Detecção de Fraudes em Compras com Cartão de Crédito Utilizando Redes Neurais Artificiais

Gabriel G. dos Santos, Lucas Shin-Iti Aoki, Lucca Giovane Gomes, Jose Artur Lima Passini

[1]Departamento de Computação – Universidade Estadual de Londrina (UEL)
Londrina – PR – Brazil

Abstract. *This article explores the utilization of deep neural networks (DNN) and convolutional neural networks (CNN) to detect credit card fraud. The study leverages transactional data from the European population throughout the year 2023, comprising over 550,000 records with anonymized cardholder information. Initially, a simple single-layer deep neural network is developed, and its performance is compared with a more efficient method, a convolutional neural network, typically utilized in image recognition.*

Resumo. *Este artigo explora o uso de Redes Neurais profundas (DNN) e redes neurais convolucionais (CNN) para identificar fraudes de cartão de crédito. O estudo utiliza dados transacionais do cartão de crédito da população européia ao longo do ano de 2023, incluindo mais de 550.000 registros cujos dados de titularidade foram modificados para manter anonimato. Inicialmente é criado uma rede neural profunda simples de uma camada, para então ser comparada com o método mais eficiente, uma rede neural convolucional que é normalmente utilizado em reconhecimento de imagens.*

1. Introdução

A detecção de fraudes em transações com cartão de crédito é um desafio contínuo e crucial para instituições financeiras e consumidores em todo o mundo. Com o aumento constante das transações financeiras eletrônicas, as fraudes também têm se tornado mais sofisticadas e difíceis de detectar utilizando métodos tradicionais. Nesse contexto, a implementação de algoritmos de aprendizado de máquina, particularmente redes neurais, emergiu como uma solução promissora para lidar com esse problema.

O objetivo deste artigo é explorar o uso de redes neurais para detecção de fraudes em transações com cartão de crédito. Para isso, será utilizado um conjunto de dados contendo informações detalhadas e anonimizadas sobre transações financeiras, incluindo características como valor da transação, localização, tipo de estabelecimento, entre outros. A partir desses dados, nosso objetivo é desenvolver e avaliar um modelo de rede neural capaz de identificar padrões e anomalias que possam indicar atividades fraudulentas.

A importância desse trabalho reside na proteção dos consumidores e das instituições financeiras contra perdas financeiras significativas decorrentes de fraudes em transações com cartão de crédito. Além disso, a detecção eficaz de fraudes pode contribuir para a construção de um ambiente financeiro mais seguro e confiável, aumentando a confiança dos consumidores no uso de serviços financeiros digitais.

Ao longo deste artigo, apresentaremos a metodologia utilizada para o desenvolvimento e avaliação do modelo de detecção de fraudes, bem como os resultados obtidos a partir da aplicação desse modelo ao conjunto de dados disponível.

2. Sobre o Conjunto de dados

Este conjunto de dados abrange transações efetuadas por portadores de cartões de crédito europeus durante o ano de 2023. Compreende mais de 550.000 registros, sendo os dados anonimizados para preservar a identidade dos portadores dos cartões. O propósito principal deste conjunto de dados é facilitar o desenvolvimento de algoritmos e modelos destinados à detecção de fraudes, visando identificar transações potencialmente fraudulentas[Delamaire et al. 2009].

Os dados foram obtidos a partir de transações realizadas por portadores de cartões de crédito europeus em 2023 através da plataforma Kaggle, foi realizada a remoção de informações sensíveis para assegurar a privacidade e estar em conformidade com as diretrizes éticas.

3. Tratamento, e análise exploratória dos dados

O conjunto de dados apresenta uma variedade de campos, sendo os mais proeminentes os campos de V1 a V28. Esses campos foram anonimizados e representam uma diversidade de informações sensíveis associadas a cada transação. Embora os detalhes exatos dessas informações não sejam aparentes, eles são dados de identificação, localização, horário, tipo de estabelecimento e outros atributos relevantes para transações com cartão de crédito. Essa técnica de anonimização é crucial para preservar a privacidade dos indivíduos envolvidos nas transações.

Adicionalmente aos campos anonimizados, o conjunto de dados contém o campo ”Amount”, que expressa o valor da transação, e o campo ”Class”, um rótulo binário indicando se a transação é considerada fraudulenta (1) ou não (0). Esses campos se mostram

fundamentais para a análise e a elaboração de modelos de detecção de fraudes, permitindo uma abordagem mais precisa e eficaz na identificação de atividades fraudulentas.

Com base na análise inicial dos dados, podemos concluir que o conjunto de dados está limpo e pronto para análise. As principais observações incluem:

- Ausência de valores faltantes: Não foram identificados valores ausentes em nenhum dos registros do conjunto de dados, indicando que todas as informações necessárias estão disponíveis para análise.
- Tipos de dados adequados: Todos os tipos de dados parecem estar em conformidade com o esperado, o que significa que não há necessidade de conversão ou ajuste dos tipos de dados para análise.
- Inexistência de duplicatas: Não foram encontradas duplicatas nos dados, o que sugere que cada registro é único e representa uma transação distinta.
- Equilíbrio da variável alvo (Class): A variável alvo, que indica se uma transação é fraudulenta ou não, está bem balanceada, o que significa que há uma distribuição adequada de exemplos positivos e negativos.

A partir dessas conclusões, podemos afirmar que o conjunto de dados está em uma condição adequada para prosseguirmos com análises mais aprofundadas e a construção de modelos preditivos. Não é necessário realizar mais etapas de limpeza de dados neste momento, o que nos permite avançar diretamente para a etapa de modelagem e análise.

3.1. Correlação entre os dados

Utilizamos um mapa de calor de correlação para visualizar as relações entre as variáveis do nosso conjunto de dados. Esta ferramenta nos permite identificar padrões, detectar multicolinearidades e selecionar variáveis relevantes para análises mais detalhadas e construção de modelos preditivos mais precisos. Ao analisar o mapa de calor de correlação, destacam-se diversas observações importantes:

- Correlações Positivas Significativas: As variáveis V16, V17 e V18 apresentam correlações positivas fortes entre si, indicando uma associação significativa entre esses atributos. Da mesma forma, as variáveis V9 e V10 também demonstram correlações positivas fortes, sugerindo uma relação substancial entre elas.
- Correlações Negativas Significativas: São observadas correlações negativas fortes entre V4 e V14, V4 e V12, V4 e V10, V10 e V11, V11 e V14, V11 e V12, bem como entre V21 e V22. Isso sugere uma associação inversa entre esses conjuntos de variáveis.
- Ausência de Correlações Positivas na Faixa de V19 a V28: Nota-se uma clara ausência de correlações positivas na faixa de V19 a V28, indicando uma baixa associação entre essas variáveis.
- Correlações Moderadas: Entre as variáveis V1 a V18, são identificadas várias correlações positivas e negativas moderadas, sugerindo uma associação razoável entre esses atributos.

Essas observações proporcionam insights valiosos sobre as relações entre as variáveis do conjunto de dados, contribuindo para a compreensão da estrutura subjacente e a identificação de padrões relevantes para análise e modelagem.

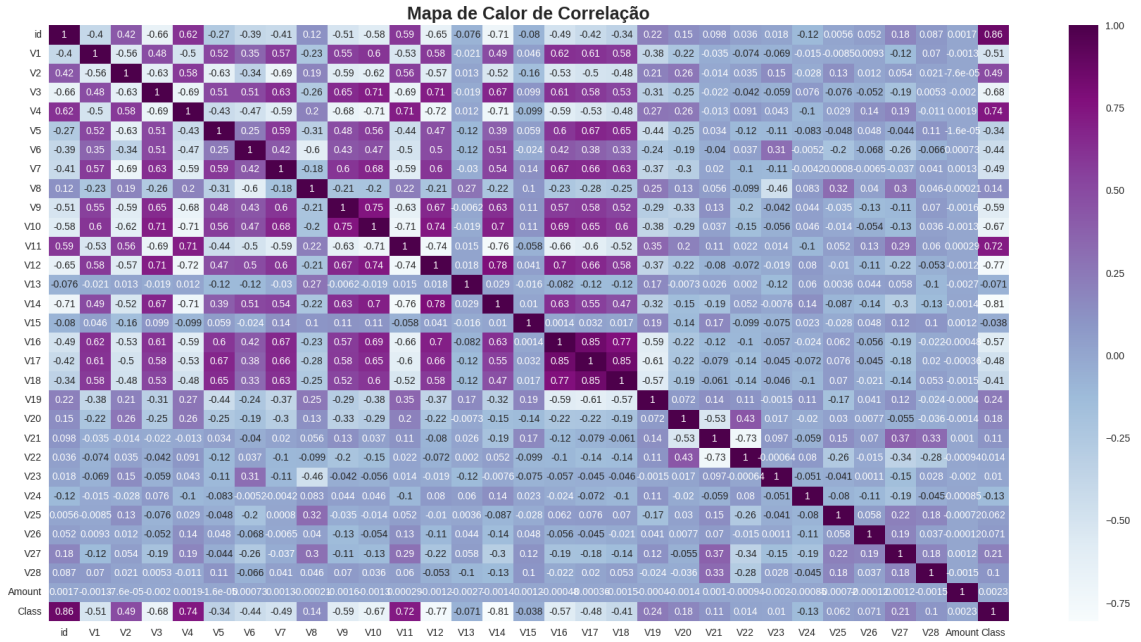


Figure 1. Mapa de Calor de Correlação

3.2. Normalização e divisão do conjunto de dados

A normalização dos dados é uma etapa crucial no pré-processamento, especialmente ao lidar com algoritmos sensíveis à escala das variáveis[Dutka and Hansen 1991]. Utilizamos o StandardScaler, uma técnica comum que padroniza as variáveis para ter média zero e desvio padrão unitário. Isso garante que todas as características tenham a mesma escala, evitando que variáveis com escalas maiores dominem o processo de treinamento do modelo.

Dividimos o conjunto de dados em dois conjuntos distintos: um conjunto de treino e um conjunto de teste. Esta divisão é crucial para avaliar o desempenho do modelo de forma imparcial. Utilizamos a função train_test_split para dividir os dados, onde especificamos o tamanho do conjunto de teste (20% dos dados neste caso) e também aplicamos uma estratificação com base na variável alvo (classe) para garantir uma distribuição semelhante entre os conjuntos de treino e teste. Além disso, configuramos o parâmetro ”shuffle=True” para embaralhar os dados antes da divisão, o que ajuda a evitar possíveis viéses na seleção dos conjuntos.

4. Primeira Abordagem Utilizando Regressão Logística

Para a primeira tentativa de teste, optamos por utilizar a regressão logística para o treinamento do modelo. A escolha deste modelo se deu pelo fato de ser uma excelente opção para problemas de classificação binária, como é o caso. A regressão logística é robusta, rápida de implementar e interpretar, além de ser capaz de lidar com conjuntos de dados de grande escala. Sua capacidade de fornecer probabilidades de classificação torna-a uma escolha ideal para identificar transações fraudulentas com base em características específicas das transações[LaValley 2008]. Portanto, a regressão logística foi selecionada como nosso modelo inicial.

4.1. Resultados da regressão logística

Obtivemos os seguintes resultados após realizar o treinamento no conjunto de dados utilizando a regresão logística.

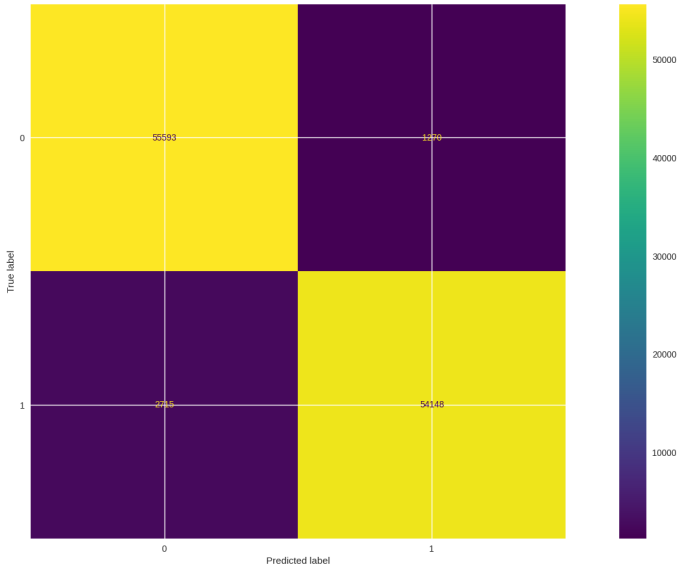


Figure 2. Matriz de Confusão Regressão Logística

	precision	recall	f1-score	support
0	0.95	0.98	0.97	56863
1	0.98	0.95	0.96	56863
accuracy			0.96	113726
macro avg	0.97	0.96	0.96	113726
weighted avg	0.97	0.96	0.96	113726

Figure 3. Métricas Do Modelo Regressão Logistica

5. Rede Neural Artificial de Uma Camada

Para nossa primeira, utilizamos uma Rede Neural Artificial (ANN) simples com uma camada oculta. As ANNs são conhecidas por sua capacidade de modelar relações complexas e não lineares nos dados, tornando-as adequadas para a detecção de fraudes, onde os padrões fraudulentos podem não ser lineares.[Caliskan et al. 2018]

5.1. Arquitetura da Rede Neural

A rede neural foi implementada utilizando a biblioteca Keras, com a seguinte arquitetura:

- Uma camada densa (fully connected com um número de neurônios igual ao número de características de entrada, utilizando a função de ativação ReLu (Rectified Linear Unit).
- Uma camada de saída com um único neurônio, utilizando a função de ativação sigmoid, que é apropriada para problemas de classificação binária.

A função de custo utilizada foi a *binary crossentropy*, e o otimizador escolhido foi o Adam, conhecido por sua eficiência e adaptabilidade em problemas de aprendizado profundo.

```
def createModelSequential(neurons):
    model = Sequential()
    model.add(Dense(neurons, activation='relu'))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(optimizer='adam',
    loss='binary_crossentropy', metrics=['accuracy'])
    return model
```

O modelo foi treinado por 5 épocas com um tamanho de lote (batch size) de 1024, e 30% dos dados de treinamento foram reservados para validação durante o treinamento.

5.2. Resultados da Rede Neural Artificial

Após o treinamento, a rede neural foi avaliada no conjunto de dados de teste. Os resultados demonstraram uma melhoria em comparação com a regressão logística, indicando que a rede neural conseguiu capturar relações mais complexas nos dados.

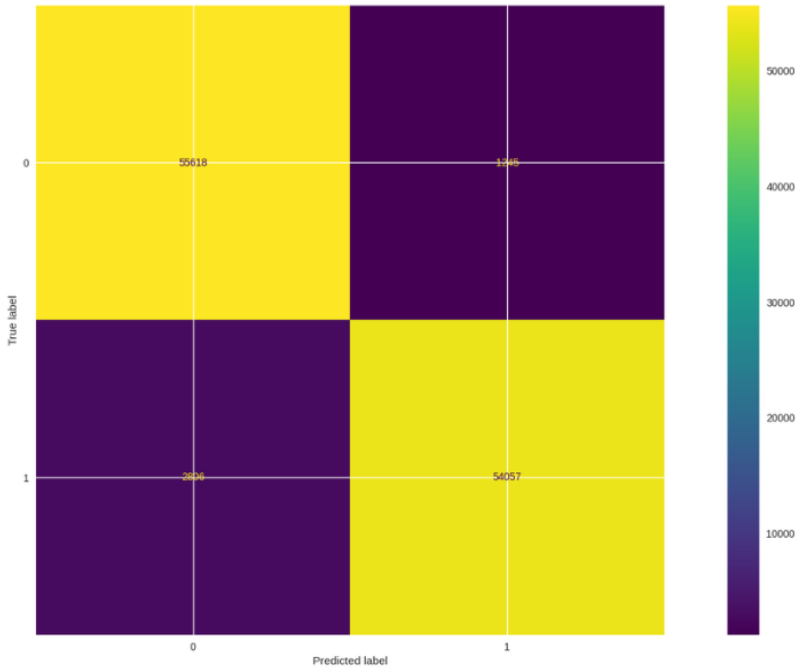


Figure 4. Matriz de Confusão Rede Neural ANN

tempo de treinamento: 8.47 segundos

	precision	recall	f1-score	support
0	0.95	0.98	0.96	56863
1	0.98	0.95	0.96	56863
accuracy			0.96	113726
macro avg	0.96	0.96	0.96	113726
weighted avg	0.96	0.96	0.96	113726

Figure 5. Métricas do Modelo Rede Neural ANN

6. Redes Neurais Convolucionais

Redes Neurais Convolucionais (CNNs) é um tipo de redes neurais artificiais projetadas para processar dados com estrutura de grades ou complexas, como séries temporais ou dados bidimensionais (como imagens). O diferencial das CNN, são as camadas de convolução, esta é formada por vários núcleos de convolução que são usados para calcular diferentes mapas de características. Cada neurônio de um mapa de características está conectado a uma região de neurônios vizinhos na camada anterior. Essa região é chamada de campo receptivo do neurônio na camada anterior. O novo mapa de características pode ser obtido primeiro convolvendo a entrada com um núcleo aprendido e depois aplicando uma função de ativação não linear elemento por elemento nos resultados da convolução.[O'shea and Nash 2015]

6.1. CNN no contexto do problema

Os dados de transação são complexos em natureza, existem muitas colunas para verificação de autenticidade. Além disso, são dados temporais que devem ser analisados com peso considerativo no parâmetro de tempo. As CNN's tem vantagem neste aspecto em relação a outras redes neurais profundas, dados sequenciais, complexos e temporais podem ser facilmente categorizados e analisados com o algorítimo de convolução.

Para criar uma CNN, precisamos definir alguns parâmetros para as camadas ocultas:

```
    def createCnnModel(input_length):
  model = keras.Sequential([
        Conv1D(filters=5,
               kernel_size=10,
               activation="relu",
               input_shape=(input_length, 1),
               strides=2),
        MaxPooling1D(pool_size=2),
        Flatten(),
        Dense(32, activation="relu"),
        Dense(1, activation="sigmoid")
    ])
  model.compile(optimizer='adam',
   loss='binary_crossentropy',
   metrics=['accuracy'])
  return model
```

Note alguns parâmetros essenciais que irão definir o desempenho da CNN[Albawi et al. 2017]:

- filters: Este parâmetro define o número de filtros (ou kernels) que serão aplicados na camada de convolução. Cada filtro é uma janela que desliza sobre a entrada para extrair características.
- kernel_size: Indica o tamanho da janela que será usada na convolução. No contexto de uma CNN de uma dimensão, isso representa o comprimento da janela que desliza sobre a entrada para calcular as convoluções.

- activation: Este parâmetro define a função de ativação que será aplicada após a convolução. No caso apresentado, é ”relu” (Unidade Linear Retificada), que é uma função comumente usada que retorna zero para valores negativos e o próprio valor para valores positivos.
- strides: Este parâmetro define o passo ou deslocamento do kernel ao se mover através da entrada durante a convolução. Um valor de 2 significa que o kernel será movido de 2 em 2 posições.
- max pooling: Esta operação reduz a dimensionalidade dos mapas de características, mantendo apenas os valores máximos em uma região específica. pool_size indica o tamanho da janela sobre a qual a operação de pooling será aplicada.

6.2. Desempenho da CNN

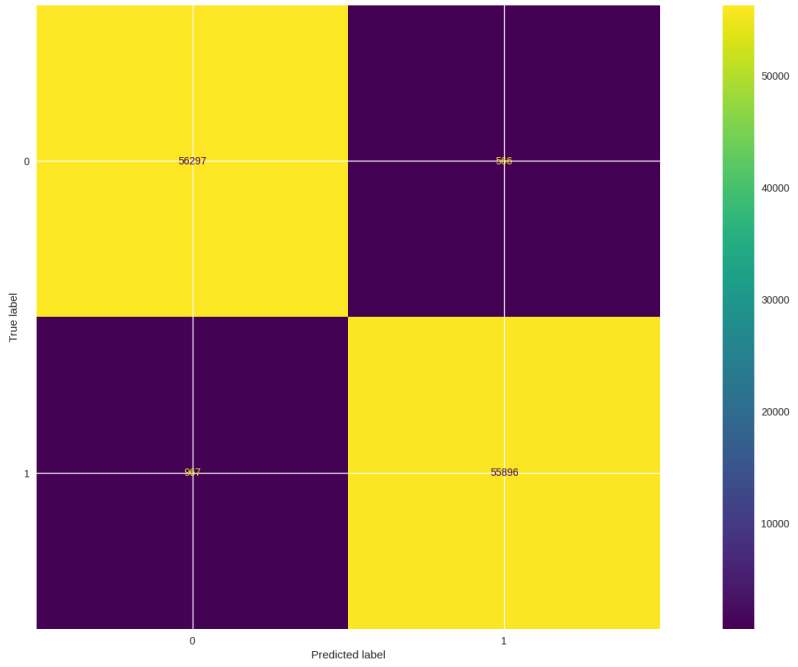
```
3554/3554 [==============================] - 73s 36ms/step
              precision    recall  f1-score   support

           0       0.98      0.99      0.99     56863
           1       0.99      0.98      0.99     56863

    accuracy                           0.99    113726
   macro avg       0.99      0.99      0.99    113726
weighted avg       0.99      0.99      0.99    113726
```

Podemos notar que o modelo CNN demorou significativamente para mostrar resultados em comparação com a DNN de uma camada e o método por regressão logística.

Já em questão da matriz de confusão, o modelo CNN conseguiu reduzir o numero de falso negativos, deixando relativamente equilibrada a taxa de erro, e com porcentagem de precisão alta.



7. Conclusão

Em resumo, através do estudo realizado em relação a detecção de fraudes em transações de cartões usando redes neurais. Obtivemos diferença gerada nos de resultados quando se compara a uma regressão logística e uma rede neural convolucional (CNN).

Com isso podemos observar por meio das nossas descobertas ao usarmos o dataset mencionado acima que existe uma melhoria significativa em relação a qualidade da avaliação dos dados quando se usa uma rede neural, com uma redução de resultados falsos positivos e falsos negativos comparado o que e gerado pela regressão logística.

Apesar dos resultados promissores que a rede neural demostra, algumas limitações devem ser citadas, sendo a principal dela o custo computacional, assim necessitando maquinas com uma potencia alta para que consiga obter os resultados em um bom tempo.

References

Albawi, S., Mohammed, T. A., and Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee.

Caliskan, A., Yuksel, M. E., Badem, H., and Basturk, A. (2018). Performance improvement of deep neural network classifiers by a simple training strategy. *Engineering Applications of Artificial Intelligence*, 67:14–23.

Delamaire, L., Abdou, H., and Pointon, J. (2009). Credit card fraud and detection techniques: a review. *Banks and Bank systems*, 4(2).

Dutka, A. F. and Hansen, H. H. (1991). *Fundamentals of data normalization*. Addison-Wesley Longman Publishing Co., Inc.

LaValley, M. P. (2008). Logistic regression. *Circulation*, 117(18):2395–2399.

O'shea, K. and Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.