UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

BRENO MOURA DE ABREU E THOMAZ HUGO SUZUKI PEREIRA

DETECÇÃO DE FRAUDES EM LICITAÇÕES PÚBLICAS ATRAVÉS DA IDENTIFICAÇÃO DE ANOMALIAS NOS VALORES COBRADOS E DA ANÁLISE DE REDES COMPLEXAS FORMADAS POR INTERAÇÕES DE COMPRA

CURITIBA

2023

BRENO MOURA DE ABREU E THOMAZ HUGO SUZUKI PEREIRA

DETECÇÃO DE FRAUDES EM LICITAÇÕES PÚBLICAS ATRAVÉS DA IDENTIFICAÇÃO DE ANOMALIAS NOS VALORES COBRADOS E DA ANÁLISE DE REDES COMPLEXAS FORMADAS POR INTERAÇÕES DE COMPRA

Fraud Detection in Public Biddings Through the Identification of Anomalies in the Charged Values and the Analysis of Complex Networks Formed by Purchase Interactions

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Sistemas de Informação do Curso de Bacharelado em Sistemas de Informação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Luiz Celso Gomes Jr.

CURITIBA

2023

BRENO MOURA DE ABREU E THOMAZ HUGO SUZUKI PEREIRA

DETECÇÃO DE FRAUDES EM LICITAÇÕES PÚBLICAS ATRAVÉS DA IDENTIFICAÇÃO DE ANOMALIAS NOS VALORES COBRADOS E DA ANÁLISE DE REDES COMPLEXAS FORMADAS POR INTERAÇÕES DE COMPRA

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Sistemas de Informação do Curso de Bacharelado em Sistemas de Informação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 05/Dezembro/2023

Prof. Dr. Luiz Celso Gomes Jr.
Doutor em Ciência da Computação
Universidade Tecnológica Federal do Paraná

Prof. Dr. João Alberto Fabro
Doutor em Engenharia Elétrica e Informática Industrial
Universidade Tecnológica Federal do Paraná

Profa. Dra. Leyza Baldo Dorini
Doutora em Ciência da Computação
Universidade Tecnológica Federal do Paraná

CURITIBA
2023

RESUMO

A detecção de anomalias em conjuntos de dados é de grande relevância em diversos cenários atuais, principalmente na área contábil onde anomalias podem ser consideradas indícios de fraude. Este tipo de análise pode ser usado para mitigar riscos e evitar perdas financeiras, sobretudo em órgãos públicos. Este trabalho tem como objetivo a detecção de anomalias em dados de notas fiscais de compras do setor público brasileiro de forma autônoma, para ajudar e acelerar o trabalho de auditores na área. Foram exploradas duas frentes distintas: (i) a detecção em dados estruturados utilizando o *Local Outlier Factor* (LOF), *Isolation Forest* (iForest) e *Self-Organizing Maps* (SOM); e (ii) a detecção em redes complexas utilizando *Generative Adversarial Attributed Network Anomaly Detection* (GAAN) e *Contrastive self-supervised Learning framework for Anomaly detection on attributed networks* (CoLA). No caso da detecção em dados estruturados (i), o iForest demonstrou ser o método mais promissor para a detecção de fraudes enquanto que o LOF apresentou resultados insatisfatórios; o SOM se mostrou mais eficaz na detecção de *outliers* isolados, tornando impossível identificar casos de fraudes. No caso da detecção de anomalias em redes complexas (ii), o modelo CoLA demonstrou um resultado mais favorável na identificação de nós irregulares, onde foi possível observar diferenças entre o nó apontado pelo modelo e nós semelhantes, sendo isso uma indicação que o nó é realmente uma instância anômala. Já o modelo GAAN identificou como nós anômalos instâncias bastante isoladas, ou seja, nós com poucas relações ligadas a ele (arestas) e poucos nós semelhantes, dificultando bastante a análise do resultado, não sendo possível identificar se de fato é uma instância possivelmente anômala ou uma instância normal.

Palavras-chave: aprendizado de máquina; detecção de anomalias; dados estruturados; redes complexas; fraude em licitação.

ABSTRACT

Anomaly detection in datasets is of great relevance to multiple current scenarios, particularly in the accounting field where anomalies can be considered signs of fraud. This type of analysis can be used to mitigate risks and avoid financial losses, especially in the public sector. The goal of this project is to use anomaly detection methods in invoice data from the Brazilian public sector autonomously, in order to help and speed up the work of financial auditors. Two distinct fronts were explored: (i) anomaly detection in structured data using Local Outlier Factor (LOF), Isolation Forest (iForest) and Self-Organizing Maps (SOM); and (ii) anomaly detection in complex networks using Generative Adversarial Attributed Network Anomaly Detection (GAAN) and Contrastive self-supervised Learning framework for Anomaly detection on attributed networks (CoLA). The results with higher anomaly scores were analyzed manually by the developers and, after comparing them with similar entries, were categorized according to their probability of being fraudulent instances. For the detection in structured data (i), the iForest proved to be the most promising method for detecting frauds, while the LOF showed unsatisfactory results; the SOM method proved to be more effective in detecting isolated outliers, making it impossible to identify fraud cases. For the detection in complex networks (ii), the CoLA model showed better results in the identification of irregular nodes where it was possible to observe differences between a node pointed by the model and other similar nodes, indicating that the node is, in fact, an anomaly instance. The GAAN model identified greatly isolated anomaly nodes, in other words nodes with few connections and few similar nodes, making it difficult to analyze the results and to identify if the instance is an anomaly or not.

Keywords: machine learning; outlier detection; structured data; complex networks; competitive bidding fraud.

LISTA DE FIGURAS

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

De acordo com a Lei Federal número 8.666/93 Art. 90, uma fraude em licitação é definida como:

> Frustrar ou fraudar, mediante ajuste, combinação ou qualquer outro expediente, o caráter competitivo do procedimento licitatório, com o intuito de obter, para si ou para outrem, vantagem decorrente da adjudicação do objeto da licitação (BRASIL, 1993)

Dessa forma, qualquer ação que utilize intencionalmente de manipulação da contabilidade com o objetivo de burlar os procedimentos de licitação favorecendo entidades específicas e assim prejudicando a Administração Pública, é considerada uma fraude em licitação (GOMES, 2017).

Um levantamento feito pelo Ministério Público Federal a partir do cruzamento de dados do Ministério do Desenvolvimento Social, Tribunal Superior Eleitoral, Tribunal de Contas e Receita Federal foi capaz de identificar mais de 1 milhão de casos de fraude no programa Bolsa Família, totalizando 2,6 bilhões de reais sendo destinados indevidamente entre os anos de 2013 e 2014 (VEJA, 2023). Outro levantamento feito pela ONG Transparência Brasil detectou possíveis fraudes em contratos públicos entre fevereiro de 2020 e outubro de 2022 (período da pandemia de COVID-19), que foram firmados com empresas que não são do mesmo ramo do produto adquirido, fornecedores inscritos no Cadastro de Empresas Inidôneas e Suspensas (CEIS), empresas com faturamento suspeito e empresas abertas 30 dias ou menos antes do acordo firmado. Esses contratos suspeitos totalizam cerca de 2 bilhões de reais (OLIVEIRA, 2022), e ao ano o Instituto Ética Saúde estima um prejuízo anual de 22,54 bilhões de reais na área da saúde (PUENTE; AMEIDA, 2021), que se melhor geridos, poderiam ter salvo diversas vidas.

Dentre as várias formas de se fraudar uma licitação pública destaca-se o sobrepreço que, de acordo com a Lei Federal 14.133/21 Art. 6, ocorre quando o valor de uma mercadoria ou serviço é “expressivamente superior aos preços referenciais de mercado”, podendo ser considerada para o valor unitário ou global do objeto (BRASIL, 2021). Pode-se ainda estabelecer o ato de superfaturamento por superdimensionamento que tem o objetivo de inflar o custo total da mercadoria, aumentando desnecessariamente a quantidade de produtos de forma a resultar no sobrepreço do contrato (LOPES *et al.*, 2021).

A detecção de fraudes e sua prevenção é uma responsabilidade do setor administrativo da entidade, portanto, a atividade de auditoria se faz necessária para a identificação de fraudes e erros contábeis (SCHWINDT; CORAZZA, 2008). Para identificar padrões e detectar possíveis fraudes e corrupção, o Tribunal de Contas da União criou em 2017 a Secretaria de Relações Institucionais de Controle no Combate à Fraude e Corrupção (Seccor), aprimorando as ferramentas utilizadas e aumentando os esforços de auditoria em processos com indícios de corrupção. Porém, a integração de informações entre órgãos da união é precária, fazendo com

que esses esforços envolvam muitos processos manuais na busca por indícios, provas e, muitas vezes, dependentes de denúncias, culminando em uma porcentagem pequena de agentes públicos e empresas mal-intencionadas envolvidos em fraudes que são levados a julgamento. Entretanto, há a possibilidade de otimizar este processo através do uso de técnicas de aprendizado de máquina para detecção de *outliers* (HAMELERS, 2021) que possibilitam a identificação automática de anomalias nas bases de dados.

Através da análise dos registros de notas fiscais gravados em uma base de dados estruturados é possível identificar instâncias que apresentam anormalidades que, após análise posterior por um auditor, podem ser identificadas como registros fraudulentos. Identificam-se três principais desafios para a detecção de *outliers* para este cenário: (i) geralmente não há a rotulação dos registros em notas que apresentam fraude e que não apresentam; (ii) os dados são provenientes de diversos contextos, dessa forma o que pode ser considerado uma anomalia em um contexto não seria em outro; e (iii) deve ser possível indicar a razão pelo qual um registro foi identificado como anômalo, ou seja, explicar quais atributos foram analisados e como a categorização foi realizada.

A resolução do problema de detecção de anomalias em um contexto financeiro conta com diferentes abordagens como o uso de modelos de regressão, redes neurais, lógica *fuzzy*, algoritmos genéticos, redes Bayesianas e árvores de decisão (AL-HASHEDI; MAGALINGAM, 2021). Algumas estratégias promissoras fazem uso de modelos como o Deep Learning AutoEncoder (PAULA *et al.*, 2016), Isolation Forests (HAMELERS, 2021), modelos de regressão que avaliam os outliers tanto localmente quanto globalmente (LIANG; PARTHASARATHY, 2016), Self-Organizing Maps (HUANG; TSAIH; YU, 2014) e através do cálculo do Local Outlier Factor (SHAN; MURRAY; SUTINEN, 2009).

Outro indício que se pode usar para identificação de fraude é a topologia das conexões nas redes de compra e venda. Essas redes formam um grafo dinâmico, onde suas mudanças no decorrer do tempo podem ser analisadas por diferentes modelos de detecção de anomalias. Embora a literatura sobre abordagens para grafos dinâmicos ainda seja escassa, uma abordagem que se mostra promissora é a baseada em comunidade (POURHABIBI *et al.*, 2020), utilizando modelos que detectam vértices anômalos, como o ECOutlier (GUPTA *et al.*, 2012), subgrafos anômalos, como o NetSpot (MONGIOVI *et al.*, 2013), redes neurais de grafos, como o CoLA (LIU *et al.*, 2021) ou rede generativa adversária, como o GAAN (CHEN *et al.*, 2020).

O objetivo deste estudo é, a partir dos dados de notas fiscais disponibilizados em um conjunto de dados estruturados e sem a categorização dos documentos em normais e fraudulentos, empregar técnicas de aprendizado de máquina para detectar anomalias visando a aplicação de uma metodologia capaz de detectar fraudes. Para tal, duas frentes distintas serão exploradas para a detecção de *outliers*: (i) a análise dos valores cobrados e sua relação com os atributos contextuais; e (ii) a análise de redes de grafos complexas onde os vértices representam os compradores e vendedores e as arestas representam as relações de venda.

Entre os objetivos específicos deste trabalho estão:

- Identificar uma boa representação em grafos para a rede de compras; o grafo deve representar as interações entre empresas, itens licitados e outras variáveis importantes para a análise de licitações além de ter um componente temporal, representando as mudanças na rede ao longo do tempo.
- Avaliar a eficácia dos modelos desenvolvidos em termos de sua capacidade de detectar anomalias e identificar padrões suspeitos em um conjunto de dados de licitações públicas reais.

As seções subsequentes do texto estão organizadas da seguinte forma: a Seção 2 apresenta os modelos utilizados para a detecção de *outliers* mais relevantes para este trabalho, e é divido em duas subseções que apresentam os principais algoritmos para as duas frentes abordadas: (i) Detecção de Anomalias em Dados Estruturados; e (ii) Detecção de Anomalias Através da Análise de Grafos Dinâmicos. A Seção 3 descreve a metodologia que foi utilizada para a implementação das soluções para ambas as frentes. A Seção 4 apresenta a Análise Exploratória dos dados que foram disponibilizados para a execução deste trabalho. A Seção 5 mostra os resultados encontrados pelos modelos aplicados e a avaliação de sua eficácia. E a parte 6 apresenta a conclusão e os pontos de melhorias para os trabalhos futuros.

2 FUNDAMENTOS E TRABALHOS RELACIONADOS

Anomalias, também chamadas de *outliers*, são instâncias que exibem um comportamento divergente dos demais registros da amostra; ou seja, que apresentam atributos comportamentais inconformes com os esperados dado um determinado contexto. Assim, a detecção de *outliers* atua sobre um conjunto de registros onde cada indivíduo é analisado e, comparando seu comportamento com os demais, é categorizado como normal ou anômalo, obtendo uma delineação clara entre as duas classes.

Para este trabalho, serão exploradas dois conjuntos distintos de métodos para a detecção de anomalias: o primeiro atua na análise dos valores cobrados fazendo uso de modelos de aprendizado de máquina para encontrar, através da relação entre os atributos contextuais e comportamentais, quais instâncias se diferem demasiadamente das demais; o segundo tem o foco no aprendizado de máquina para ajudar na detecção de anomalias em estruturas de grafos onde estes são formados a partir de nós, representando os compradores e vendedores, e arestas, representando as relações de venda.

2.1 Detecção de Anomalias em Dados Estruturados

Para realizar a detecção de *outliers* em dados estruturados é necessário considerar três principais desafios: (i) raramente há uma diferenciação rotulada entre quais registros são normais e quais são anômalos; (ii) há múltiplos contextos diferentes presentes nas amostras, o que dificulta uma análise global dos registros, assim, cada indivíduo deve ser analisado levando em consideração o contexto ao qual pertence; e (iii) a explicabilidade do sistema é um componente fundamental para que seja possível não só apontar quais registros fogem à normalidade mas também explicar o por quê.

Para o primeiro problema (i), como as bases de dados que serão analisadas raramente apresentam a distinção entre instâncias normais e anômalas, impõe-se a necessidade de utilizar o aprendizado não-supervisionado que irá identificar registros que fogem à normalidade em relação aos outros indivíduos sem a necessidade de haver uma classificação prévia do estado de cada entrada (HAMELERS, 2021)(PAULA *et al.*, 2016). Assume-se, neste caso, que os dados que não apresentam anomalias são mais frequentes que os que *outliers*, ou seja, há uma diferença de proporção considerável entre as duas classes e só assim é possível realizar a detecção destas anomalias para o tipo de dados em questão (CHANDOLA; BANERJEE; KUMAR, 2009).

Para o segundo problema (ii) caracteriza-se um atributo contextual aquele que identifica o contexto, ou vizinhança, ao qual um determinado registro pertence (CHANDOLA; BANERJEE; KUMAR, 2009). Dessa forma, haverá diversas vizinhanças distribuídas pelo espaço de características e o que pode ser considerado uma anomalia para uma vizinhança pode não ser para

outra dificultando, assim, a detecção de *outliers*. Dessa forma, separam-se os atributos em duas categorias:

1. Atributos contextuais: normalmente são variáveis categóricas utilizadas para localizar a qual vizinhança um registro pertence.

2. Atributos comportamentais: são as variáveis que indicam o comportamento do indivíduo, ou seja, os valores de saída para um determinado contexto.

Assim, para se detectar um *outliers*, os atributos contextuais revelam um comportamento esperado e caso os atributos comportamentais sejam destoantes com os resultados esperados, aponta-se uma anomalia.

Para o terceiro problema (iii) percebe-se a necessidade de explicar o motivo pelo qual um registro foi identificado como anômalo. Isto se torna relevante para aumentar a transparência e confiabilidade do sistema identificador e garantir que o sistema não viole os direitos humanos (HAMELERS, 2021). O processo de explicabilidade indica quais foram as variáveis encontradas que divergem dos resultados esperados dado um determinado contexto, e permite compreender a razão pela qual o sistema apontou a anormalidade do registro. Após análise posterior de um analista é possível determinar se a instância refere-se apropriadamente a um caso de anomalia ou se o sistema gerou um ocorrência de falso positivo.

A resolução do problema de detecção de anomalias em dados estruturados conta com diferentes abordagens como o uso de modelos de regressão, redes neurais, lógica *fuzzy*, algoritmos genéticos, redes bayesianas, árvores de decisão e Máquinas de Vetor Suporte (Support Vector Machines - SVMs) (AL-HASHEDI; MAGALINGAM, 2021). Estes algoritmos realizam um processo inicial de aprendizado que irá analisar os registros da base de dados para encontrar quais as características classificam uma entrada como normal ou anômala. A saída pode ser um *score*, um valor contínuo que indica numericamente a conformidade do indivíduo com seu comportamento esperado e onde uma anomalia é assinalada com base num valor de limiar preestabelecido ou um valor binário categórico que indica simplesmente se a entrada pertence ou não à categoria de *outliers* (CHANDOLA; BANERJEE; KUMAR, 2009).

Nas subseções a seguir serão descritos os algoritmos Local Outlier Factor, Isolation Forest e Self-Organizing Maps, os métodos mais promissores encontrados para a detecção de fraudes em notas fiscais a partir da análise de dados estruturados.

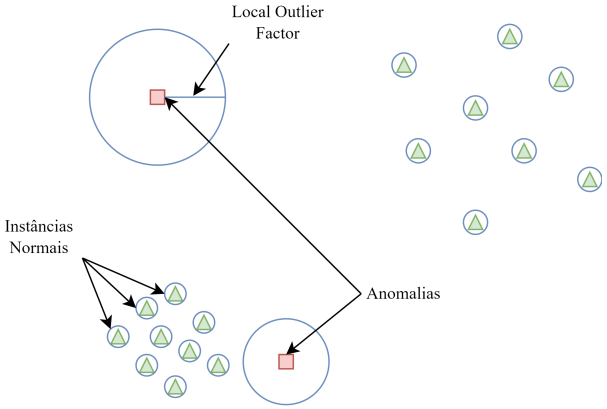
2.1.1 Local Outlier Factor

Local Outlier Factor (LOF) é uma métrica aplicada para cada ponto em um conjunto de dados que visa calcular o grau de divergência deste ponto em relação aos seus k-vizinhos mais próximos. Para tal, compara-se a densidade local de uma entrada com a densidade local dos seus vizinhos; havendo uma disparidade significativa entre os resultados em que a densidade da

entrada analisada é menor que a de seus vizinhos, pode-se classificar o registro como anômalo (BREUNIG *et al.*, 2000).

Esta métrica permite encontrar *outliers* em contextos específicos, dado que a forma em que é calculada apenas leva em consideração os vizinhos mais próximos de uma instância. Assim, é ideal para identificar anomalias a partir de *clusters* de dados que apresentam densidades diferentes entre si. O método em questão determina que quanto maior o LOF de um ponto, mais provável de ser uma anomalia. Dessa forma, calcula-se o LOF para cada instância de um conjunto de dados a partir uma quantidade predeterminada de vizinhos (k), e as instâncias com maior pontuação, dado um certo valor de contaminação, são compreendidos como *outliers*. O método também pode ser utilizado para localizar *clusters* de anomalias dependendo do valor de k utilizado.

Figura 1 – Representação da aplicação do Local Outlier Factor.



Fonte: Autoria própria (2023).

Para determinar o LOF de uma instância é necessário considerar os seguintes conceitos:

1. K-vizinhos mais próximos: o número de vizinhos mais próximos de uma instância que serão utilizados no cálculo da pontuação.

2. K-distância: a distância entre a instância e seu vizinho mais distante, considerando o valor determinado de k-vizinhos mais próximos. Um valor pequeno desta variável sugere que a instância se encontra em uma área densa; de modo contrário, um valor alto significa uma área de baixa densidade.

3. Distância de acessibilidade: é o valor mais alto entre a distância entre um ponto A e B e a k-distância do ponto B.

4. Densidade de acessibilidade local: é o cálculo da densidade da vizinhança em volta de uma determinada instância. Valores mais altos apresentam uma vizinhança mais densa.

5. *Score* do Local Outlier Factor (LOF): é a pontuação final onde a densidade de acessibilidade local de uma instância é comparada com a de seus k-vizinhos mais próximos,

possibilitando indicar se sua densidade é destoante das demais. Valores mais altos indicam uma maior probabilidade do registro ser anômalo.

2.1.2 Isolation Forests

O Isolation Forest, também chamado apenas de iForest, é um algoritmo utilizado especificamente para a detecção de *outliers* e parte do princípio que como as anomalias são escassas e divergem das instâncias normais, há uma alta probabilidade de que com a construção de uma árvore de decisão estas sejam encontradas a uma distância menor da raiz da árvore (LIU; TING; ZHOU, 2008). Uma iForest apresenta as seguintes definições:

1. *Isolation Tree*: uma árvore de isolação onde os nós-folhas representam as saídas esperadas dada a combinação dos valores de cada atributo de um registro de entrada, e os demais são nós de decisão onde a partir do valor de uma determinada dimensão, separam os dados em dois grupos; ou seja, cada nó de decisão determina para qual o seguinte nó de decisão o algoritmo irá. No final dos testes o algoritmo indicará um nó-folha com o resultado esperado.

2. Distância do Caminho: é a distância entre um nó-folha e a raiz da árvore e é medido pela quantidade de arestas entre os nós.

3. *Anomaly Score*: é um score derivado da distância do caminho utilizado para determinar se um nó é ou não um *outliers*. No caso das Isolation Forests, um valor próximo de 1 indica uma anomalia; um valor menor que 0.5 quase certamente indica uma instância normal; e valores acima de 0.6 indicam uma provável anomalia. Sendo assim, quando um conjunto de dados apresenta scores próximos de 0.5 para todas as instâncias pode-se concluir que não há anomalias significativas na base. O score é encontrado a partir do cálculo das distâncias entre um nó-folha e a raiz de múltiplas árvores de isolação, que formam uma floresta de isolação.

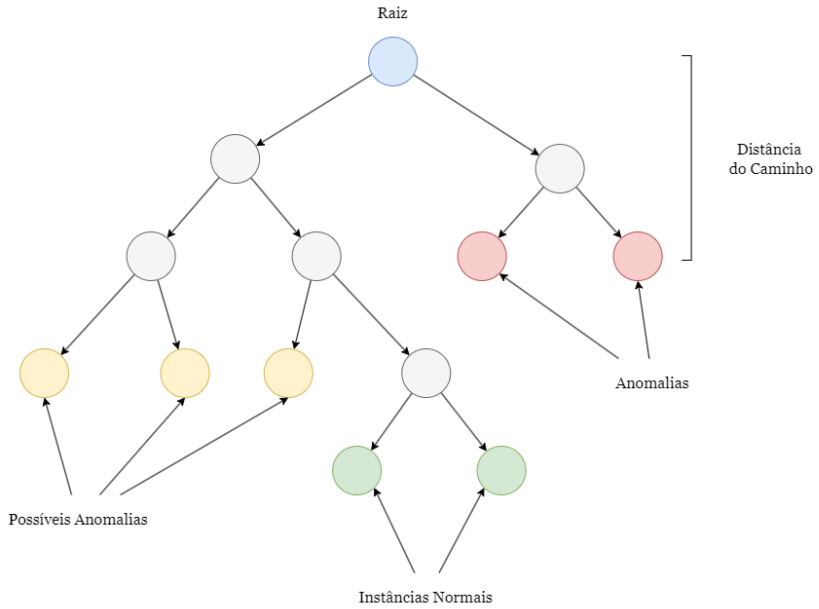
A construção de uma Isolation Forest é feita em duas etapas:

1. Etapa de Treinamento: onde uma amostra dos dados é inserida no algoritmo que recursivamente constrói múltiplas árvores até que as instâncias sejam isoladas ou cada árvore atinja uma determinada altura.

2. Etapa de Avaliação: é calculada a distância entre a raiz e cada nó-folha das múltiplas árvores e um score é calculado a partir deste valor. Os scores são ordenados de forma decrescente e os n primeiros nós da lista são indicados como as primeiras *n* anomalias.

A Figura 2 ilustra a ideia de que em uma árvore de isolamento (*isolation tree*) de uma iForest as anomalias se encontram mais próximas da raiz e a probabilidade de uma instância

Figura 2 – Representação de uma Isolation Tree e seus principais componentes.



Fonte: Autoria própria (2023).

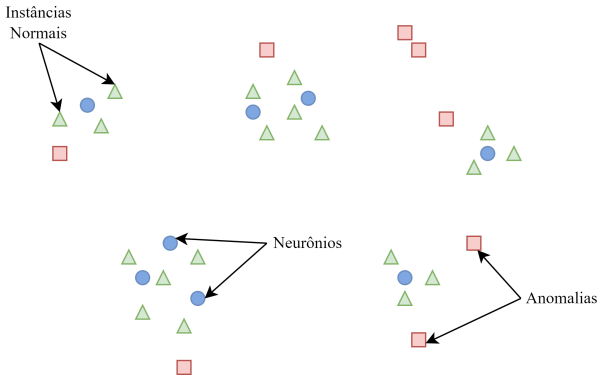
ser uma anomalia real é reduzida quanto mais distante a folha se encontra da raiz. A figura é apenas uma simplificação da estrutura de uma árvore de isolamento e na realidade uma *isolation tree* pode conter mais camadas ainda seguindo a lógica que quanto mais próximo da raiz, mais provável de ser uma anomalia.

2.1.3 Self-Organizing Maps

Self-Organizing Maps (SOM) é um algoritmo utilizado para agrupamento e redução de dimensionalidade que pode também ser utilizado para a detecção de *outliers*. Seu funcionamento permite realizar a projeção dos dados em um espaço de dimensionalidade reduzida e em que as anomalias são encontradas através da análise da distância entre uma instância e seus valores esperados. O SOM faz uso de uma matriz de neurônios, normalmente com duas ou três dimensões, que com o processo de aprendizado tendem a acompanhar a topologia dos dados originais. Ou seja, a posição espacial dos neurônios após a etapa de aprendizagem se assemelha à posição espacial das instâncias do conjunto de dados mas em um espaço com um número menor de dimensões. Com isso, é possível deduzir que anomalias são encontradas a uma distância relativamente distante dos neurônios em comparação com as instâncias normais, ou em pequenos grupos que destoam dos demais possibilitando, assim, detectar registros que possuem um comportamento atípico (BRZEZINSKA; HORYN, 2022).

O SOM é uma rede neural artificial que adapta os pesos dos neurônios de acordo com as características dos atributos em um processo chamado aprendizado competitivo. O processo pode ser resumido nos seguintes passos:

Figura 3 – Representação da estrutura dos Self-Organizing Maps.



Fonte: Autoria própria (2023).

1. Os neurônios do SOM são inicializados dado uma determinada topologia da rede que determina a quantidade de neurônios da rede e sua posição.
2. Uma instância dos dados de treinamento é escolhida aleatoriamente.
3. Calcula-se a distância entre a instância e cada neurônio, encontrando o Best Matching Unit (BMU), o neurônio mais próximo.
4. Os pesos, ou posição, do BMU e seus vizinhos mais próximos são atualizados, e são aproximados da posição da instância.
5. Os passos 2 - 4 são repetidos por n iterações.

O resultado do processo é um mapa que exibe as dimensões comprimidas dos dados e é organizado de forma a permitir a visualização das instâncias em um espaço com uma quantidade reduzida de dimensões. Os grupos podem ser determinados a partir da posição final dos neurônios já que estes são aproximados dos centros de cada *cluster* durante a execução do algoritmo; e os *outliers* podem ser encontrados observando instâncias relativamente distantes do seu BMU mais próximo ou da formação de regiões de baixa densidade de instâncias anômalas.

2.1.4 Aplicação de Detecção de *Outliers* em Dados Estruturados no Contexto de Fraudes Financeiras

O uso do cálculo do Local Outlier Factor (LOF) para detecção de anomalias demonstrou resultados positivos em um estudo de 2009 que fez uso do método para encontrar faturamentos inapropriados no contexto de saúde pública. A pesquisa concluiu que o LOF é um método efetivo para encontrar anomalias em faturamentos tendo sido validado com a ajuda de especialistas da área contábil; e seus resultados são comparáveis com a análise aprofundada dos dados realizada por especialistas (SHAN; MURRAY; SUTINEN, 2009).

Outro estudo sugere o uso do Isolation Forest, que permite isolar os *outliers* além de tornar a interpretação da detecção fácil e rápida quando comparada a outros métodos. No

contexto de detecção de fraudes financeiras em faturas, a utilização de iForests em comparação aos algoritmos Local Outlier Factor (LOF) e One Class Support Vector Machine (OCSVM) se mostra superior não somente na detecção de anomalias em si mas também na performance e interpretabilidade dos resultados (HAMELERS, 2021).

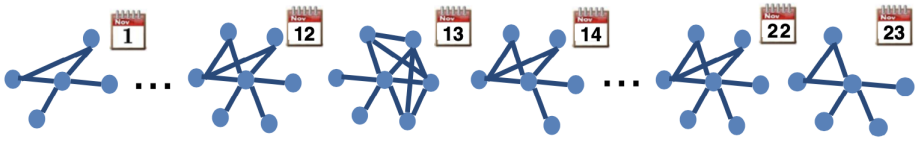
Também é possível utilizar abordagens que utilizam versões aprimoradas do SOM para permitir a detecção de registros fraudulentos com desvios mais sutis. Uma delas é a criação de um Growing Hierarchical Self-Organizing Map (GHSOM) que permite criar estruturas mais complexas de SOMs progressivamente construindo múltiplos mapas e os organizando em uma estrutura hierárquica em diferentes níveis, possibilitando lidar melhor com registros com alta dimensionalidade. Ainda, o modelo apresentado facilita a detecção de de pequenos grupos de instâncias anômalas assim como de registros que se encontram relativamente distantes de seus respectivos BMUs (HUANG; TSAIH; YU, 2014).

2.2 Detecção de Anomalias Através da Análise de Grafos Dinâmicos

A detecção de anomalias baseada em grafos vem sendo reconhecida por especialistas como uma técnica bastante promissora nos últimos anos (POURHABIBI *et al.*, 2020), devido a capacidade de entender relações complexas e se adaptar podendo ser aplicada a ambientes dinâmicos.

Grafos são uma ferramenta poderosa para a representação de interações e relacionamentos entre objetos distintos. Sistema financeiro, sensores em uma fábrica, linhas de energia em uma cidade, transações entre pessoas e interações entre usuários em redes sociais são todos exemplos de redes dinâmicas. Tais redes podem ser representados em grafos, no qual sua estrutura e atributos estão em constante mudança. As mudanças possíveis em um grafo dinâmico são a inserção ou remoção de um vértice, inserção ou remoção de aresta, e modificação nos atributos de vértices e arestas no decorrer do tempo (RANSHOUS *et al.*, 2015), como ilustrado na Figura 4.

Figura 4 – Mudanças em grafos com o decorrer do tempo.



Fonte: (AKOGLU; TONG; KOUTRA, 2015).

Na criação e desenvolvimento de um sistema de detecção de anomalias baseado em grafos, temos primeiramente que entender três características principais sobre os dados a serem utilizados: (i) disponibilidade de uma classificação de dados que diferencie uma relação normal e uma relação anômala, (ii) natureza da rede de entrada, (iii) tipos de anomalias que se quer detectar (POURHABIBI *et al.*, 2020).

Classificação dos dados (i): Temos três possibilidades de abordagem de detecção de anomalias: supervisionada, semi-supervisionada e não supervisionada. Para as técnicas supervisionadas e semi-supervisionada é necessário a classificação dos dados para o treinar o modelo. Já a abordagem não supervisionado pode ser aplicada mesmo sem essa informação.

Natureza da rede de entrada (ii): Alguns atributos da rede de entrada devem ser estudados antes do início do desenvolvimento do modelo de detecção de anomalia, pois influenciam na escolha e projeto do modelo a ser utilizado. Com a análise exploratória dos dados podemos identificar alguns atributos importantes como a propagação de informação na rede: ao analisar de onde o produto sai (vendedor) e para onde o produto vai (comprador), formando uma relação de compra e venda, e a data em que foi criada essa relação. Podemos também analisar atributos referentes aos vértices, como tipo do vértice (comprador ou vendedor), nome e endereço. Por último devemos analisar quais informações seriam importantes para as arestas que irão compor o grafo, como a data em que a relação foi criada e valor da transação.

Tipos de anomalias (iii): Diferentes modelos possuem diferentes aplicações e por isso é importante saber qual o tipo de anomalia se quer encontrar para que possa escolher o modelo que melhor atende o problema a ser resolvido. Atualmente, as anomalias mais comuns de serem detectadas em grafos, estáticos ou dinâmicos, são o vértice anômalo, subgrafo anômalo, aresta anômala e evento anômalo, existindo modelos especializados em cada um deles. (POURHABIBI *et al.*, 2020).

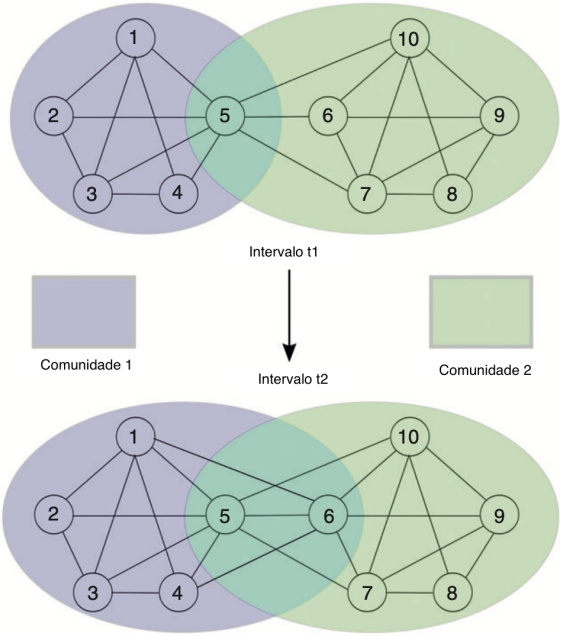
2.2.1 Métodos Baseados em Comunidades

Esse método é baseado em analisar padrões em comunidades de vértices (vértices relacionados e com comportamento semelhante), possibilitando ao modelo considerar não apenas os atributos dos vértices, mas também suas relações com outros vértices do grafo e o contexto das comunidades, detectando anomalias onde os atributos e comportamento são significativamente diferentes dos vértices semelhantes (AKOGLU; TONG; KOUTRA, 2015).

Uma pessoa vegetariana, por exemplo, no contexto de um restaurante vegano seria considerado um comportamento normal, porém a mesma pessoa em uma churrascaria seria considerado uma anomalia, como exemplo da Figura 5, onde o nó 6 (pessoa vegetariana) no tempo t1 está relacionado à comunidade 2 (restaurante vegano), porém em t2 passa a se ligar com a comunidade 1 (churrascaria), ou seja, os mesmos nós com os mesmos atributos podem ser considerados anômalos, ou não, dependendo de suas relações e o contexto onde estão inseridos.

Um modelo promissor que utiliza dessa abordagem é o ECOutilier, que busca achar padrões e anomalias ao criar uma matriz de pertencimento de cada vértice para cada comunidade encontrada. O modelo compara o comportamento do vértice em um espaço de tempo com cada comunidade para detectar uma possível anomalia. As características e particularidades do modelo ECOutlier serão melhor descritas na sessão 2.2.2.

Figura 5 – Exemplo de vértice anômalo. Em um grafo dinâmico temos o vértice 6 associado à comunidade 2 no intervalo de tempo t1, e associado as comunidades 1 e 2 no intervalo de tempo t2. Devido a essa mudança de comportamento, o vértice 6 pode ser considerado anômalo (RANSHOUS *et al.*, 2015).



Fonte: (RANSHOUS *et al.*, 2015) .

Outro modelo que segue a abordagem baseada em comunidade é o NetSpot, que diferentemente do ECOutlier, em vez de olhar para vértices individuais e sua probabilidade de pertencer a uma comunidade e seu grau de anomalia, foca em encontrar subgrafos com comportamento anômalo ao observar o comportamento das comunidades existentes no grafo com o passar do tempo (RANSHOUS *et al.*, 2015) e (MONGIOVI *et al.*, 2013). Esse modelo, similar ao ECOutlier, também faz a utilização de uma técnica de alternância, onde se fixa um subgrafo e encontra-se uma janela de tempo ideal, então fixa-se a janela de tempo e encontra-se o subgrafo ideal, repetindo esse processo até encontrar um conjunto de subgrafos altamente anômalos e suas janelas de tempo correspondentes. Uma explicação mais detalhada sobre o modelo NetSpot será desenvolvida na seção 2.2.3.

Existem também modelos baseados em redes neurais de grafos aplicados a redes atribuídas. Nesse contexto, temos como representantes o modelo CoLA, que se baseia em identificar diferenças entre nó e sub-grafo para encontrar nós anômalos (LIU *et al.*, 2021), e o modelo GAAN, que tem como princípio a geração de instâncias randômicas, e a comparação entre instâncias reais e randômicas para a detecção de nós anômalos (CHEN *et al.*, 2020). Ambos os modelos serão melhor detalhados nas seções 2.2.5 e 2.2.4, respectivamente.

2.2.2 ECOutlier: Evolutionary Community Outliers

O modelo ECOutlier supõe que a maioria dos vértices que pertencem a uma comunidade devem seguir uma tendência de evolução parecida no decorrer do tempo. Porém, para alguns vértices dessa comunidade isso pode não ser verdade - sua tendência de evolução difere bastante dos outros vértices presentes na mesma comunidade. O objetivo do modelo é detectar esses vértices anômalos, que chamaremos de ECOutliers (GUPTA *et al.*, 2012). Um ECOutlier apresenta as seguintes definições (GUPTA *et al.*, 2012):

1. Comunidade: É um grupo de vértices relacionados que possuem atributos e comportamento mais semelhantes entre si do que com outras comunidades.

2. Matriz de Pertencimento: As linhas representam os vértices, as colunas representam as comunidades encontradas no grafo e cada posição da matriz corresponde a probabilidade de cada vértice pertencer a cada comunidade.

3. Matriz de Correspondência: É a matriz que possui as informações de correspondência de comunidades entre diferentes intervalos de tempo.

4. Matriz de Anomalias: A matriz de anomalias representa o grau de anomalia para cada vértice relacionado a cada comunidade, onde as linhas representam os vértices e as colunas representam as comunidades.

5. Evolutionary Community Outlier (ECOutlier): Um vértice é considerado um ECOutlier em relação a uma comunidade se ocorrer uma grande diferença na probabilidade de pertencimento do vértice em intervalos de tempo distintos.

Primeiramente o algoritmo encontra as diferentes comunidades no grafo e cria a Matriz de Pertencimento relacionando o comportamento do vértice com cada comunidade encontrada para dois intervalos de tempo diferentes. Após a criação da Matriz de Pertencimento, o modelo cria a Matriz de Correspondência entre as diferentes comunidades, para que possa comparar a mesma comunidade em diferentes intervalos de tempo. De posse das Matrizes de Pertencimento e Matrizes de Correspondência, o modelo compara a probabilidade do vértice pertencer à comunidade nos dois intervalos de tempo escolhidos. Se o comportamento do vértice difere do comportamento esperado em relação aos outros vértices pertencentes a comunidade, é atribuído ao vértice um alto grau de anomalia, sendo ele considerado um ECOutilier.

2.2.3 NetSpot

O objetivo do modelo NetSpot é identificar um conjunto regiões anômalas, que chamaremos de SAR (*Significant Anomalous network Regions*), de um grafo dinâmico em diferentes intervalos de tempo (MONGIOVI *et al.*, 2013).

Para encontrar as SAR, o modelo NetSpot se baseia no algoritmo *very large-scale neighborhood search*, que utiliza o conceito de busca local, onde uma solução inicial é iterativamente melhorada por meio de movimentos que exploram um conjunto amplo de vizinhanças. Ao utilizar uma abordagem em busca local, é preciso atentar-se aos pontos de busca iniciais para que o modelo não fique preso em um máximo local para se ter uma boa solução final.

Para esquivar-se desse problema o modelo NetSpot, além de abranger um conjunto grande de soluções vizinhas, utiliza uma heurística própria para a geração dos pontos de busca iniciais chamadas *Seeds*, que leva em consideração o *Heaviest Dynamic Subgraph* que contenha o nó X em um tempo T e maior subsequência no tempo que contenha o mesmo nó X, para encontrar a melhor localidade no grafo e no tempo para criar os pontos de busca iniciais, fugindo dos máximos locais e tendendo ao resultado ótimo (MONGIOVI *et al.*, 2013).

2.2.4 GAAN: *Generative Adversarial Attributed Network Anomaly Detection*

O modelo GAAN se propõe a identificar nós anômalos em um grafo por meio da comparação entre uma amostra real do dado e uma amostra fictícia. Por meio desse treinamento, o modelo seria capaz de capturar as características dos nós e a estrutura do grafo, associando ao nó uma pontuação, onde quanto maior a pontuação, mais anômalo é considerado o nó (CHEN *et al.*, 2020). Para executar tal tarefa o modelo utiliza de três principais passos:

1. Gerador: Durante o treinamento o gerador aprende a mapear uma entrada aleatória a uma amostra pertencente aos dados originais.

2. Discriminador: O Discriminador é treinado para distinguir se uma amostra do dado é uma amostra real ou se é uma amostra gerada pelo Gerador contendo o nó randômico.

3. Detecção de Anomalia: Essa é a etapa na qual é atribuída uma pontuação para cada nó, onde quanto maior a pontuação, mais anômalo é o nó. Para o cálculo desse escore são utilizadas duas principais métricas: Perda de Reconstrução (G) e Perda do Discriminador de Estrutura (D). Uma pontuação alta em G sugere que os atributos do nó não foram devidamente reconstruídos pelo gerador, ou seja, indicando anormalidade em relação aos dados utilizado no treinamento. Uma pontuação alta em D pode indicar que o nó não faz parte da estrutura na qual está inserido, pois identifica as diferenças do nó que está sendo analisado com os nós adjacentes identificados como reais.

2.2.5 CoLA: *Contrastive self-supervised Learning framework for Anomaly detection on attributed networks*

O objetivo do modelo CoLA é identificar um conjunto de regiões anômalas, explorando de maneira abrangente as informações presentes nos dados pertencentes ao grafo. O modelo

utiliza pares de instâncias contrastantes escolhidos de forma a capturar as relações entre cada nó e suas subestruturas vizinhas. O par utilizado para capturar tal relação é composto do nó e do subgrafo ao qual pertence, atribuindo um valor quantitativo para representar a anormalidade de cada nó, ou seja, o modelo de aprendizado é treinado com ênfase específica na detecção de anomalias (LIU *et al.*, 2021). Para executar tal tarefa o modelo CoLA é baseado nos seguintes princípios:

1. Incorporação de Rede e Redes Neurais de Grafos

   A incorporação de rede visa inserir nós em espaços vetoriais latentes, mantendo intactas as propriedades intrínsecas do grafo, abrangendo informações estruturais e aspectos semânticos. Nesse contexto, as Redes Neurais de Grafos (GNNs) se destacam como uma categoria de redes neurais profundas especialmente desenvolvidas para modelar as complexas relações provenientes de redes ou grafos.

2. Detecção de anomalias em redes atribuídas

   O modelo CoLA faz o uso de amostras de pares compostos de nó e subgrafo, não sendo necessária a rede completa. Essa abordagem possibilita a aplicação em redes de larga escala, capturando de maneira eficaz as informações semânticas e estruturais do grafo utilizando aprendizado auto-supervisionado, no intuito de fornecer indicações da anormalidade dos nós.

3. Aprendizado Contrastivo Auto-Supervisionado

   CoLA utiliza um modelo contrastivo para calcular a pontuação de anormalidade de cada nó. Para isso, ele compara os pares contrastantes (par composto de nó e subgrafo ao qual pertence, e par composto de nó e subgrafo aleatório). Dessa forma esse novo tipo de comparação de pares é capaz de capturar informações semânticas e estruturais locais, em vez de propriedades globais.

2.2.6 Aplicação de Detecção de *Outliers* em Grafos no Contexto de Fraudes Financeiras

Velasco et al. (VELASCO *et al.*, 2020) aplicaram a detecção de anomalias em processos de licitação do governo brasileiro, focando no que denominam *Top Losers* (empresas que mais perdem os processos de licitação). O estudo sugere o uso de análise de grafos gerados pelas interações dessas empresas no processo licitatório e subgrafos gerados pelos vértices mais relevantes para a detecção de possíveis fraudes financeiras (VELASCO *et al.*, 2020).

Outro estudo sugere a aplicação de um modelo não supervisionado em matrizes baseadas em grafos para a detecção de possíveis fraudes em seguro, cartão de crédito e lavagem de dinheiro (HUANG *et al.*, 2019). A estrutura de detecção desenvolvida se mostrou eficiente, podendo ajudar auditores do ramo financeiro a detectar padrões de fraude e rastrear a fraude original baseando-se em características suspeitas.

Molloy et. al (MOLLOY *et al.*, 2017) utilizam o conceito de detecção de fraude baseado em comunidade aplicada em grafos de transações financeiras. Para a análise de comunidade é utilizado alguns atributos de centralidade, clusterização, e subgrafos formados por vértices altamente conectados. O estudo mostrou que os atributos dos grafos são uma excelente fonte de informações para a discriminação entre transações normais e fraudulentas (MOLLOY *et al.*, 2017).

3 METODOLOGIA

Nesta seção são descritos os passos realizados para o tratamento dos dados e a aplicação dos algoritmos em cada contexto. O link para o repositório contendo o código-fonte criado pelos autores e utilizado nos passos de análise exploratória, preparação dos dados, implementação dos métodos e avaliação dos resultados pode ser encontrado no Apêndice A.

3.1 Origem dos Dados

Para este trabalho foi disponibilizada uma base de dados estruturados contendo informações sobre documentos de licitação provindos do Sistema de Saúde do Estado da Paraíba referentes ao ano 2016. A base apresenta 2,089,317 registros e os valores são distribuídos em 62 colunas. As linhas representam itens de uma nota fiscal, sendo que uma nota pode apresentar mais de um item. Sendo assim, existe uma relação 1 para N entre a nota fiscal e seus itens. Os dados são referentes à produtos farmacêuticos apenas e não há a classificação de cada registro que o identifique como fraude ou não.

As principais informações presentes na base são:

- O número identificador da nota;
- A data de emissão;
- O identificador das entidades emissora e destinatária;
- A localização física das entidades emissora e destinatária;
- O valor total da nota;
- Outros valores que constituem o valor total da nota (valor da base de cálculo do Imposto Sobre Circulação de Mercadorias e Serviços (ICMS), do frete, entre outros)
- O identificador do tipo de cada produto;
- O valor unitário de cada produto;
- A quantidade do produto;
- Outros valores que constituem o valor total do produto (valor do Imposto Sobre Circulação de Mercadorias e Serviços (ICMS), do Imposto Sobre Produtos Industrializados (IPI), entre outros)

Em relação a essas informações a base de dados apresenta os seguintes valores distintos:

- 66,765 notas fiscais;
- 9,384 entidades emissoras;
- 1,168 entidades destinatárias;
- 921 municípios;
- 6,005 tipos de produtos com base no código NCM [1].

3.2 Processamento dos Dados

As seguintes etapas de limpeza e transformação de dados foram realizadas a partir dos dados originais:

1. A partir da análise da base de dados, as colunas irrelevantes ou contendo uma quantidade majoritária de valores nulos ou zeros foram retiradas, mantendo apenas os seguintes dados sobre as notas fiscais:
    - Número da nota fiscal
    - Número do item
    - Data de emissão
    - Valor total da nota
    - Nome da empresa emitente
    - CNPJ da empresa emitente
    - Bairro da empresa emitente
    - Município da empresa emitente
    - CEP da empresa emitente
    - Nome da empresa destinatária
    - CNPJ da empresa destinatária
    - Bairro da empresa destinatária
    - Município da empresa destinatária
    - CEP da empresa destinatária

[1] A Nomenclatura Comum do Mercosul (NCM) diz respeito é um código que "[...] permite, pela aplicação de regras e procedimentos próprios, determinar um único código numérico para uma dada mercadoria.". Dessa forma, cada tipo de produto possui um código NCM associado a ele e usado, fundamentalmente, na aplicação de tributos em operações de comércio exterior (BRASIL, 2019). "CFOP é a abreviação de Código Fiscal de Operações e Prestações. Esse código identifica uma determinada operação por categorias no momento da emissão da nota fiscal"(TORRES, 2022).

- Descrição do produto ou serviço prestado
- NCM do produto ou serviço
- CFOP do produto ou serviço
- Quantidade do item
- Unidade de medida do item
- Valor unitário do item
- Valor total dos itens

2. Criação de uma nova coluna contendo o valor da data de emissão transformada para segundos a partir da Era Unix.
3. Criação de duas novas colunas contendo a latitude e longitude do município informado da entidade emitente e destinatária.

A nova base de dados descrita acima foi utilizada como a base de referência no processo de avaliação dos resultados encontrados pelos algoritmos de detecção de anomalias. Porém, uma terceira base foi criada com o fim de ser usada na aplicação dos modelos de aprendizado de máquina. Para esta base, os seguintes passos foram realizados:

1. Escolha das características mais relevantes para o aprendizado de máquina:
    - Tempo, em segundos a partir da Era Unix, da data de emissão da nota fiscal
    - Latitude do emitente
    - Longitude do emitente
    - Latitude do destinatário
    - Longitude do destinatário
    - Valor do NCM
    - Quantidade de itens
    - Valor unitário do item
2. Para cada coluna foram aplicados os seguintes métodos de redimensionamento e normalização de dados:
    - Tempo: *Min Max Scaling*
    - Latitude do emitente: *Robust Scaling*
    - Longitude do emitente: *Robust Scaling*
    - Quantidade de itens: transformação logarítmica
    - Valor unitário do item: transformação logarítmica

Em relação à transformação realizada na coluna de tempo, foi necessário transformá-la em segundos para que fosse possível utilizá-la como uma variável numérica, pois os algoritmos utilizados apenas podem receber como entradas variáveis deste tipo. Ou seja, datas e *timestamps* não são numéricas e é necessário transformá-la para segundos para que o tempo passe a ser representado numericamente. Além disso, ao transformá-la em segundos é possível ter uma noção de evolução temporal de forma que o tempo possa ser utilizado também como uma variável contextual. É importante reconhecer que com o tempo os valores podem mudar por causa da inflação ou de mudanças sazonais de preços. A Era Unix tem início em 1 de Janeiro de 1970 e é comumente utilizada como a base para as transformações de datas para seu equivalente em segundos. Ou seja, o valor em segundos resultante da transformação tem seu primeiro segundo no início da Era Unix e o valor relativo à uma data é calculada a partir da diferença em segundos entre a data da instância e a data do início da era Unix.

As técnicas de normalização e redimensionamento têm o objetivo de modificar os valores das colunas de forma a otimizar ou facilitar a a análise dos dados pelos algoritmos de aprendizado de máquina e não permitir que diferenças de escala influenciem negativamente o processo de aprendizado. Há uma variedade de técnicas diferentes que devem ser escolhidas dependendo do tipo de dado e do objetivo da aplicação.

O *Min Max Scaling* foi utilizado para normalizar a coluna de tempo resultando em valores entre 0 e 1. Esse método foi escolhido pois a distribuição de dados para essa coluna é aproximadamente uniforme como visto na etapa de análise exploratória, não havendo a necessidade de utilizar outra forma de escalonamento.

O *Robust Scaling*, utilizado nas variáveis longitude e latitude, é um método de normalização mais resistente aos efeitos de *outliers* (SPENCE; LEWANDOWSKY, 1989). Este método foi usado para balancear o impacto das variáveis e possivelmente melhorar a performance do algoritmo. Visto que a localização das entidades não está centralizada em um lugar específico, o uso do método *Robust Scaling* é justificado para evitar que cidades localizadas a grandes distâncias para com as demais demais afetem negativamente a normalização dos dados. Isso ajuda a manter a distribuição de dados intacta mesmo com a presença de *outliers*.

A transformação logarítmica foi usada nas colunas de quantidade e valor unitário dos itens. O método foi escolhido para balancear os dados pois, de acordo com a análise exploratória, essas características possuem uma faixa extensa de valores. A fim de comprimir os dados permitindo a melhor visualização e menor sensitividade a *outliers* a transformação logarítmica foi aplicada.

Após o processo de limpeza, transformação e normalização de dados, uma nova base foi criada contendo apenas os dados com o valor do NCM mais comum (código 3004.90.99). Essa decisão foi tomada para que fosse possível aplicar modelos de aprendizado de máquina, como o SOM, que demandam uma quantidade mais alta de memória em comparação a algoritmos como o iForest e o LOF e não poderiam ser aplicados com os dados em sua inteireza. Com isso, além da quantidade reduzida de dados (totalizando 76457 entradas), também houve a redução

de dimensionalidade permitindo a melhor aplicação dos algoritmos de detecção e facilitando a análise dos resultados encontrados visto que todos os itens pertencem à mesma categoria.

3.3 Implementação dos Métodos

3.3.1 Métodos em Dados Estruturados

Três algoritmos de aprendizado de máquina para detecção de *outliers* foram aplicados utilizando a base de dados restrita pelas entradas com o valor do NCM mais comum: Self-Organizing Maps (SOM), Isolation Forest (iForest) e Local Outlier Factor (LOF). A Tabela 1 apresenta o valor dos parâmetros relevantes usados em cada modelo.

Quadro 1 – Métodos e parâmetros utilizados.

Método	Biblioteca	Parâmetros
SOM	Minisom (VETTIGLI, 2018)	contamination = 0.01 map size = 40x40 sigma = 3 learning rate = 0.5 neighborhood function = triangle random seed = 26 training iterations = 1,000,000
iForest	Scikit IsolationForest (PEDREGOSA *et al.*, 2011)	contamination = 0.01 random state = 26 number of estimators = 100 max features = 1.0 max samples = auto
LOF	Scikit LocalOutlierFactor (PEDREGOSA *et al.*, 2011)	contamination = 0.01 number of neighbors (k) = 10 algorithm = auto leaf size = 30 metric = minkowski

Fonte: Autoria própria (2023).

No caso do SOM, o valor do tamanho do mapa (*map size*) foi definido utilizando como base a equação apresentada por Versanto para o Toolbox do Matlab que utiliza como métrica principal a quantidade de instâncias da base de dados (N) (SHALAGINOV; FRANKE, 2015):

$$S = 5 \cdot \sqrt{N}$$

Para o LOF a quantidade de vizinhos (*number of neighbors*) foi escolhido com base na quantidade mínima informada pelos criadores do método que definem que um número menor que 10 implicaria na existência de ruídos que afetariam negativamente o cálculo do LOF (BREUNIG *et al.*, 2000). Dado que não há conhecimento prévio do tamanho dos *clusters* encontrados na base de dados e tendo em mente que para muito casos o tamanho de um *cluster* relativo à

um produto específico e suas características temporais e espaciais é relativamente pequeno (< 10 instâncias), o valor especificado parece ser adequado para o problema.

As instâncias anômalas são descobertas analisando o seu *anomaly score* que pode ser inferido de diferentes maneiras para cada algoritmo. No caso do SOM a pontuação é definida a partir do cálculo da distância de cada instância em relação a seu BMU. Os pontos com distâncias mais altas são considerados anomalias. Para o iForest a pontuação é encontrada a partir do cálculo da distância em que um nó-folha que representa uma instância está da raiz. Distâncias pequenas indicam prováveis *outliers*. E para o LOF a pontuação é o próprio valor do Local Outlier Factor que é calculado para cada instância. Quanto maior o valor, mais alta a probabilidade do registro ser anômalo.

A partir do *anomaly score* de cada instância, os resultados de cada algoritmo foram ordenados de forma que os registros com maior probabilidade de serem anomalias fossem encontrados no topo da lista. Então, as 10 instâncias com pontuações maiores foram escolhidas a fim de serem analisadas manualmente pelos desenvolvedores que, ao realizar comparações entre as instâncias categorizadas como anômalas e os demais registros relevantes, puderam avaliar os resultados encontrados a partir dessa amostra.

Um exemplo de análise realizada pode ser observada na Figura 6 em que o modelo gerado pelo iForest apontou duas instâncias anômalas para o mesmo produto. O primeiro passo para a análise foi identificar as características relevantes sobre o registro categorizado como *outlier* pelo algoritmo; no caso do exemplo, foram utilizadas as informações apresentadas pelo campo de descrição do produto onde são informados seu nome e dosagem. A partir dessa descrição, foram escolhidos os termos "sinvastatina" e "40mg" para realizar uma pesquisa na base de dados e permitir a comparação entre os registros. Como é possível perceber na tabela resultante da pesquisa, as instâncias com índice 71968 e 71970 (precisamente os registros apontados pelo iForest), apresentam um valor unitário relativamente mais baixo que as demais instâncias, assim como quantidades expressivamente mais altas. Esses registros foram identificados como verdadeiras anomalias pelo desenvolvedor e dois possíveis casos de superdimensionamento.

É importante, porém, ressaltar que os valores encontrados podem ser provenientes de diferentes contextos como a localização da entidade destinatária ou situações anômalas mas legítimas que podem justificar a compra de quantidades altas do produto. Sendo assim, em trabalhos futuros, é importante incluir mais dados de contextualização principalmente variáveis socioeconômicas dos lugares que realizaram a compra dos produtos, e realizar uma análise mais minuciosa para procurar razões que podem justificar os valores encontrados nas notas fiscais.

Figura 6 – Exemplo de comparação de instâncias anômalas (índice 71968 e 71970) com registros similares (parte das linhas e colunas foram omitidas para facilitar a clareza e legibilidade).

	prod_ncm	prod_desc	prod_quant	prod_valor_unit	prod_valor_total
18896	30049099	sinvastatina 40mg (sanval) sinvaston	30.0	0.28	8.4
18924	30049099	sinvastatina 40mg (sanval) sinvaston - lote: at412 31/05/2017	100.0	0.35	35.0
19123	30049099	sinvastatina 40mg (sanval) sinvaston - lote: at412 31/05/2017	500.0	0.19	95.0
20331	30049099	sinvastatina 40mg comprimidos	500.0	0.37	185.0
20336	30049099	sinvastatina 40mg comprimidos	1020.0	0.37	377.4
21195	30049099	sinvastatina 40mg comprimidos	2100.0	0.21	441.0
23443	30049099	sinvastatina 40mg (sanval) sinvaston	300.0	0.35	105.0
23913	30049099	sinvastatina 40mg comprimidos	2100.0	0.21	441.0
26088	30049099	sinvastatina 40mg (sanval) sinvaston	500.0	0.35	175.0
26491	30049099	sinvastatina 40mg comprimidos	2100.0	0.21	441.0
27046	30049099	sinvastatina 40mg (sanval) sinvaston	600.0	0.19	114.0
27606	30049099	sinvastatina 40mg comprimidos	394.0	0.40	157.6
28264	30049099	sinvastatina 40mg comprimidos	500.0	0.37	185.0
71968	30049099	sinvastatina 40mg comprimido multilab	150000.0	0.12	18000.0
71970	30049099	sinvastatina 40mg comprimido multilab	60000.0	0.13	7800.0

Fonte: Autoria própria (2023).

3.3.2 Métodos em Grafos

Dois algoritmos de aprendizado de máquina para detecção de *outliers* foram aplicados utilizando a base de dados restrita pelas entradas com o valor do NCM mais comum: *Contrastive self-supervised Learning framework for Anomaly detection on attributed networks* (CoLA) e *Generative Adversarial Attributed Network Anomaly Detection* (GAAN). O quadro 2 apresenta o valor dos parâmetros relevantes usados em cada modelo.

Quadro 2 – Métodos e parâmetros utilizados.

Método	Biblioteca	Parâmetros
CoLA	PyGOD CoLA (LIU *et al.*, 2022)	contamination = 0.01 layers = 2 training iterations = 500 hidden dimension = 64
GAAN	PyGOD GAAN (LIU *et al.*, 2022)	contamination = 0.01 layers = 2 training iterations = 500 hidden dimension = 64

Fonte: Autoria própria (2023).

Na utilização dos algoritmos a maioria dos parâmetros foram mantidos com seus valores padrão no intuito de se manter o mais fiel possível à implementação original. Diferentemente dos métodos aplicados anteriormente, foi atribuído a cada nó do grafo (vendedores e compradores)

números para facilitar sua representação e aplicação do modelo. Ambos os algoritmos retornam um indicador de anomalia para cada nó presente no grafo, o *raw outlier score*, em que quanto maior o número, mais anômala é o nó em relação aos outros.

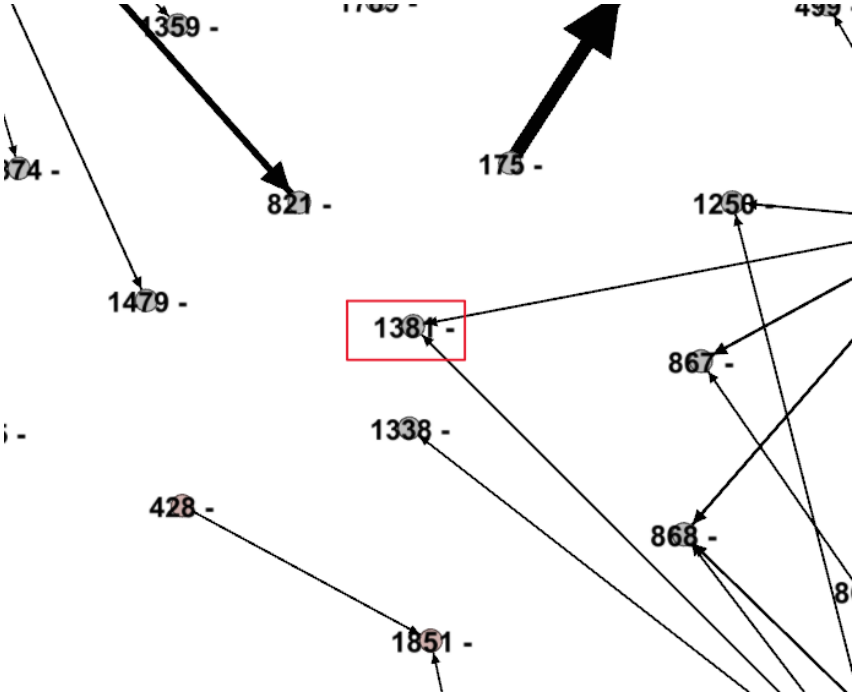
Os resultados obtidos após a utilização dos algoritmos foram ordenados a fim de se obter as 5 instâncias que possuíam o *raw outlier score* mais alto. Após a escolha das 5 instâncias mais anômalas para cada algoritmo, esse resultado foi analisado manualmente realizando comparações entre as instâncias categorizadas como anômalas e os demais registros presentes nos dados para avaliar os resultados obtidos. Um exemplo é a análise feita para o nó 1381, apontado pelo algoritmo CoLA como uma possível anomalia. Utilizando as informações presentes no campo de descrição do produto podemos descobrir de que o nó 1381 comprou 4 produtos diferentes que podem ser observados na Figura 7. Comparando os valores atribuídos ao nó 1381 podemos ver que três produtos estão com o valor um pouco acima da média, como podemos observar no quadro 3, e em um dos casos chega a ser o valor mais alto quando comparado às interações de compras de produtos similares efetuadas por outros nós, podendo esse ser um dos fatores para ser classificado como anômalo. Com base nessas informações podemos identificar os valores um pouco acima da média como um possível fator para ser considerado uma instancia anômala. No caso do produto "lyrica 150mg cx 28 cap" possuir o maior valor de compra quando comparado com todos os produtos similares presentes no conjunto de dados e pelo fato do nó ter executado apenas duas compras totalizando quatro produtos, sendo um nó pouco conexo no grafo, como podemos ver na Figura 8, contribuindo também para a sua classificação como anômalo.

Figura 7 – Exemplo de interações de compra efetuadas pelo nó 1381.

prod_desc	prod_ncm	prod_cfop	prod_quant	prod_unid	prod_valor_unit
fumarato de quetiapina 100mg 30cp bios	30049099	5929	3.0	und	263.85
carbolitium 300mg 50cp	30049099	5929	2.0	und	36.70
lyrica 150mg cx 28 cap	30049099	5929	1.0	und	202.58
ganfort sol est adt 3ml	30049099	5929	1.0	und	106.72

Fonte: Autoria própria (2023).

Figura 8 – Grafo representando interações de compra feitas pelo nó 1381.



Fonte: Autoria própria (2023).

Quadro 3 – Análise do Nó 1381.

Produto	Valor Nó 1381	Valor Máximo	Valor Mínimo	Valor Médio	Nó x Média
fumarato de quetiapina	263.85	273,12	150	218,92	+20,52%
carbolitium	36,70	40,37	9,45	35,42	+3,61%
lyrica	202,58	202,58	50,12	193,91	+4,47%
ganfort	106,72	109,8	49,76	99,81	+6,92%

Fonte: Autoria própria (2023).

4 ANÁLISE EXPLORATÓRIA

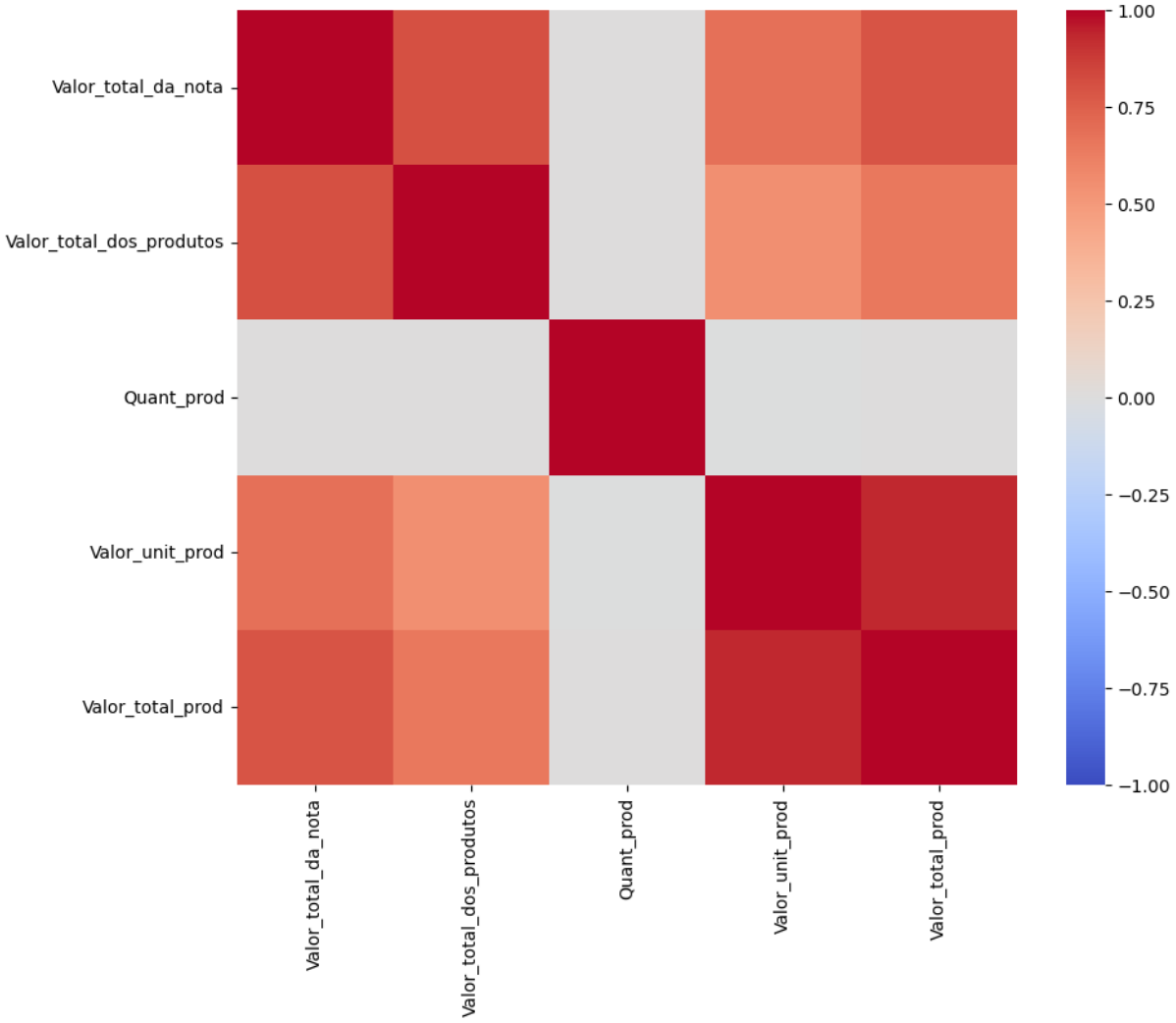
Nesta seção serão discutidos os resultados encontrados no processo de análise exploratória dos dados que visa compreender a distribuição dos dados e avaliar se a base é apropriada para ser utilizada na aplicação dos métodos de aprendizado de máquina. O link para o repositório contendo o código-fonte criado pelos autores e utilizado nos passos de análise exploratória, preparação dos dados, implementação dos métodos e avaliação dos resultados pode ser encontrado no Apêndice A.

4.1 Análise Exploratória em Dados Estruturados

A partir da análise inicial, percebeu-se que muitas colunas referentes aos valores obtidos das notas fiscais apresentavam em sua maioria valores nulos. Como consequência, as colunas que apresentaram esse comportamento foram descartadas da análise pois sua presença não seria adequada na aplicação dos modelos. Das colunas numéricas restantes que têm seus valores na maior parte não-nulos, foi observado que em algumas delas havia uma quantidade desproporcional de valores zerados que também foram descartadas. O processo de omitir as colunas desnecessárias foi realizado manualmente partir da análise que indica se ela é relevante ou não.

Utilizando as variáveis numéricas remanescentes foi gerada uma matriz de correlação para compreender a relação entre as variáveis. A Figura 9 apresenta os resultados encontrados para esta análise, que indica que a correlação entre as variáveis, quando existe, é positiva; ou seja, quando uma aumenta, a outra também aumenta. A exceção é para o atributo Quantidade do Item, que não estabelece uma correlação com nenhum outro atributo. Apesar deste fator, a variável em questão será mantida pois ainda assim é possível detectar anomalias como quantidades de itens que não estão de acordo com a quantidade normal encontrada em seu contexto.

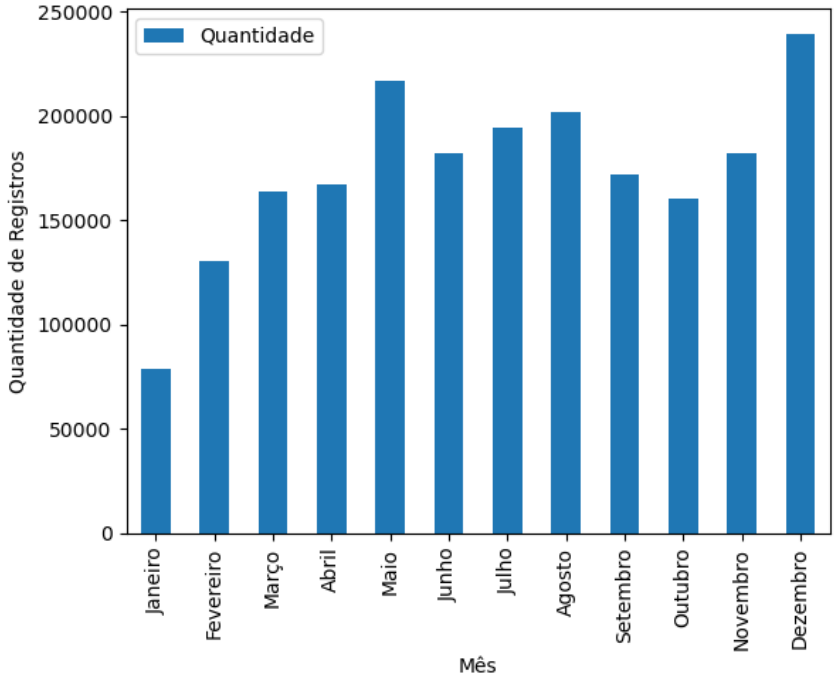
Figura 9 – Matriz de correlação entre as variáveis numéricas escolhidas para análise.



Fonte: Autoria própria (2023).

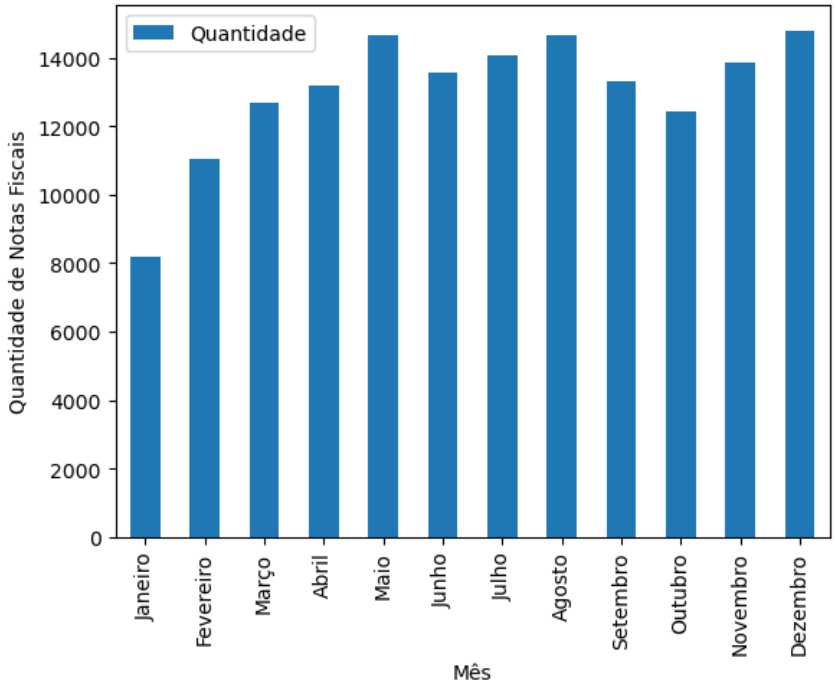
Os dados disponibilizados são referentes apenas ao ano de 2016 e em relação à distribuição registros por mês, a partir das Figuras 10, 11 e 12 observa-se que os meses de Janeiro e Fevereiro apresentam um quantidade menor de registros comparado aos demais meses. Outro apontamento está na quantia anormal da soma do valor dos produtos para o mês de Dezembro que é relativamente maior que aos demais meses. Uma análise de detecção de *outliers* poderá apontar o motivo. Ainda que hajam algumas discrepâncias entre os valores para cada mês, os resultados sugerem que as análises realizadas a partir do uso do atributo Data serão adequados para compor o conjunto de atributos contextuais; há uma quantidade suficiente de registros para cada mês que permite realizar o processo de construção dos modelos de aprendizado de máquina.

Figura 10 – Quantidade de registros por mês.



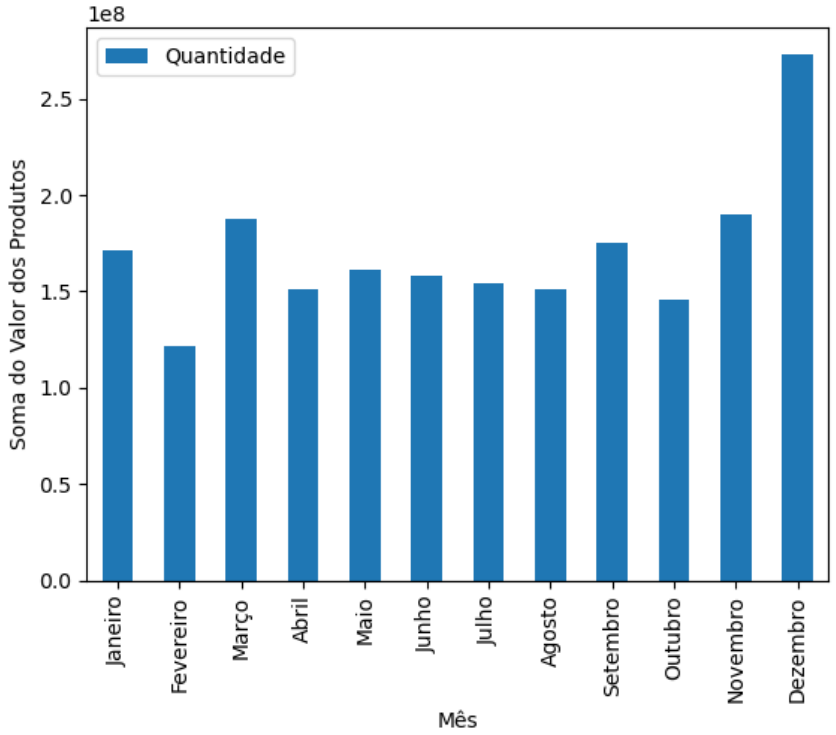
Fonte: Autoria própria (2023).

Figura 11 – Quantidade de notas fiscais por mês.



Fonte: Autoria própria (2023).

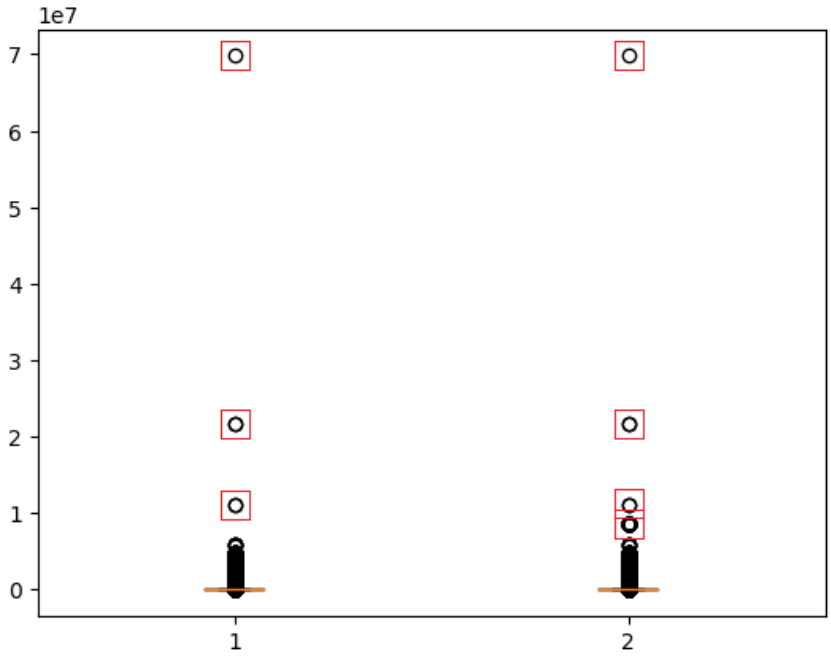
Figura 12 – Soma do valor dos produtos das notas fiscais por mês. A soma do valor dos produtos (eixo Y) é o valor multiplicado por 10^8 em reais.



Fonte: Autoria própria (2023).

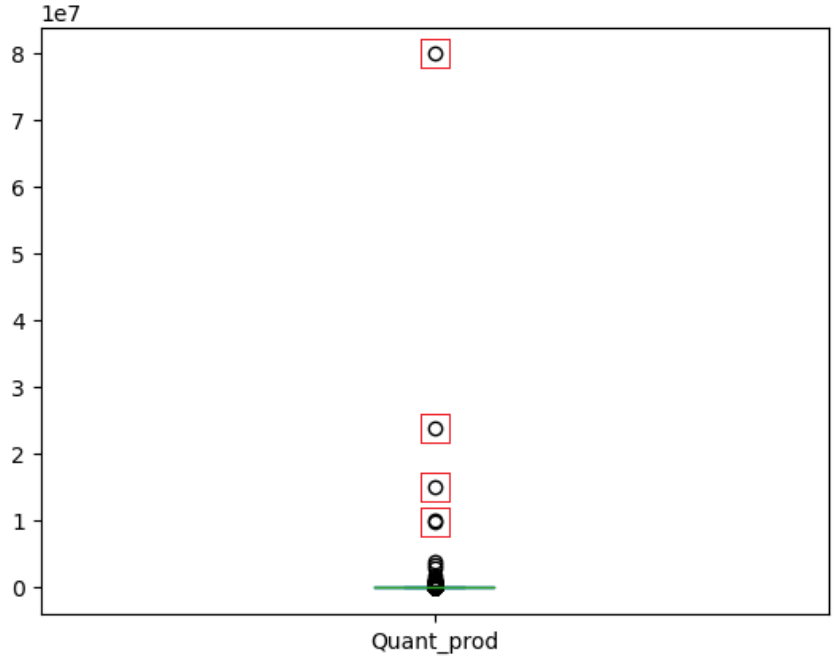
Outra análise efetuada foi a construção de *box plots* a partir dos atributos definidos. Percebe-se através das Figuras 13, 14 e 15 que as faixas de valores encontradas nos atributos numéricos é muito extensa. A maior parte dos valores é relativamente baixo, mas há a presença de valores expressivamente altos que atrapalham a visualização da distribuição dos atributos. As instâncias evidenciadas pelos quadrados vermelhos nas figuras indicam anomalias dado que seus valores se diferem notadamente dos demais. Outra observação que também indica um *outlier* é encontrada na Figura 13 onde para ambas as colunas (Valor Total e Valor Total dos Produtos) seria evidente que se comportassem de maneira similar, porém há uma instância no *box plot* da segunda coluna que não é refletida na primeira.

Figura 13 – Box plot das colunas Valor Total (1) e Valor Total dos Produtos (2). Os quadrados vermelhos indicam fortes candidatos a serem *outliers*. O valor real das variáveis é inferido a partir da multiplicação dos valores do eixo Y por 10^7.



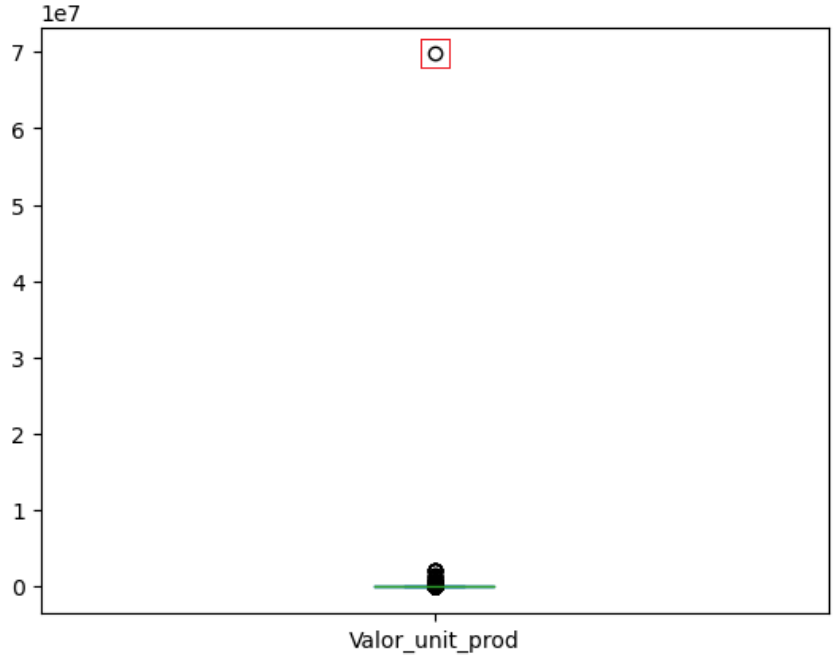
Fonte: Autoria própria (2023).

Figura 14 – Box plot da coluna Quantidade do Item. Os quadrados vermelhos indicam fortes candidatos a serem *outliers*. O valor real da variável é inferido a partir da multiplicação dos valores do eixo Y por 10^7.



Fonte: Autoria própria (2023).

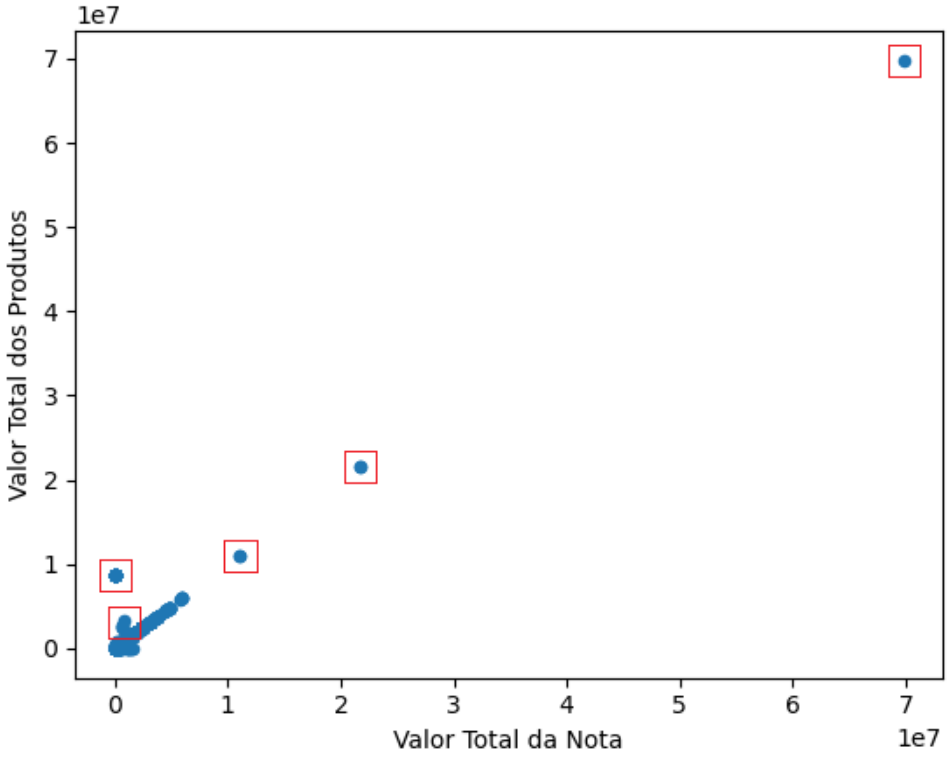
Figura 15 – Box plot da coluna Valor Unitário do Item. O quadrado vermelho indica um forte candidato a ser *outliers*. O valor real da variável é inferido a partir da multiplicação dos valores do eixo Y por 10^7.



Fonte: Autoria própria (2023).

Outros indícios de *outliers* podem ser observados na Figura 16, que além de indicar instâncias em que os valores para ambas as colunas diferem significativamente das demais, também mostra que há anomalias na relação entre as variáveis. Para esta correlação o diagrama deveria apresentar uma reta diagonal formada pelas instâncias, o que sugere que as variáveis estão fortemente correlacionadas, mas algumas instâncias apresentam um comportamento divergente. Os registros apresentadas nos quadrados vermelhos podem ser classificadas como anomalias. Percebe-se ainda a presença de instâncias que pertencem à diagonal mas que possuem valores altos que os distanciam do conjunto de dados que contém a maior parte das instâncias. Estas também podem ser consideradas como anomalias dado que estão afastadas do conjunto principal.

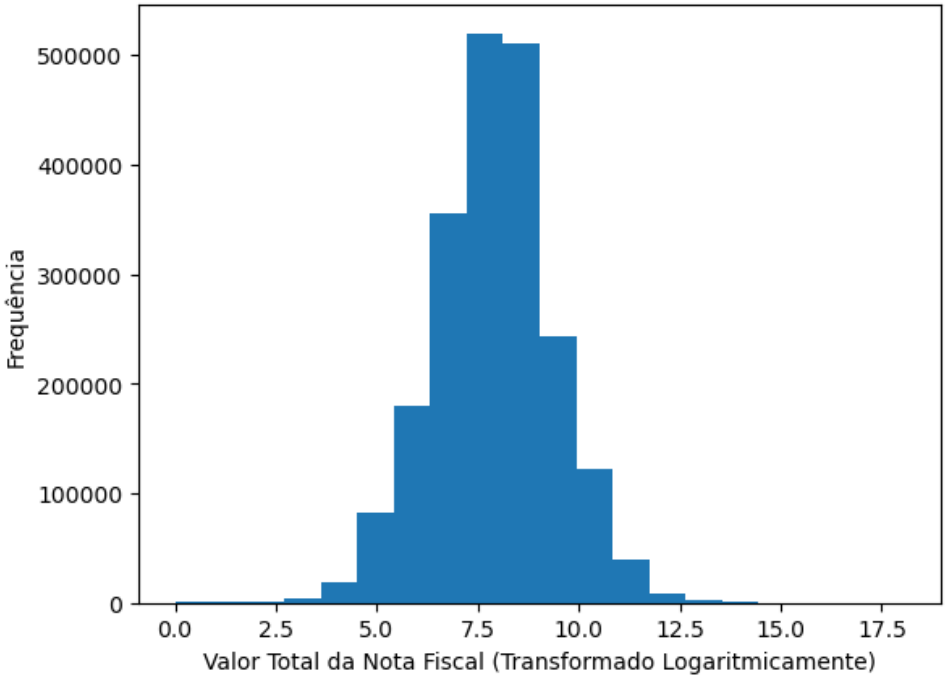
Figura 16 – Gráfico de dispersão entre as colunas Valor Total e Valor Total dos Produtos.



Fonte: Autoria própria (2023).

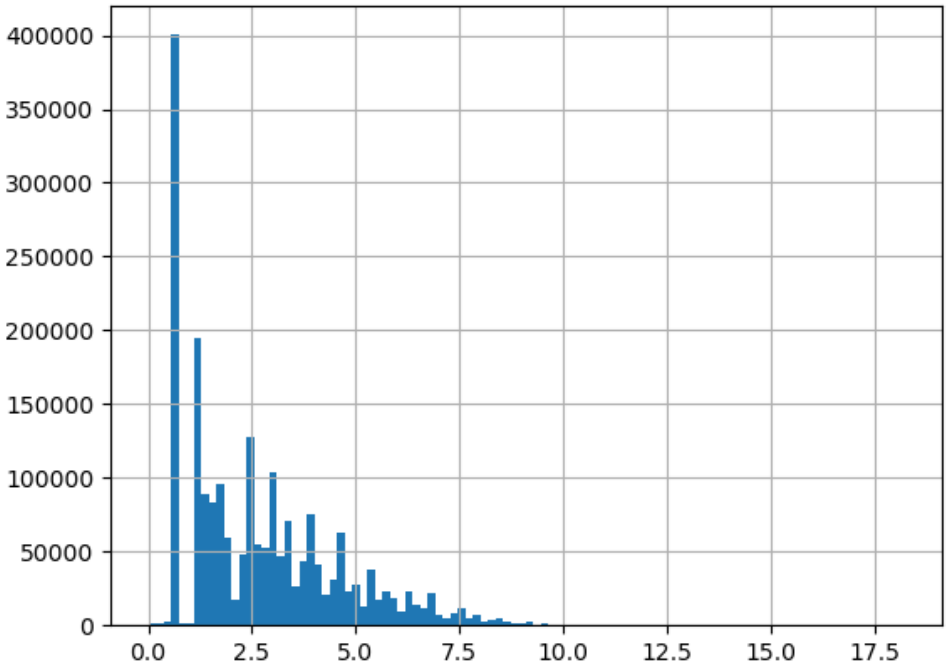
Para procurar compreender melhor as variáveis numéricas visto que possuem uma faixa de valores muito extensa, foi efetuada uma transformação logarítmica para comprimir os valores e permitir uma melhor visualização. As Figuras 17, 18 e 19 apresentam os dados após essa transformação e indicam a relativa grande extensão dos valores encontrados, mas não apontam nenhuma inconsistência relevante.

Figura 17 – Distribuição da coluna Valor Total. (Transformado Logaritmicamente)



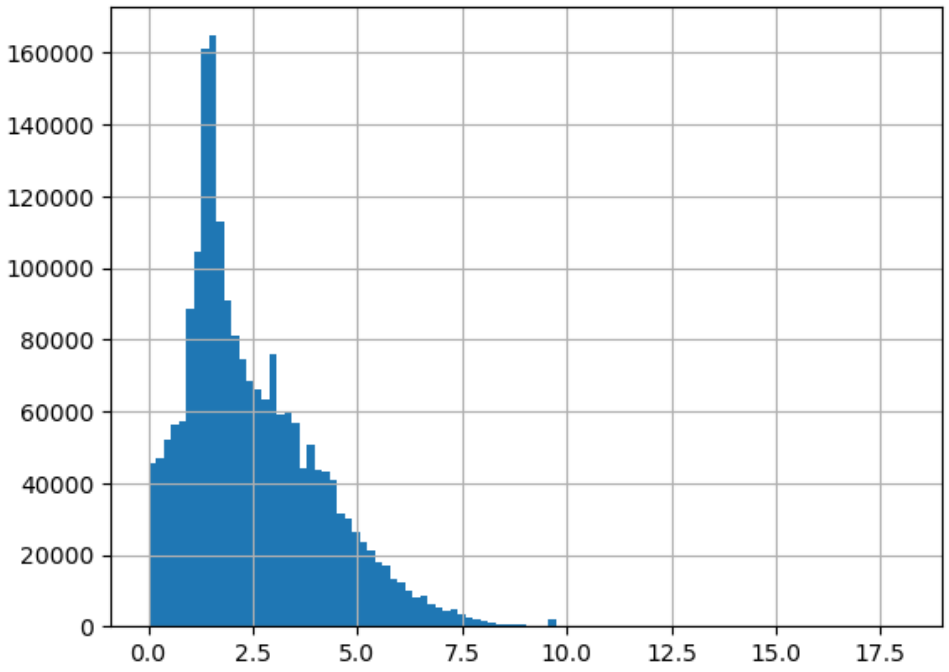
Fonte: Autoria própria (2023).

Figura 18 – Distribuição da coluna Quantidade do Item. (Transformado Logaritmicamente)



Fonte: Autoria própria (2023).

Figura 19 – Distribuição da coluna Valor Unitário do Item. (Transformado Logaritmicamente)



Fonte: Autoria própria (2023).

A análise exploratória dos dados estruturados apontou que a base de dados não é completa para todas as colunas contendo os valores das notas fiscais, o que levou à redução das variáveis úteis. Ainda assim, os atributos mais importantes para a aplicação dos modelos se mostram válidos e em sua maior parte consistentes. A análise inicial validou a adequação dos dados que, em conjunto com os atributos contextuais, serão utilizados no processo de aprendizagem de máquina.

4.2 Análise Exploratória em Grafos

A análise em grafos foi feita em intervalos de tempo específicos e estáticos, não utilizando o conjunto completo de dados devido a restrições de memória do computador utilizado. Foram utilizados os softwares Gephi (versão 0.1.1), e Cytoscape (versão 3.10.0), para a análise exploratória.

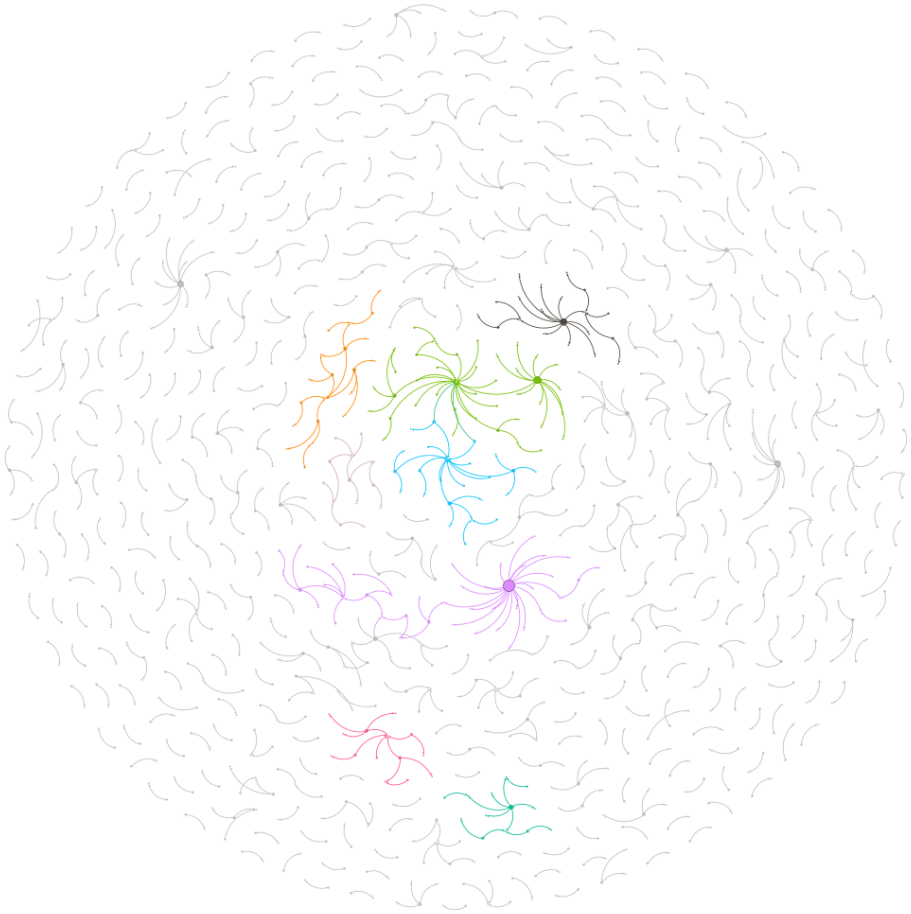
O grafo apresentado na Figura 20 foi gerado utilizando os dados de 01 de Janeiro de 2016 até 07 de Janeiro de 2016. O grafo é direcionado, com o vértice saindo do vendedor e chegando no comprador. Os nós foram coloridos usando o algoritmo de modularidade, para detectarmos possíveis comunidades. O tamanho dos nós reflete o grau de saída de cada nó. O algoritmo de disposição utilizado é o *Fruchterman Reingold*. Esse grafo possui 1248 vértices, entre compradores e vendedores, e 849 arestas representando a relação de compra. Podemos notar algumas comunidades se formando mais ao centro, e vértices mais dispersos nas extremidades (coloridos em cinza).

O grafo apresentado na Figura 21 segue o mesmo padrão de criação da Figura 20, citada anteriormente, porém utilizando os dados de 01 de Janeiro de 2016 até 01 de Março de 2016 e o algoritmo de disposição *ForceAtlas 2*. Esse grafo possui 9906 vértices, entre compradores e vendedores, e 11682 arestas representando a relação de compra. Podemos notar uma grande expansão do grafo analisando apenas os dois primeiros meses, com um aumento expressivo de relações e aumento das comunidades.

Os grafos apresentado nas Figuras 22 e 23 foram gerado utilizando o intervalo de tempo entre 1 de Janeiro de 2016 e 1 de Março de 2016. Podemos observar diversos nós desconexos do grafo principal que poderiam ser consideradas anomalias. Ao selecionar algumas dessas estruturas desconexas (destacadas em vermelho) para análise, percebemos que exemplificam um único vendedor atendendo diversos compradores de forma exclusiva, ou seja, o comprador não teve nenhuma relação com outro vendedor, podendo indicar uma possível instância anômala.

Os grafos apresentados nas Figuras 24 e 25 foram gerados com os dados referentes a 1 de Janeiro de 2016 a 7 de Janeiro de 2016 e 1 de Janeiro de 2016 a 1 de Março de 2016 respectivamente. A Figura 24 teve seu grafo filtrado para mostrar as 10 maiores comunidades e a Figura 25 teve seu grafo filtrado para mostrar as 5 maiores comunidades encontradas utilizando o algoritmo de modularidade. Nesses grafos podemos observar as comunidades mais fortes

Figura 20 – Grafo gerado pelas relações de compra e venda no período entre 1 de Janeiro de 2016 a 7 de Janeiro de 2016.



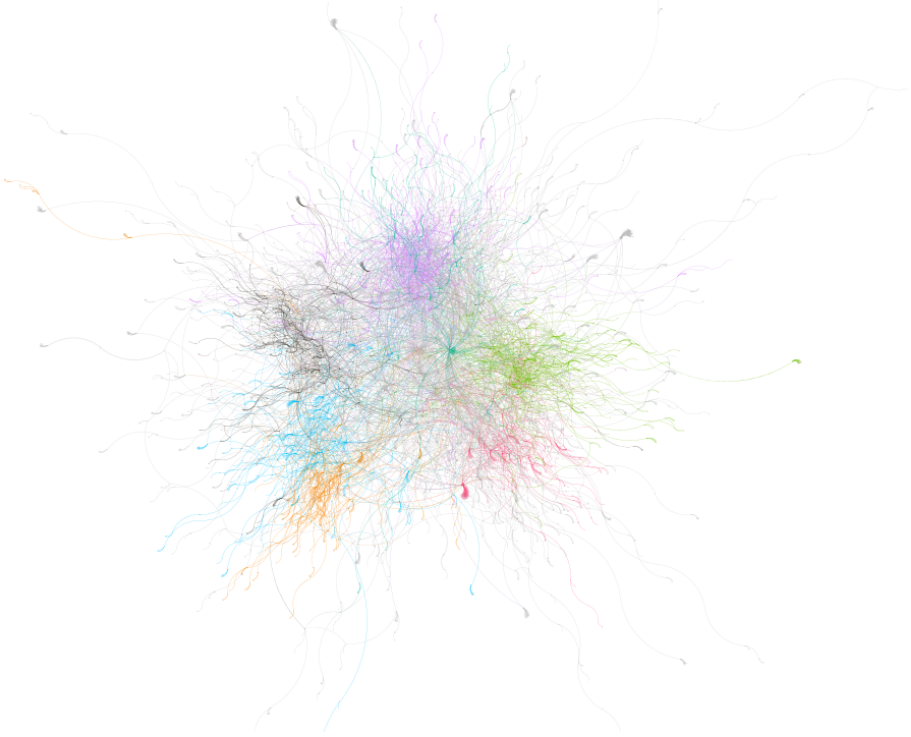
Fonte: Autoria própria (2023).

(que possuem muitos nós pertencentes e mais interconectados), ou seja, são os compradores e vendedores que possuem mais relações de compras entre si nos intervalos de tempo citados.

Para analisar os vendedores com o maior número de vendas e compradores com um maior número de compras, foi gerado um gráfico de distribuição de graus de saída dos nós (Figura 26) e distribuição de graus de entrada dos nós (Figura 27) no período de 1 de Janeiro de 2016 a 1 de Março de 2016. Foi notado que os 10 maiores compradores possuem de 32 a 160 interações de compra com diferentes vendedores e os 10 maiores vendedores possuem de 39 a 80 interações de venda para diferentes compradores. Porém, a maioria dos vértices possuem um baixo número de conexões, fazendo com que a média do grau dos vértices seja 1,179, ou seja, a maioria dos nós possuem apenas uma interação de compra e venda.

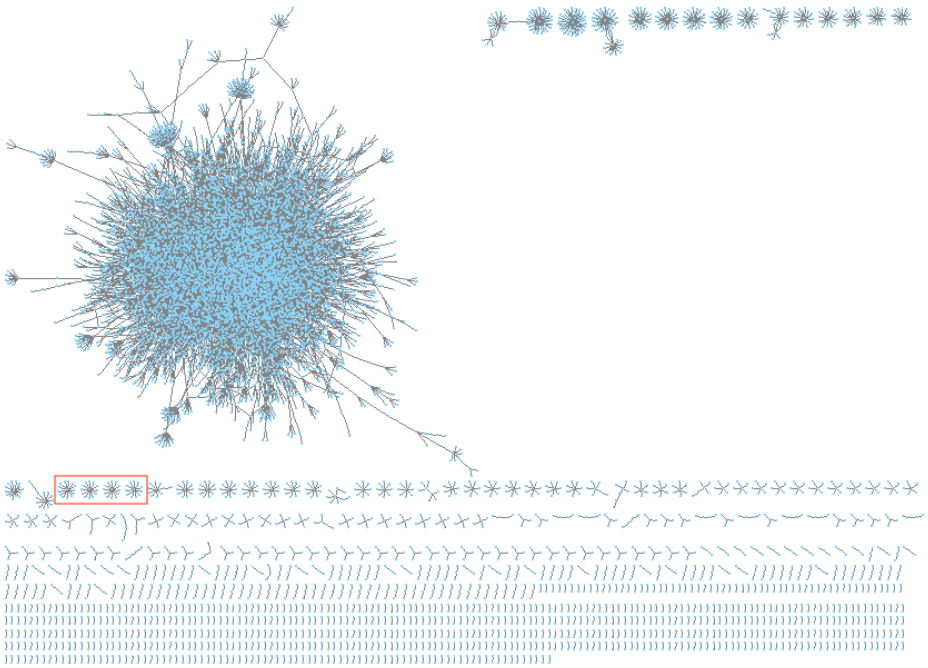
Para a análise de quais vértices seriam os mais influentes no grafo foi gerado o gráfico de distribuição de centralidade de autovetor (Figura 28), onde vértices com maior medida de centralidade são mais importantes e centrais dentro da estrutura do grafo, possuindo mais conexões com vértices de maior importância.

Figura 21 – Grafo gerado pelas relações de compra e venda no período entre 1 de Janeiro de 2016 a 1 de Março de 2016.



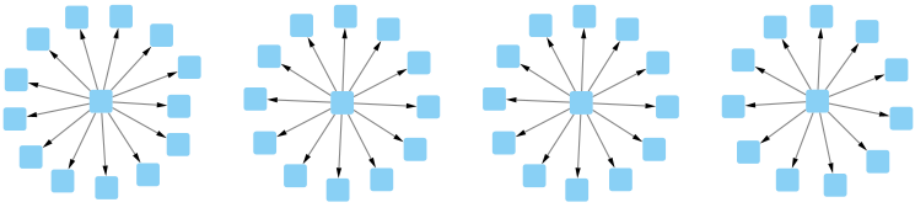
Fonte: Autoria própria (2023).

Figura 22 – Grafo gerado pelas relações de compra e venda no período entre 1 de Janeiro de 2016 a 1 de Março de 2016. Alguns nós desconexos foram destacados com retângulo vermelho.



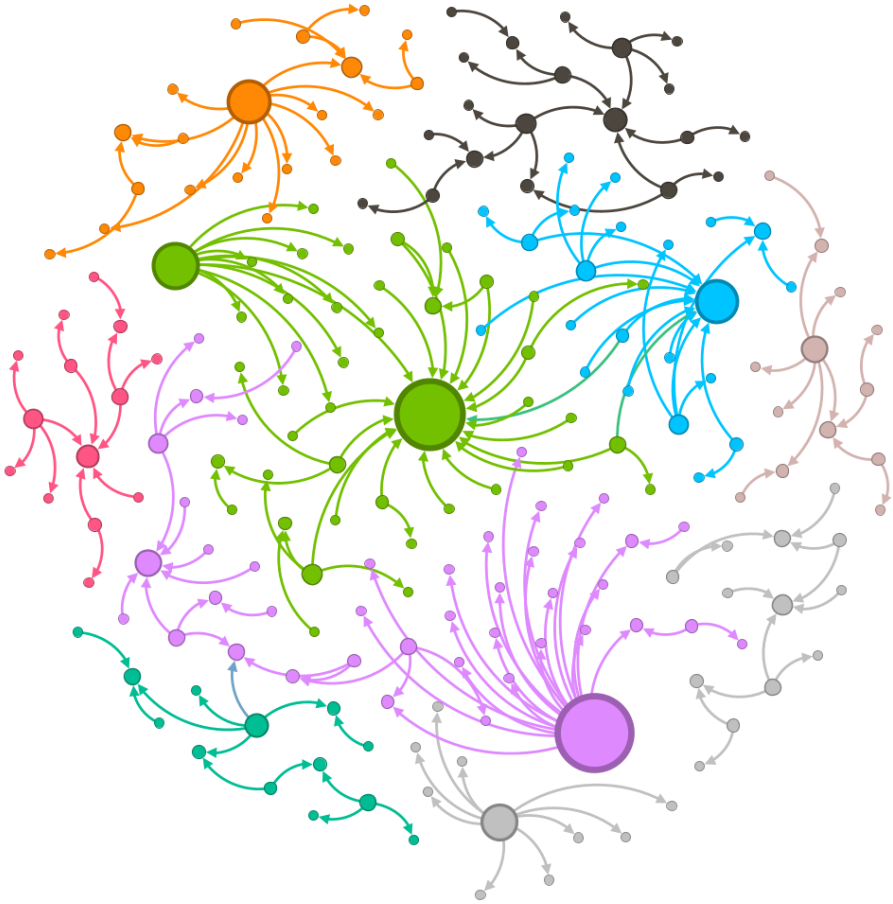
Fonte: Autoria própria (2023).

Figura 23 – Grafo gerado pelas relações de compra e venda no período entre 1 de Janeiro de 2016 a 1 de Março de 2016. Ampliação da região destacada na Figura 14, onde um vendedor atende diversos compradores exclusivamente, podendo ser um indício de anomalia.



Fonte: Autoria própria (2023).

Figura 24 – Grafo gerado pelas relações de compra e venda no período entre 1 de Janeiro de 2016 a 7 de Janeiro de 2016. O grafo destaca as 10 maiores comunidades encontradas no intervalo de tempo utilizado.



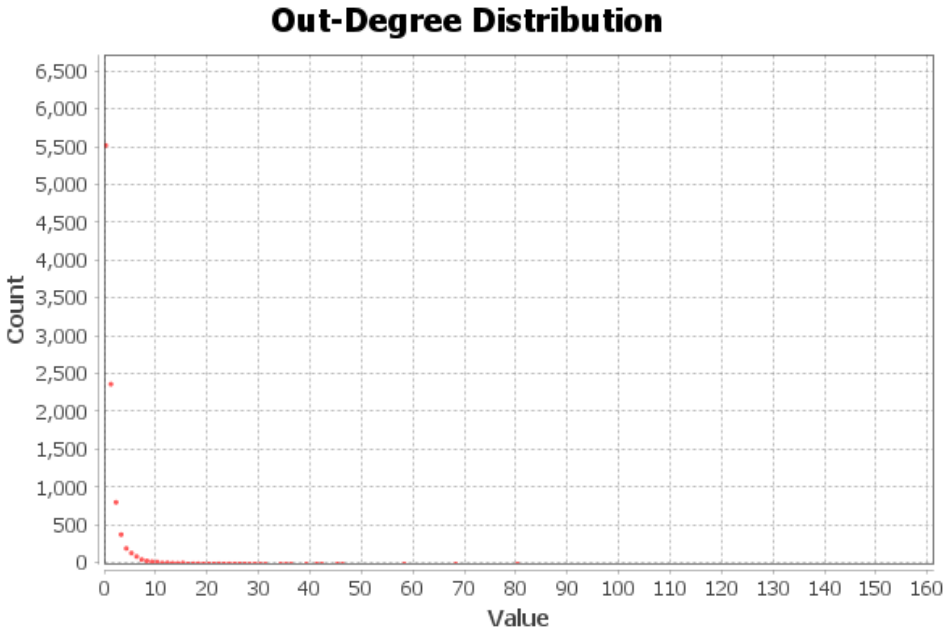
Fonte: Autoria própria (2023).

A análise dos dados utilizando grafo mostrou que os dados variam bastante, onde a maioria dos compradores e vendedores interagem poucas vezes, porém alguns vértices que se destacam devido ao seu alto grau de entrada ou saída e alta medida de centralidade, possivelmente indicando uma anormalidade.

Figura 25 – Grafo gerado pelas relações de compra e venda no período entre 1 de Janeiro de 2016 a 1 de Março de 2016. O grafo destaca as 5 maiores comunidades encontradas no intervalo de tempo utilizado.

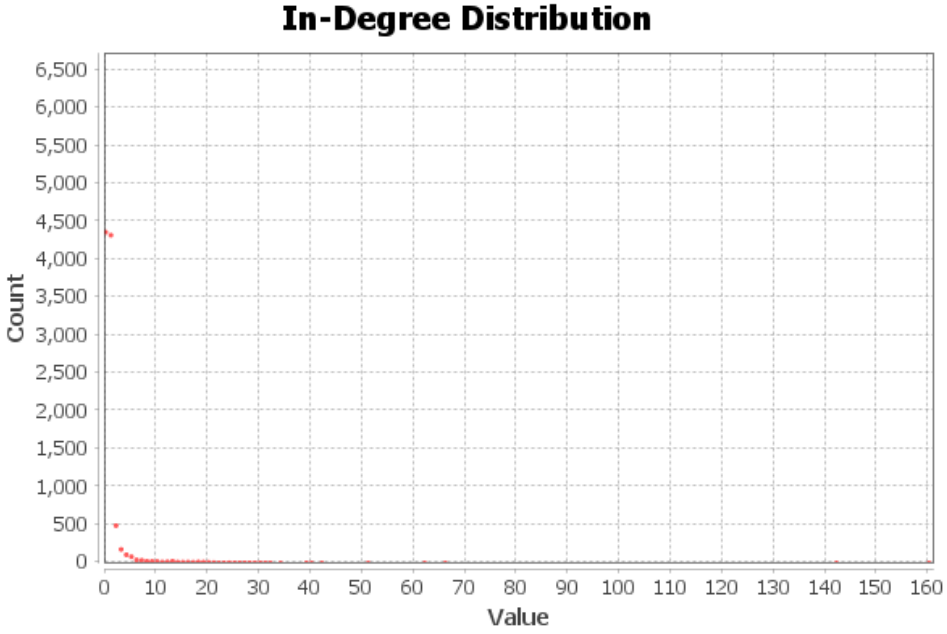
Fonte: Autoria própria (2023).

Figura 26 – Gráfico representando a distribuição de graus de saída entre os vértices no período entre 1 de Janeiro de 2016 a 7 de Janeiro de 2016.



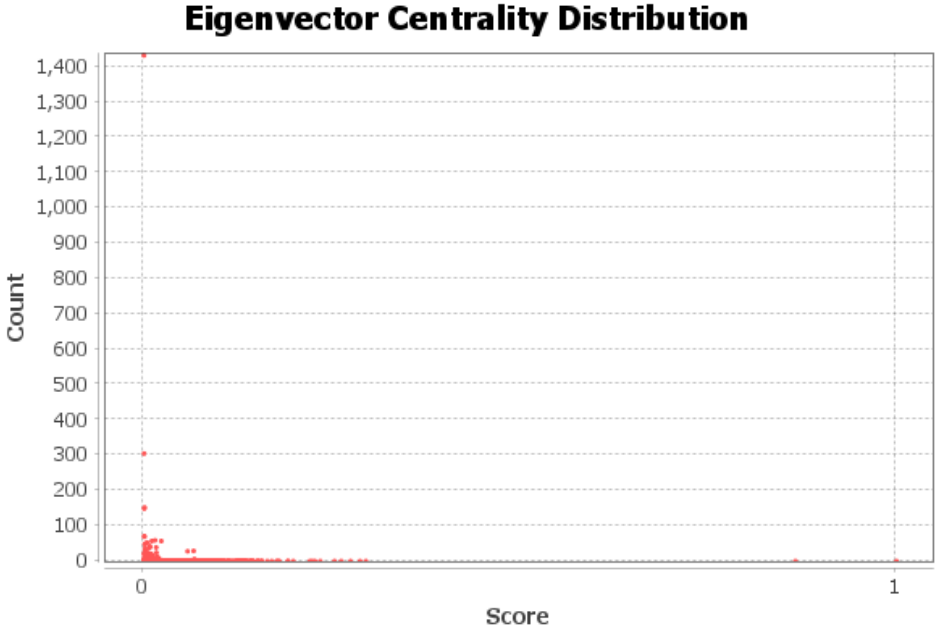
Fonte: Autoria própria (2023).

Figura 27 – Gráfico representando a distribuição de graus de entrada entre os vértices no período entre 1 de Janeiro de 2016 a 7 de Janeiro de 2016.



Fonte: Autoria própria (2023).

Figura 28 – Gráfico representando a distribuição de centralidade de autovetor entre os vértices no período entre 1 de Janeiro de 2016 a 7 de Janeiro de 2016.



Fonte: Autoria própria (2023).

5 RESULTADOS

Esta seção contém a análise resultados encontrados pela aplicação dos métodos das duas frentes. Foi necessário realizar a avaliação manual dos resultados pois dado que os métodos são para o aprendizado não-supervisionado, não é possível avaliá-los utilizando as ferramentas convencionais usadas em métodos de aprendizado supervisionado e semi-supervisionado. Além disso, os autores não contaram com o auxílio de um especialista na área que poderia indicar se os casos apontados pelos algoritmos são, de fato, casos de fraude. Sendo assim, foi utilizado um método de avaliação que parte da análise e comparação manual dos resultados encontrados pelos algoritmos com as demais instâncias similares e então sua categorização em quatro grupos com características distintas, como explicado abaixo. O link para o repositório contendo o código-fonte criado pelos autores e utilizado nos passos de análise exploratória, preparação dos dados, implementação dos métodos e avaliação dos resultados pode ser encontrado no Apêndice A.

5.1 Dados Estruturados

Para avaliar a eficácia dos métodos LOF, iForest e SOM foram escolhidos os 10 *outliers* com maior pontuação encontrados não só por cada modelo individualmente, mas também pela interseção dos resultados de cada algoritmo (instâncias iguais que aparecem nos resultados de dois ou mais modelos). Os resultados foram analisados manualmente pelos desenvolvedores e classificados levando em consideração a potencialidade do registro ser fraudulento. Dessa forma, foram analisados 3 grupos de instâncias contendo os 10 principais resultados de cada método: (i) SOM; (ii) iForest; e (iii) LOF.

As potenciais anomalias foram classificadas em quatro categorias:

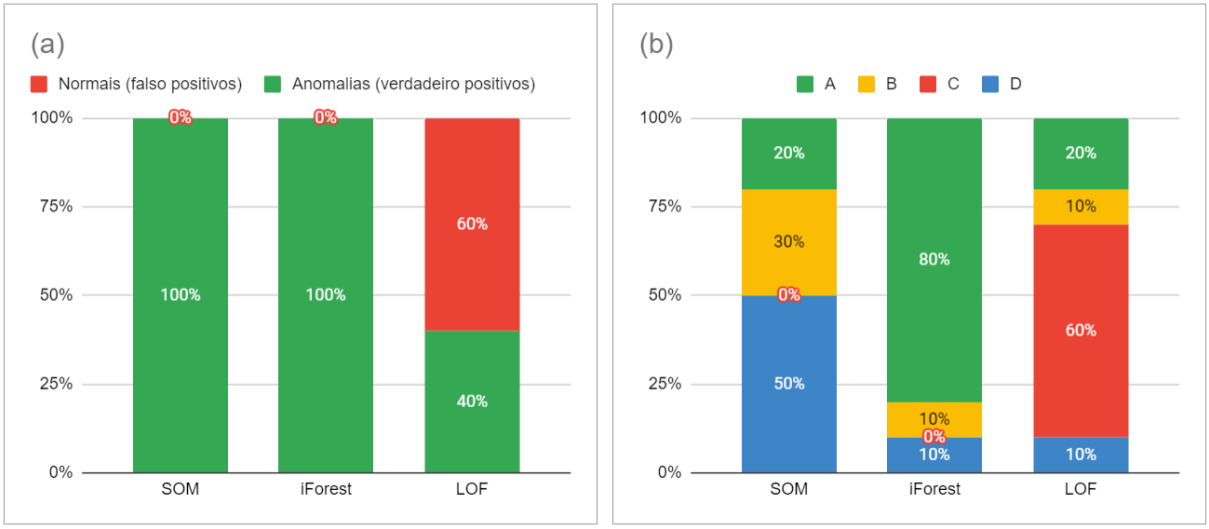
- Categoria A: Potencial fraude (significativo): quando há fortes indícios de que a instância é uma fraude, apresentando valores discrepantes (como quantidades e valor unitário do item) em relação aos demais registros;
- Categoria B: Potencial fraude (moderado): quando há evidências que a instância é uma possível fraude mas não há dados suficientes para realizar uma afirmação segura;
- Categoria C: Potencial não-fraude: quando não há evidências que a instância é uma fraude e os valores encontrados são condizentes com os dos demais registros;
- Categoria D: Indeterminado: quando a instância está isolada e não permite uma comparação com os demais registros.

Dessa forma, pode-se afirmar que as instâncias classificadas como A, B ou D são reais *outliers* e que seus valores são, de fato, incondizentes com os demais registros da base de da-

dos. Porém, é necessário esclarecer que mesmo quando uma instância é uma anomalia, não necessariamente será uma possível fraude. Como indicado pela categoria D há instâncias que são *outliers* mas que não pertencem a *clusters* de dados específicos, impossibilitando determinar se são possíveis fraudes ou não. Além disso, ressalta-se que a análise dos dados foi feita pelos próprios desenvolvedores e não por especialistas da área contábil ou financeira.

A Figura 29(a) apresenta os resultados encontrados para cada grupo em relação a porcentagem de instâncias que, de fato, são anomalias. Para tal, considera-se o número de registros classificados como tipo A, B ou D para Anomalias (verdadeiros positivos) e do tipo C como Normais (falso positivos). A Figura 29(b) apresenta os resultados encontrados para cada grupo e para cada categoria individual.

Figura 29 – (a) Proporção de instâncias anômalas e normais para cada grupo. (b) Proporção de instâncias pertencentes a cada categoria para cada grupo.



Fonte: Autoria própria (2023).

Percebe-se que os 10 principais *outliers* encontrados para o SOM e para o iForest são verdadeiras anomalias. Porém, o iForest apresentou resultados que possuem uma probabilidade mais alta de apresentarem algum tipo de fraude, enquanto que o SOM indicou mais instâncias isoladas ou que não apresentam indícios fortes de serem fraudulentas. No caso do LOF, a maior parte dos resultados foram falso-positivos, ou seja, instâncias que não apresentam nenhuma evidência de serem nem anomalias, nem possíveis fraudes.

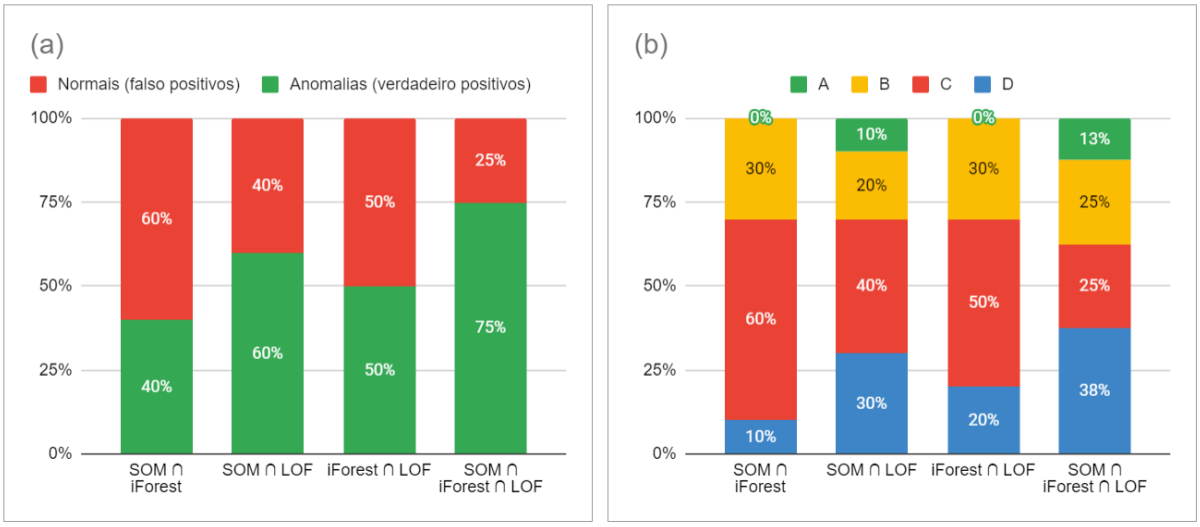
Ainda, é necessário apontar que as instâncias classificadas como sendo do grupo A em geral apresentaram valores unitários mais baixos em comparação com as demais instâncias enquanto que a quantidade era significativamente mais alta, o que pode indicar casos de superdimensionamento. Casos de sobrepreço foram raramente encontrados nas amostras analisadas.

Outra análise realizada foi em relação às instâncias encontradas em dois ou mais conjuntos de anomalias simultaneamente. Estes conjuntos foram gerados a partir dos resultados

encontrados por cada modelo individualmente onde subconjuntos contendo 1% das instâncias com maior *anomaly score* para cada modelo foram criados, originando três listas com 746 instâncias cada. A partir dessas listas foram criados os seguintes novos conjuntos de anomalias que pertencem à interseção entre as listas originais: (iv) SOM e iForest; (v) SOM e LOF; (vi) iForest e LOF; e (vii) SOM e iForest e LOF. Os 10 *outliers* mais significativos de cada conjunto foram escolhidos para análise; apenas para o último grupo (vii) foram analisados apenas 8 registros, que foram os únicos encontrados na interseção de todos os resultados dos modelos. O objetivo de analisar as instâncias que pertencem a dois ou mais modelos é verificar se o sistema de detecção possui mais confiabilidade quando os *outliers* encontrados pertencem a mais de um conjunto.

Para os grupos de instâncias que pertencem a mais de um grupo, observou-se que 28% das instâncias foram iguais entre o SOM e o iForest; 6% foram iguais entre o SOM e LOF; 4% foram iguais entre o iForest e LOF; e 1% foram iguais entre todos os modelos. A Figura 30(a) apresenta a porcentagem de falso-positivos e verdadeiro-positivos para as instâncias analisadas e a Figura 30(b) apresenta a proporção de cada categoria individual.

Figura 30 – (a) Proporção de instâncias anômalas e normais para cada grupo de interseção entre os conjuntos originais. (b) Proporção de instâncias pertencentes a cada categoria para cada grupo de interseção entre conjuntos originais.



Fonte: Autoria própria (2023).

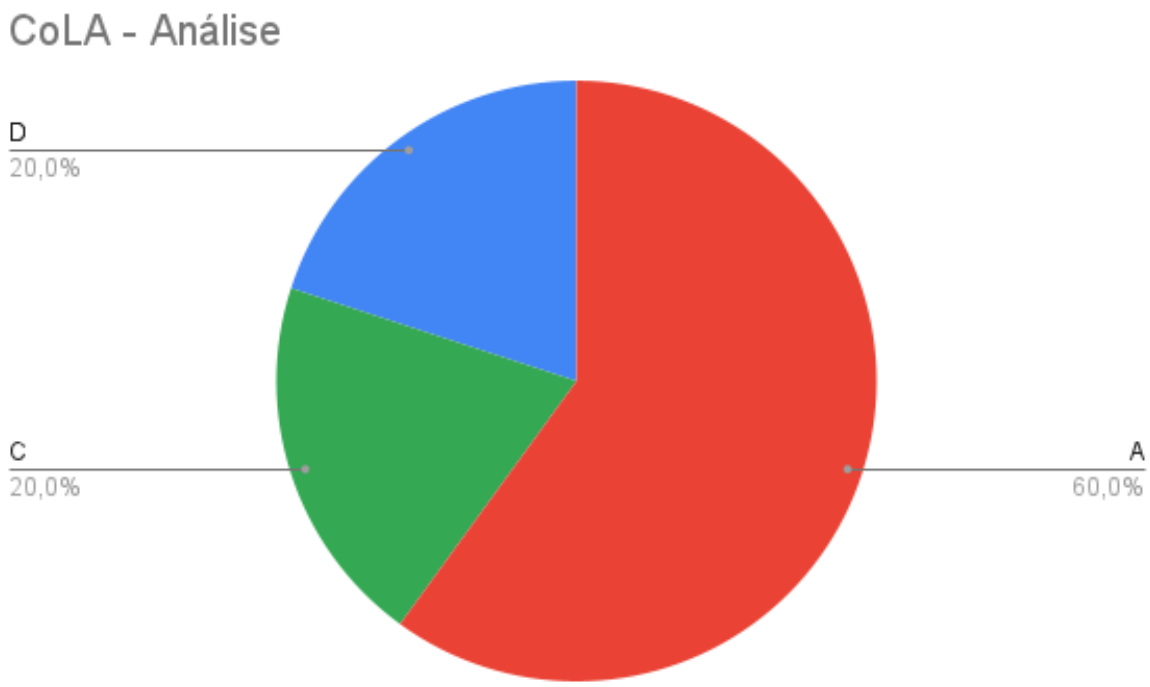
Nenhum dos subconjuntos de interseções apresentaram resultados superiores aos encontrados pela aplicação individual de cada algoritmo. Os conjuntos iv, v e vi exibem proporções expressivas de falso-positivos e, comparativamente, nenhum dos subconjuntos demonstrou uma capacidade satisfatória de apontar possíveis fraudes.

5.2 Grafos

Para avaliar a eficácia dos métodos usados foram escolhidos os 5 nós com os maiores *raw outlier score* encontrados em cada modelo. Os resultados foram analisados manualmente pelos desenvolvedores e classificados em quatro categorias, como descrito anteriormente na sessão 5.1, levando em consideração a potencialidade do registro ser fraudulento. Dessa forma, foram analisados 2 grupos de instâncias contendo os 5 principais resultados de cada método: (i) CoLA; e (ii) GAAN.

A Figura 31 mostra a distribuição dos nós apontados como anômalos pelo algoritmo ColA nas categorias propostas. Foi possível observar que os nós considerados verdadeiramente anômalos (categoria A) possuíam em sua maioria compras efetuadas com o valor bem acima ou abaixo das médias encontradas ao analisar produtos semelhantes, sendo que em alguns casos foi o valor mais alto encontrado. O grau dos nós não pareceu afetar o grau de anomalia atribuído pelo algoritmo, pois alguns nós eram pouco conexos (poucas interações de compra ou venda) e outros bastante conexos (muitas interações de compra ou venda), e todos pertenciam a comunidades diferentes (atribuído pelo algoritmo de modularidade). Os nós classificados na categoria D possuíam muita flutuação de valor dos produtos dificultando a análise, e os nós classificados na categoria C possuíam seus parâmetros condizentes com interações semelhantes no grafo.

Figura 31 – Algoritmo CoLA: Proporção de nós pertencentes a cada categoria.

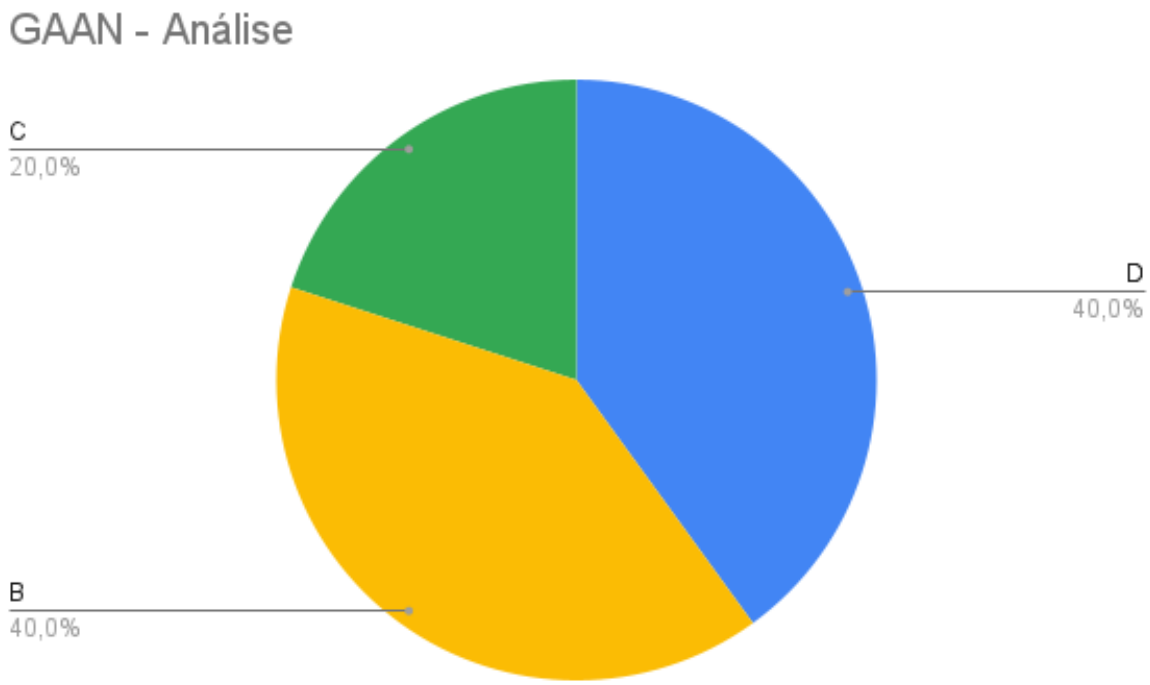


Fonte: Autoria própria (2023).

A Figura 32 mostra a distribuição dos nós apontados como anômalos pelo algoritmo GAAN nas categorias propostas. É possível observar a ausência de nós classificados na cate-

goria A. Isso é devido aos nós apontados como mais anômalos pelo algoritmo GAAN possuírem interações envolvendo produtos com muita flutuação em seus preços ou com poucas interações semelhantes. Também não foram encontradas discrepâncias no grau dos nós e os mesmos pertenciam a diferentes comunidades, dificultando uma comparação mais assertiva dos valores entre interações semelhantes. Os nós classificados na categoria B apresentavam interações de compra onde o valor do produto era o maior ou próximo do maior valor encontrado para produtos semelhantes no conjunto de dados. Os nós classificados na categoria D possuíam grande flutuação no valor do produto comparado a interações semelhantes no conjunto de dados, e os nós classificados na categoria C apresentavam valores condizentes com interações de outros nós semelhantes no conjunto de dados.

Figura 32 – Algoritmo GAAN: Proporção de nós pertencentes a cada categoria.



Fonte: Autoria própria (2023).

5.3 Discussões

Os resultados encontrados na pesquisa apontam que o iForest apresentou melhores resultados para a identificação de possíveis fraudes em licitações em comparação com o SOM e o LOF. Além disso, também foi concluído que os *outliers* que foram encontrados simultaneamente por mais de um algoritmo não possuem uma probabilidade mais alta de serem verdadeiras anomalias nem possíveis fraudes.

Ainda, em comparação com o SOM, o iForest ainda se mostrou mais rápido e mais eficiente no uso de recursos. O primeiro levou minutos para ser treinado e requisitava mais uso de memória, o segundo realizou o processo de treinamento em segundos e não consumiu uma quantidade problemática de memória. Para ser possível treinar o SOM com a quantidade original de dados (mais de 2 milhões de registros) e utilizando um tamanho de mapa ideal, seria necessário utilizar uma quantidade alta de memória e o processo seria mais lento em comparação aos dos demais algoritmos, sendo necessário fazer o uso de máquinas com maior capacidade de processamento e mais memória do que as disponibilizadas para a realização do trabalho. O LOF, assim como o iForest, apresentou um desempenho melhor que o SOM, porém o problema principal para o treinamento desse algoritmo está na definição do número ideal de vizinhos mais próximos. Os autores originais do método sugerem utilizar um valor baseado na quantidade de instâncias dos *clusters* do conjunto de dados. Apenas com informações mais aprofundadas desse dado seria possível inferir um número ideal de vizinhos. Porém, para o caso da base de dados utilizada para este trabalho, não é possível identificar a forma com que os *clusters* são organizados. Mesmo com o uso de técnicas para analisar a topologia dos dados, os desenvolvedores tiveram dificuldade em determinar um valor apropriado para ser utilizado neste método pois os conjuntos de pontos de dados de um item específico se mostraram pequenos, mas podem facilmente se mesclar com outros *clusters*.

Os resultados encontrados na pesquisa ao analisar os algoritmos aplicados ao grafo mostram que o CoLA apresentou resultados melhores quando comparado ao algoritmo GAAN, pois identificou nós com maiores probabilidades de serem anômalos devido às suas diferenças em relação aos nós semelhantes. Também podemos observar que o grau dos nós e a comunidade na qual pertencem não aparentam contribuir em grande parte com o aumento ou diminuição da probabilidade do nó ser considerado um *outlier*. Em relação o desempenho, os dois algoritmos foram semelhantes, onde ambos levaram alguns minutos para serem treinados e a quantidade de memória que utilizam impediu que os desenvolvedores utilizarem o conjunto de dados completos.

6 CONCLUSÃO E TRABALHOS FUTUROS

Os resultados obtidos ao longo deste trabalho elucidam a suposição que, a partir do uso de métodos de aprendizado de máquina e da análise das redes complexas de interações de compra, é possível descobrir possíveis casos de fraude de forma a agilizar seu processo de detecção juntamente a um auditor. A utilização do grafo para rede de compras se mostrou eficaz para representar e possibilitar a análise de anomalias estruturais e a mudança das relações no decorrer do tempo. Os nós do grafo foram utilizados para representar as empresas e suas informações, como a longitude e latitude. As arestas do grafo representam as interações de compra e venda, incluindo informações sobre os produtos que compunham a transação, seus valores e a data, possibilitando representar o caráter mutável das interações. O objetivo de avaliar a eficácia dos modelos estudados em sua capacidade de detectar anomalias foi alcançado, possibilitando localizar algoritmos úteis para a resolução do problema e determinar a qualidade de seus resultados. Dos métodos utilizados, destacam-se o iForest e o CoLA, que se mostraram mais promissores na detecção de possíveis instâncias fraudulentas em comparação aos demais métodos apresentados aqui. Conclui-se que estes métodos têm a capacidade de detectar anomalias no contexto financeiro e podem ser utilizados em organizações como uma ferramenta auxiliar na detecção de fraudes. Porém, é necessário ressaltar a importância da análise humana, propriamente de um auditor especializado, para validar os resultados encontrados pelos modelos. Os algoritmos servem como instrumentos para agilizar o processo e detectar divergências possivelmente imperceptíveis, mas para determinar a veracidade dos resultados é fundamental analisar os contextos a partir do ponto de vista humano, levando em conta todas as sutilezas e complexidades contextuais do sistema financeiro.

6.1 Pontos de Melhoria Para Trabalhos Futuros

Foram identificados cinco possíveis pontos de melhoria para a metodologia:

1. Encontrar uma forma de melhor identificar os itens da base de dados: apesar da base apresentar o valor do NCM, que é um identificador do tipo de produto do item, este valor agrega um conjunto alto de produtos diferentes, mas que são do mesmo tipo. Se houvesse uma forma de identificar os produtos com uma menor granularidade, permitindo atingir um nível de detalhamento maior dos tipos presentes, os modelos poderiam utilizar mais uma variável contextual, permitindo compreender melhor como os *clusters* de dados são organizados e atingindo melhores resultados.

2. Ampliar as variáveis de contextualização: para permitir que os contextos das instâncias sejam melhor analisados, é necessário incluir variáveis socioeconômicas como a população e o PIB das cidades a fim de promover uma análise mais completa do con-

texto dos resultados encontrados e permitir procurar justificativas para certos valores anômalos encontrados na base.

3. Compreender mais profundamente os dados originais: com o auxílio de especialistas na área, entender melhor quais campos do conjunto de dados são utilizados por fiscais para identificar uma possível ação fraudulenta, no intuito de melhor modelar o conjunto de dados para o treinamento dos modelos.

4. Utilizar um número maior de amostras na etapa de avaliação: como mencionado, na etapa de avaliação foram utilizados os 10 *outliers* mais significativos para dados estruturados e os 5 *outliers* mais significativos para grafos encontrados por cada algoritmo. Sendo assim, para melhor avaliar os resultados seria necessário analisar uma amostra maior de dados.

5. Avaliar os resultados com o auxílio de especialistas na área: os resultados encontrados pelos modelos foram avaliados pelos próprios desenvolvedores que, por sua vez, não possuem o conhecimento necessário para avaliar se as instâncias anômalas são, de fato, possíveis fraudes. Para avaliar a qualidade dos resultados encontrados seria necessário obter a ajuda de especialistas do setor contábil e financeiro.

REFERÊNCIAS

AKOGLU, L.; TONG, H.; KOUTRA, D. Graph based anomaly detection and description: a survey. **Data Mining and Knowledge Discovery**, 2015. Acessado em: 28 Maio 2023. Disponível em: https://link.springer.com/article/10.1007/s10618-014-0365-y.

AL-HASHEDI, K.; MAGALINGAM, P. Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019. **Computer Science Review**, 2021. Acessado em: 27 Maio 2023. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S1574013721000423.

BRASIL. Lei federal nº 8.666/93 art. 90, de 21 de junho de 1993. **Diário Oficial [da] República Federativa do Brasil**, 1993. Acessado em: 03 Junho 2023. Disponível em: https://www.planalto.gov.br/ccivil_03/leis/l8666cons.htm.

BRASIL. Ncm. 2019. Disponível em: https://www.gov.br/receitafederal/pt-br/assuntos/aduana-e-comercio-exterior/classificacao-fiscal-de-mercadorias/ncm.

BRASIL. Lei federal nº 14.133/21 art. 90, de 1 de abril de 1993. **Diário Oficial [da] República Federativa do Brasil**, 2021. Acessado em: 25 Outubro 2023. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2019-2022/2021/lei/l14133.htm.

BREUNIG, M. *et al.* Lof: Identifying density-based local outliers. **SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data**, 2000. Acessado em: 15 Outubro 2023. Disponível em: https://dl.acm.org/doi/abs/10.1145/342009.335388.

BRZEZINSKA, A.; HORYN, C. Self-organizing map algorithm as a tool for outlier detection. **Procedia Computer Science**, 2022. Acessado em: 27 Maio 2023. Disponível em: https://www.sciencedirect.com/science/article/pii/S1877050922011620.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM Computing Surveys**, v. 41, n. 3, 2009. Acessado em: 27 Maio 2023. Disponível em: https://dl.acm.org/doi/10.1145/1541880.1541882.

CHEN, Z. *et al.* Generative adversarial attributed network anomaly detection. 2020. Acessado em: 15 Novembro 2023. Disponível em: https://dl.acm.org/doi/abs/10.1145/3340531.3412070.

GOMES, Y. Fraude em licitação no regime militar de 1964. **Anais do IV Simpósio de História do Direito**, 2017. Acessado em: 03 Junho 2023. Disponível em: https://www.uemg.br/images/pdfs-noticias/diamantina-anais-iv-simposio-historia-direito.pdf.

GUPTA, M. *et al.* Integrating community matching and outlier detection for mining evolutionary community outliers. **Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining**, 2012. Acessado em: 29 Maio 2023. Disponível em: https://web.cs.ucla.edu/~yzsun/papers/kdd12_rt110_gupta.

HAMELERS, L. Detecting and explaining potential financial fraud cases in invoice data with machine learning. **University of Twente**, 2021. Acessado em: 27 Maio 2023. Disponível em: http://essay.utwente.nl/85533/.

HUANG, D. *et al.* Codetect: Financial fraud detection with anomaly feature detection. **IEEE Access**, 2019. Acessado em: 29 Maio 2023. Disponível em: https://ieeexplore.ieee.org/document/8325544.

HUANG, S.-Y.; TSAIH, R.-H.; YU, F. Topological pattern discovery and feature extraction for fraudulent financial reporting. **Expert Systems with Applications**, 2014. Acessado em: 27 Maio 2023. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0957417414000244.

LIANG, J.; PARTHASARATHY, S. Robust contextual outlier detection: Where context meets sparsity. **arXiv.org**, 2016. Acessado em: 27 Maio 2023. Disponível em: https://arxiv.org/abs/1607.08329.

LIU, F.; TING, K.; ZHOU, Z.-H. Isolation forest. **2008 Eighth IEEE International Conference on Data Mining**, 2008. Acessado em: 27 Maio 2023. Disponível em: https://ieeexplore.ieee.org/document/4781136.

LIU, K. *et al.* Pygod: A python library for graph outlier detection. 2022.

LIU, Y. *et al.* Anomaly detection on attributed networks via contrastive self-supervised learning. 2021. Acessado em: 15 Novembro 2023. Disponível em: https://arxiv.org/abs/2103.00113.

LOPES, A. *et al.* O superfaturamento está definido na lei nº 14.133/2021, e agora? 2021. Acessado em: 15 Outubro 2023. Disponível em: https://www.ibraop.org.br/wp-content/uploads/2021/05/Superfaturamento-e-agora3-Alan-Lopes-Alexandre-Raupp-Rafael-Magro-Regis-Signor-PF.pdf.

MOLLOY, I. *et al.* Graph analytics for real-time scoring of cross-channel transactional fraud. **Financial Cryptography and Data Security**, 2017. Acessado em: 29 Maio 2023. Disponível em: https://link.springer.com/chapter/10.1007/978-3-662-54970-4_2.

MONGIOVI, M. *et al.* Netspot: Spotting significant anomalous regions on dynamic networks. **International Conference on Data Mining**, 2013. Acessado em: 29 Maio 2023. Disponível em: https://epubs.siam.org/doi/epdf/10.1137/1.9781611972832.4.

OLIVEIRA, C. **Governo Bolsonaro: possíveis fraudes durante pandemia de COVID-19 somam R$ 2 bilhões: Transparência Brasil analisou 248 compras e contratações de serviços firmadas entre fevereiro de 2020 e outubro de 2022**. 2022. Acessado em: 04 maio 2023. Disponível em: https://www.brasildefato.com.br/2023/02/13/governo-bolsonaro-possiveis-fraudes-durante-pandemia-de-covid-19-somam-r-2-bilhoes#:~:text=A\%20ONG\%20Transpar\%C3\%AAncia\%20Brasil\%20detectou,2020\%20e\%20outubro\%20de\%202022.

PAULA, E. *et al.* Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering. **2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)**, 2016. Acessado em: 27 Maio 2023. Disponível em: https://ieeexplore.ieee.org/document/7838276.

PEDREGOSA, F. *et al.* Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

POURHABIBI, T. *et al.* Fraud detection: A systematic literature review of graph-based anomaly detection approaches. **Decision Support Systems**, 2020. Acessado em: 28 Maio 2023. Disponível em: https://www.sciencedirect.com/science/article/pii/S0167923620300580.

PUENTE, B.; AMEIDA, P. **Brasil pode perder mais de R$ 20 bilhões por ano com desvios na saúde. CNN Brasil, Rio de Janeiro. 2021**. 2021. Acessado em: 04 maio 2023. Disponível em: https://www.cnnbrasil.com.br/politica/brasil-pode-perder-mais-de-r-20-bilhoes-por-ano-com-desvios-na-saude/.

RANSHOUS, S. *et al.* Anomaly detection in dynamic networks: a survey. **WIREs Computational Statistics**, 2015. Acessado em: 28 Maio 2023. Disponível em: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.1347.

SCHWINDT, C.; CORAZZA, H. **Princípios Fundamentais e Normas Brasileiras de Contabilidade**. Brasília: CFC, 2008.

SHALAGINOV, A.; FRANKE, K. A new method for an optimal som size determination in neuro-fuzzy for the digital forensics applications. **Lecture Notes in Computer Science**, 2015. Acessado em: 15 Outubro 2023. Disponível em: https://link.springer.com/chapter/10.1007/978-3-319-19222-2_46.

SHAN, Y.; MURRAY, D.; SUTINEN, A. Discovering inappropriate billings with local density based outlier detection method. **AusDM '09: Proceedings of the Eighth Australasian Data Mining Conference - Volume 101**, 2009. Acessado em: 15 Outubro 2023. Disponível em: https://dl.acm.org/doi/10.5555/2449360.2449380.

SPENCE, I.; LEWANDOWSKY, S. Robust multidimensional scaling. **Psychometrika 54, 501–513 (1989)**, 1989. Acessado em: 15 Outubro 2023. Disponível em: https://link.springer.com/article/10.1007/BF02294632.

TORRES, V. Cfop. 2022. Acessado em: 10 Dezembro 2023. Disponível em: https://www.contabilizei.com.br/contabilidade-online/o-que-e-cfop-e-como-usar/.

VEJA, R. **Bolsa Família perdeu R$ 2,6 Bilhões com fraudes: Levantamento inédito mostra o volume de recursos desviado do programa. Funcionários públicos, mortos e até doadores de campanha estão entre os beneficiados**. 2023. Acessado em: 04 maio 2023. Disponível em: https://veja.abril.com.br/brasil/bolsa-familia-perdeu-r-26-bilhoes-com-fraudes.

VELASCO, R. B. *et al.* A decision support system for fraud detection in public procurement. **International Transactions in Operational Research**, 2020. Acessado em: 29 Maio 2023. Disponível em: https://www.researchgate.net/publication/341703812_A_decision_support_system_for_fraud_detection_in_public_procurement.

VETTIGLI, G. **MiniSom: minimalistic and NumPy-based implementation of the Self Organizing Map**. 2018. Acessado em: 15 Outubro 2023. Disponível em: https://github.com/JustGlowing/minisom/.

APÊNDICE A – Repositório de Códigos

Todos os códigos utilizados para a realização deste trabalho podem ser encontrados no repositório do GitHub via este link: https://github.com/breno-abreu/TCC_Abreu_Pereira.git

O repositório conta com os códigos utilizados para realizar a análise exploratória dos dados, a limpeza de dados, a implementação dos algoritmos LOF, iForest, SOM, CoLA e GAAM e o código utilizado para realizar a avaliação manual dos resultados encontrados.

Para baixar os arquivos, acesso o link acima, pressione "Code" e então "Download ZIP".