Reasoning	Hidden Markov Models	Cartão de Crédito
(RAHMAWATI et al., 2017)	Reasoning	Hidden Markov Models	Fraude bancária
(WANG; WU; YI, 2018)	Embedded	HMM (+k-Means)	Fraude bancária
(GUPTA; MUNDRA, 2015)	Embedded	Hidden Markov Models	Fraude em sistema de lances
(ASSIS et al., 2014)	Reasoning	Genetic Programming	Cartão de crédito
(ÖZÇELİK et al., 2010)	Reasoning	Genetic Programming	Cartão de crédito

APÊNDICE B — TAXONOMIA DETALHADA

Figura B.1: Taxonomia detalhada de técnicas emergentes



Fonte: Elaborado pelo autor