Detecção de Fraudes no Cartão de Crédito via Análise de Componentes Principais

Vilmar Mário Oro Boff

Universidade Federal do Rio Grande do Sul

Escola de Administração

10 de dezembro de 2020

Resumo

A tecnologia e os meios de pagamento evoluem constantemente desde a invenção da moeda na China, em 1100 a.C.. Mais recentemente, dentre os meios de pagamento digitais, emerge o cartão de crédito, um frequente alvo de operações criminosas para manipular o sistema financeiro. Via Análise de Componentes Principais (PCA), o presente relatório visa obter um modelo estatístico capaz de detectar fraudes em transações no cartão de crédito, classificando as observações através de um Escore de Reconstrução e de Anomalia. Como resultados, verifica-se - através das medidas Precision e Recall e Area Under the Precision-Recall Curve - que o modelo identifica fraudes satisfatoriamente.

1 Problema

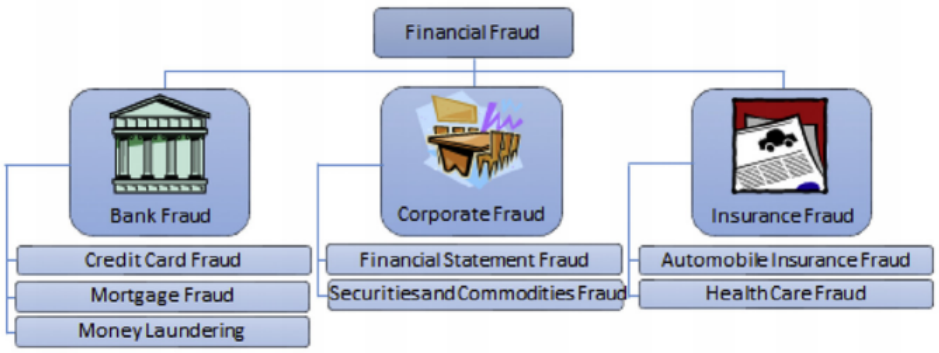
O sistema financeiro em geral sofre com a ocorrência de fraudes nos mais diversos cenários. Elas podem estar ligadas a bancos, corporações administrativas ou seguradoras e causar prejuízos nos negócios e na sociedade [2], [5], [7]. Para nosso propósito, atenderá por fraude financeira os métodos ou práticas ilegais visando obter ganhos financeiros [6], restringindo-se a outros potenciais significados.

Dentre os tipos de fraude financeira (dipostos na Figura 1), vem à tona as fraudes num mecanismo desenhado para fazer transações sem a necessidade de um meio físico de pagamento, o cartão de crédito. Dessa forma, essa modalidade oferece mais praticidade, podendo se tornar um alvo para potenciais fraudadores.

Num âmbito expandido, esse tipo de crime nos meios de pagamentos afeta não só a vítima, mas o sistema econômico por inteiro, reduzindo sua credibilidade. Um retrato disso é o fato de que as fraudes nos meios de pagamentos eletrônicos causaram prejuízos crescentes de US$ 7,6 bi em 2010, US$ 21,81 bi em 2015 e estima-se que sejam US$ 31,67 bi em 2020 [7].

Por outro lado, essa quantias subtraídas representam uma pequena parcela do montante transacionado. Segundo CRUZ; GARCIA [2] , para cada dólar gasto com cartão de crédito, 0,05 centavos são fraude. Corroborando essa hipótese, os artigos mais recentes chegaram a

Figura 1: Tipos de Fraude



Fonte: WEST, Jarod [6].

percentuais de 0.2%, 0.8%, 0.5% e 0.4% de fraudes nos *log*s (registros de eventos relevantes) das bases de dados desse serviço bancário [1].

Além disso, alguns obstáculos são comuns nesse tipo de problema, como o desbalanceamento entre as classes de fraude e não fraude, das quais a última representa 99% das observações. Uma implicação disso vai ser a mudança nas métricas de análise, uma vez que a acurácia sempre estará perto de 99%. Isso gera a necessidade de dar mais atenção para as taxas de erro (falsos positivos e falsos negativos) relativas ao total de fraudes, que geralmente contam por menos de 1% dos dados.

Outros empecilhos podem ser o *overlapping* de classes, uma vez que os criminosos tentam imitar as condutas - e por consequência, as variáveis - dos não-fraudadores. Ademais, as várias dimensões da base também podem representar alguma dificuldade ao processar os dados.

Mediante ao exposto, várias discussões e bases de dados estão disponíveis na internet visando a melhoria dos métodos. Em virtude de uma dessas discussões [4], o presente trabalho visa replicar (semelhantemente) uma das análises usando bases abertas na internet, agregando o conhecimento de otimização quadrática.

2 Modelagem

A modelagem do problema será baseada na busca por anomalias na base de dados. Dessa forma, através de uma explicação na base de dados, reconstrói-se a variável e determina-se se ela é anomalia ou não, ou seja, se é fraude ou não. A classificação é analisada com base em métricas já consolidadas da área.

Mais especificamente, o processo a ser executado é o seguinte:

1. Gerar uma explicação dos dados ;
2. Reconstruir as variáveis em função de 1 ;
3. Testar as reconstruções de 2 num Escore de Anomalia;
4. Avaliar a Classificação feita.

2.1 Explicação dos Dados: PCA

Para o item 1, vamos executar a Análise de Componentes Principais (ou Principal Component Analysis, PCA) para interpretar os dados. Dessa forma, dadas as n variáveis da base, extrai-se a matriz de Covariância A($n \times n$).

A fim de obter a maior explicação possível da variância, busca-se obter um vetor $x \in \mathbb{R}^n$ de norma 1. Ou seja, o problema passa a ser

$$\begin{aligned} &\text{Maximizar} && x^T Ax && (1)\\ &\text{sujeito a} && x^T x = \|x\| = 1 && (2) \end{aligned}$$

2.1.1 Prova da Otimalidade

Como **Prova da Otimalidade**, pode-se citar o teorema que indica a relação entre o valor máximo e mínimo da função objetivo e o maior e menor autovalores da função, respectivamente:

Teorema 2.1. *Se A é simétrica e tem autovalores $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ e autovetores normalizados associados $P_1, P_2, \ldots, P_n$, então:*

$$\lambda_1 \leq \frac{x^T Ax}{\|x\|} \leq \lambda_n$$

e ainda, para $x \neq 0$:

$$\lambda_1 = \min \frac{x^T Ax}{\|x\|} = \frac{P_1^T AP_1}{\|P_1\|} \quad e \quad \lambda_n = \max \frac{x^T Ax}{\|x\|} = \frac{P_n^T AP_n}{\|P_n\|}$$

A prova do teorema é omitida. Mas para a segunda afirmação do teorema, é facil ver que cada autovetor P_i aplicado na função objetivo resultará no autovalor λ_i:

$$\frac{P_i^T AP_i}{\|P_i\|} = \lambda_i \frac{P_i^T P_i}{\|P_i\|} = \frac{\|P_i\|}{\|P_i\|} = \lambda_i$$

Esses valores P_i são chamados de *componentes principais*. Vale lembrar também que como A é simétrica, seus n autovalores são reais, possíveis de ordenamento.

2.2 Escore de Reconstrução e de Anomalia

Outro fato benéfico vem da mudança de $y = P^T x \implies x = Py$ (pois $P^T P = I$), onde x é uma observação da amostra e P é a matriz dada pelas colunas $P_1, \ldots, P_n$ dos n autovetores de A. Dessa forma, tem-se

$$x^T Ax = (Py)^T APy = y^T P^T APy = y^T Dy,$$

onde $P^T AP = D$. D é a matriz diagonal cujos elementos são os autovalores de A. Sendo assim, se

$$D = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

então

$$x^T Ax = y^T Dy = \lambda_1 y_1^2 + \lambda_2 y_2^2 + \cdots + \lambda_n y_n^2.$$

Isso fornece uma ferramenta útil na redução de dimensionalidade, pois podemos considerar D_k as primeiras k diagonais de D e as k primeiras componentes principais $P_1, \dots, P_k$ e obter $y_k^T D_k y_k$ na tentativa de computar $x^T Ax$ com as com um certo erro. Esse erro é definido como **Escore de Reconstrução**.

Esse Escore de Reconstrução será usado para definir se uma variável é anomalia ou não, dando origem ao **Escore de Anomalia**, dado por

$$EA = \begin{cases} 0 & \text{se } |ER| \leq t \\ 1 & \text{se } |ER| > t \end{cases}$$

onde:

EA: é o Escore de Anomalia;

ER: é o Escore de Reconstrução;

t: é um parâmetro que define o limiar entre fraudes e não-fraudes.

Dessa forma, se $EA = 1$, classifica-se como fraude e se $EA = 0$, atribui-se como não-fraude.

2.3 Métricas de Análise

Para analisar o quão boa foi a classificação dada pelo Escore de Anomalia, pode-se introduzir as medidas *Precision* e *Recall*, definidas por:

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

onde TP são os verdadeiros positivos da amostra, FN são os falsos negativos e FP são os falsos positivos. Essas medidas são úteis uma vez que os conjuntos de dados acerca de fraudes tende a ter muitos verdadeiros negativos (99%), item que não entra nos cálculos de Precision e Recall.

Sendo assim, para um t fixo, calcula-se o Precision e o Recall gerando um ponto pertencente ao $\mathbb{R}^2$. Iterando sobre t, tem-se o gráfico dada pela curva gerada por esses pontos, chamado de **Precision-Recall Curve**. Um ponto interessante é que quanto mais próximo de 1 as duas medidas, melhor a classificação, implicando que a área abaixo da Precision-Recall Curve (Area Under the Precision-Recall Curve (**AUPRC**)) sugere um grau de análise e será considerado na mensuração dos resultados.

3 Dados

As informações a serem analisados são abertos e retirados do site Kaggle.com [3], um fórum mundialmente conhecido por suas competições no desenvolvimento de algoritmos na área de *Machine Learning*. Foi selecionado uma base de dados acerca de transações europeias no cartão de crédito durante dois dias do mês de setembro de 2013, totalizando 284 807 transações no cartão de crédito, dentre as quais 492 são fraudulentas.

Dessa forma, as fraudes representam 0,172% de todas as observações, evidenciando o desbalanceamento entre as classes (fraudes e não fraudes). Uma amostra (de parte da tabela) dos dados estão na Figura 2.

Figura 2: Amostra dos Dados

V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0,1269	0,5172	-0,035	-0,465	0,3202	0,0445	0,1778	0,2611	-0,143	0	1
2,1023	0,6617	0,4355	1,376	-0,294	0,2798	-0,145	-0,253	0,0358	529	1
-0,43	-0,294	-0,932	0,1727	-0,087	-0,156	-0,543	0,0396	-0,153	239,93	1
-0,172	0,5736	0,177	-0,436	-0,054	0,2524	-0,657	-0,827	0,8496	59	1
0,0091	-0,379	-0,704	-0,657	-1,633	1,4889	0,5668	-0,01	0,1468	1	1
0,4884	0,3645	-0,608	-0,54	0,1289	1,4885	0,508	0,7358	0,5136	1	1
0,5877	0,3705	-0,577	-0,67	-0,76	1,6051	0,5407	0,737	0,4967	1	1
0,2698	0,1566	-0,652	-0,552	-0,717	1,4157	0,5553	0,5305	0,4045	1	1
0,3883	0,2088	-0,512	-0,584	-0,22	1,4748	0,4912	0,5189	0,4025	1	1
0,5046	0,5897	0,1095	0,601	-0,365	-1,843	0,3519	0,5945	0,0994	1	1
0,4227	0,5512	-0,01	0,7217	0,4732	-1,959	0,3195	0,6005	0,1293	1	1
0,2041	0,3433	-0,054	0,7097	-0,372	-2,032	0,3668	0,3952	0,0202	1	1
0,316	0,5015	-0,547	-0,077	-0,426	0,1236	0,322	0,264	0,1328	1	1
0,3641	0,454	-0,578	0,046	0,4617	0,0441	0,3057	0,531	0,2437	1	1
0,4833	0,375	0,1454	0,2406	-0,235	-1,005	0,4358	0,6183	0,1485	1	1
0,5767	0,6156	-0,406	-0,737	-0,28	1,1068	0,3239	0,8948	0,5695	1	1
0,6327	0,5369	-0,546	-0,605	-0,264	1,5399	0,5236	0,891	0,5727	1	1
0,5627	0,7433	0,064	0,6778	0,083	-1,911	0,3222	0,6209	0,185	1	1
0,5496	0,7561	0,1402	0,6654	0,1315	-1,908	0,3348	0,7485	0,1754	1	1
0,5008	0,6451	-0,504	-5E-04	0,0717	0,092	0,3085	0,5526	0,299	1	1
0,3609	0,6679	-0,516	-0,012	0,0706	0,0585	0,3049	0,418	0,2089	1	1
-0,039	0,4818	0,146	0,117	-0,218	-0,139	-0,424	-1,002	0,8908	1,1	1
0,3991	0,7348	-0,436	-0,385	-0,286	1,0079	0,4132	0,2803	0,3039	1	1
0,4087	0,7167	-0,448	-0,402	-0,289	1,0118	0,426	0,4131	0,3082	1	1
-3,043	-1,052	0,2048	-2,119	0,1703	-0,394	0,2964	1,9859	-0,9	1809,7	1
1,4938	1,6465	-0,278	-0,665	-1,165	1,7018	0,6908	2,1197	1,1089	1	1
0,4404	0,1499	-0,602	-0,614	-0,403	1,5684	0,5219	0,5279	0,4119	1	1

Fonte: Kaggle.com .

A base conta com 31 colunas associadas as variáveis. A primeira representa o tempo em segundos desde a primeira transação e as duas últimas variáveis são a classe e a quantidade transacionada. As outras 28 são as componentes principais de um PCA realizado nos dados originais nomeados de V1, V2, ..., V28, visando manter a anonimidade de dados sensíveis.

4 Resultados (2 pág.)

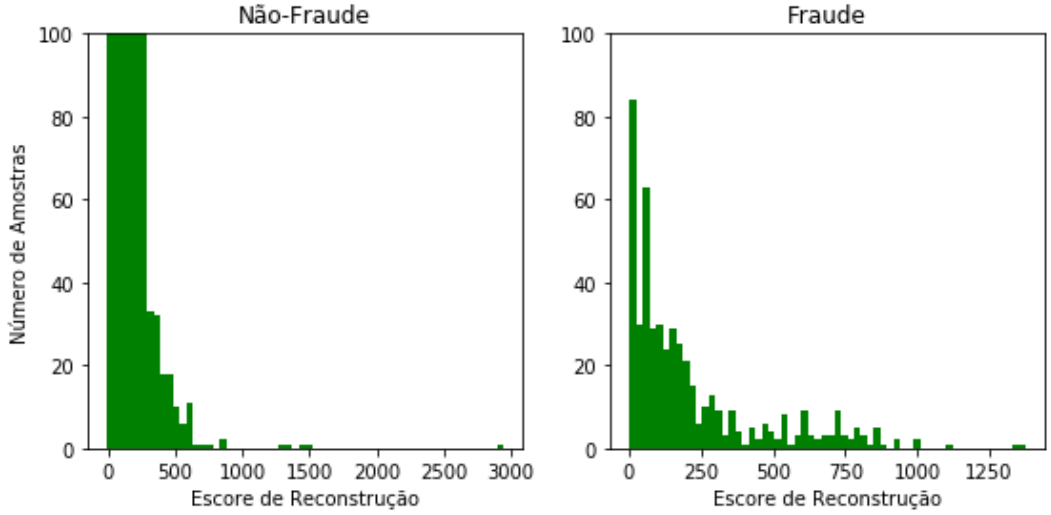
Nessa seção serão expostos os resultados obtidos. A subseção 4.1 descreve como se comporta o escore de reconstrução e a última subseção (4.2) descreve o quão boa é a classificação.

4.1 Escore de Reconstrução

Após uma breve descrição dos dados na seçao anterior, faz-se o processo de análise de componentes principais (PCA) nas variáveis V1, V2, ..., V28. São selecionadas 10 componentes principais, representando 64,7% da variância total da base dos dados.

Reconstruindo as 284 807 variáveis com base nas 10 componentes principais, temos o seguinte histograma do Escore de Reconstrução:

Figura 3: Histograma do Escore de Reconstrução com 10 Componentes Pricipais



Fonte: autoria própria.

Como forma de ilustração, os dois gráficos foram diferenciados baseados na classificação real deles. No gráfico, o número de amostras está restringido até 100, porém 281 081 (98,67%) das observações (fraudes e não-fraudes) têm escore entre -6,082 e 43,097. Para se ter uma ideia da magnitude dessa concentração, o Escore de Reconstrução tem uma média de 9,186 e desvio padrão de 21,763.

4.2 Classificação

Passando para as medidas de avaliação, o diagnóstico da classificação será dado pelas métricas de Precision e Recall, dado um limiar t no Escore de Reconstrução. Considerando μ a média do Escore de Reconstrução e σ seu desvio padrão, a tabela abaixo exemplifica os resultados para vários valores de t:

Tabela 1: Precision e Recall com 10 Componentes Principais

t	**Precision**	**Recall**
μ	0,074	0,821
$\mu + 2\sigma$	0,122	0,740
$\mu + 5\sigma$	0,216	0,520
300	0,467	0,230
500	0,683	0,144
700	0,808	0,077

Retomando os conceitos, os coeficientes perfeitos de precision e recall seriam 1 em ambos, mas podemos ver o que significa aumentos ou decréscimos em cada um dos dois.

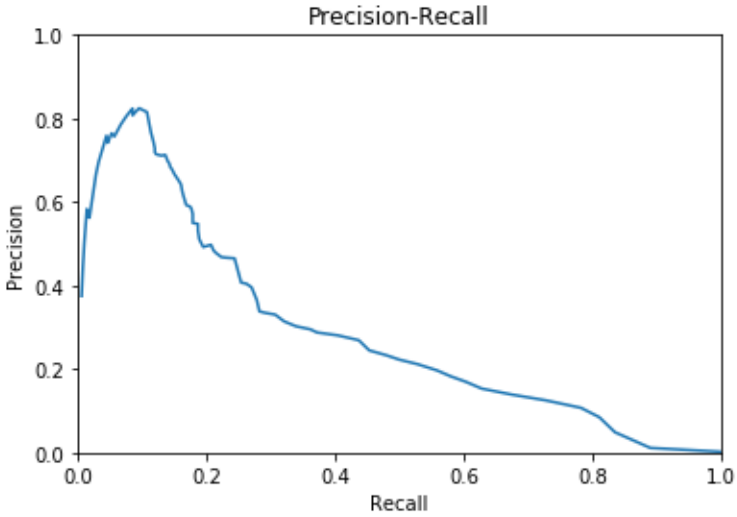
Quanto maior a métrica precision, menos os testes erram com falsos positivos comparados aos verdadeiros positivos, ou seja, menos se atribui positivo a resultados falsos e com isso menos o classificador erra quando aponta fraudes em transações normais. Relacionado a métrica Recall, temos que quanto mais próxima de 1, menos o algoritmo atribui negativo a resultados verdadeiros, ou seja, menos ele aponta para transações não-fraudulentas quando há uma fraude

ocorrendo.

Ou seja, da tabela acima, quando t aumenta, o precision decai e o recall sobe. Isso indica que quanto maior o t, menos indicações falsas de fraude, mas por outro lado deixa passar batido mais fraudes.

Esses conceitos podem ser resumidos pela Precision-Recall Curve:

Figura 4: Precision-Recall Curve com 10 Componentes Principais



Fonte: autoria própria.

Visto que o ponto (1,1) do gráfico seria o ideal, a métrica que resume conjuntamente o quão boa são precision e recall é a área abaixo desse gráfico, que na instância em questão tem valor de 0,521. Essa observação conclui a exibição dos resultados do trabalho proposto, mostrando que a metodologia de detecção de anomalias apresenta uma solução relevante para a classificação de fraudes no cartão de crédito.

A Código (1 pág.)

```
# Casa do Projeto
import os
path = r'C:\Users\vilma\Desktop\UFRGS (upload 12-09-20)\LISTAS PPGA\2020-2\PNL
Trabalho NLP'
os.chdir(path)

import pandas as pd
import numpy as np
kaggle = r'.\Bases de Dados\creditcard.csv'
df = pd.read_csv(kaggle,header=0)

# ============================================================================
# PCA
# ============================================================================
```

```
15
16 import numpy.linalg as la                    # Importa pacote numpy.linalg
17 A = np.cov(np.transpose(df.iloc[:,1:29])) # Matriz A de Covariancia
18 lamb,P=la.eig(A)                           # extrai lambda autovalores e P autove
19
20 # Ordena  o decrescente dos autovalores:
21 ind=np.argsort(lamb)[::-1]
22 lamb=lamb[ind]
23 P=P[ind]
24
25 D = np.diag(lamb)
26
27 # Porcentagem  da vari ncia explicada por cada autovalor
28 assoc a seu autovetor:
29 evr=lamb/np.sum(lamb)
30
31
32 # ============================================================================
33 # Reconstruir o PCA / Escore de reconstru  o
34 # ============================================================================
35 def reconstruir(escore,i,x, n_componentes):
36     # y = (P^(-1))*x   &    P^T = P^-1
37     y = np.transpose(P[:,0:n_componentes]).dot(x)
38
39     # xx- yy = xAx - yDy:
40     yy = np.transpose(y).dot(D[0:n_componentes,0:n_componentes]).dot(y)
41     xx = np.transpose(x).dot(A).dot(x)
42     escore[i] = xx-yy
43     return(escore)
44
45
46
47 n_componentes = 10 ## DEFINA AQUI QUANTAS COMPONENTES QUER USAR
48
49 var = np.transpose(df.iloc[:,1:-2])
50 escore = np.zeros((var.shape[1]))
51 for i in var.columns:
52      reconstruir(escore,i, var[i], n_componentes)
53
54
55 import matplotlib.pyplot as plt
56 fig, (ax2, ax1) = plt.subplots(1, 2, figsize=(9, 4))
```

```
ax1.hist(escore[[df['Class']==1]],60, color='g')
ax1.set_ylim(0, 100)
ax1.set_title(f'Fraude')
ax1.set_xlabel('Escore de Reconstru  o')

ax2.hist(escore[[df['Class']==0]],60, color='g')
ax2.set_ylim(0, 100)
ax2.set_title(f'N o-Fraude')
ax2.set_xlabel('Escore de Reconstru  o')
ax2.set_ylabel('N mero de Amostras')
plt.show()

# ============================================================================
# M tricas
# ============================================================================

dx = 100
treshold = np.linspace(min(escore),1000,dx)
# treshold = [np.mean(escore) + np.std(escore), np.mean(escore) + 2*np.std(esc
precision = np.zeros(len(treshold))
recall = np.zeros(len(treshold))

for i in range(0,len(treshold)):
    previsao = (abs(escore) > treshold[i])+0
    dif = np.array(df['Class']) + np.transpose(4*previsao) #*4 pra diferenciar
    t_p = np.sum(dif == 5)
    f_p = np.sum(dif == 4)
    t_n = np.sum(dif == 0)
    f_n = np.sum(dif == 1)
    #              Realidade
    # F V           V   F
    # 0 1  Prev V   5   4
    # 0 4        F  1   0
    precision[i]=t_p/(t_p+f_p)
    recall[i] = t_p/(t_p+f_n)
conf = pd.DataFrame([[t_p,f_p],[f_n,t_n]],
                    index = ['V_Pred','F_Pred'], columns= ['V','F'])


import matplotlib.pyplot as plt
plt.figure(figsize=(6, 4))
plt.title('Precision-Recall')
```

```
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.xlim(0, 1)
plt.ylim(0, 1)
plt.plot(recall, precision)
print('computed AUPRC using np.trapz: {}'.format(np.trapz(precision,
                                                          dx = 1/dx)))
```

Referências

[1] Carcillo, F., Le Borgne, Y.-A., Caelen, O., and Bontempi, G. Streaming active learning strategies for real-life credit card fraud detection: assessment and visualization. *International Journal of Data Science and Analytics 5*, 4 (2018), 285–300.

[2] Cruz Quispe, L. M., and Rantes García, M. T. Detección de fraudes usando técnicas de clustering.

[3] Kaggle. Credit card fraud detection. https://www.kaggle.com/mlg-ulb/creditcardfraud.

[4] Meazzini, L. Fraud detection - unsupervised anomaly detection, Junho 2020. https://towardsdatascience.com/fraud-detection-unsupervised-anomaly-detection-df43d81fce67.

[5] Oliveira, P. H. M. A. *Detecção de fraudes em cartões: um classificador baseado em regras de associação e regressão logística*. PhD thesis, Universidade de São Paulo, 2016.

[6] West, J. Intelligent financial fraud detection: A comprehensive review. *Computers and Security 57* (2016), 47 – 66.

[7] Zamini, M., and Montazer, G. Credit card fraud detection using autoencoder based clustering. In *2018 9th International Symposium on Telecommunications (IST)* (2018), IEEE, pp. 486–491.