

Utilizando aprendizado por representação para a classificação de laços sociais da IoT

Jamisson J. Júnior^{1,*}, Thiago S. Figueiredo^{1,*},
Ramon Lopes², Luiz C. B. Torres¹, Bruno P. Santos¹

¹Departamento de Computação e Sistemas
Universidade Federal de Ouro Preto (UFOP) – MG – Brasil.

²Sistemas Elétricos e Computacionais
Universidade Federal do Recôncavo da Bahia (UFRB) – BA – Brasil.

{jamisson.junior, thiago.figueiredo}@aluno.ufop.edu.br

rlopes@ufrb.edu.br, {luiz.torres, bruno.ps}@ufop.edu.br

*Estes autores contribuíram igualmente para este trabalho.

Abstract. *The Internet of Things (IoT) has been marked by the interactions between devices that cooperate in a common goal. From this connected cyberphysical environment, the Social IoT (SIoT) is a possible derived paradigm, where several types of relationship and trustworthiness can be established among devices. In this scenario, we address the issues of modeling social ties in IoT and proposing models to classify and predict ties in SIoT. Those tasks are important for example to create privacy rules based on link's type. This paper proposes using learning representation to classify different types of social ties between IoT devices. In order to do this, we use as classification strategies either Graph Neural Networks (GNN) or Traditional Classification Algorithms (TCA). In our experiments, GNN is fast to train and shows F1- $\{\text{macro, micro}\}$ scores of 0.61 and 0.88, respectively. By using TCA, the train step is $121 \times$ up to $11.235 \times$ slower than GNN, but it presents F1-scores of 0.86 and 0.95, respectively.*

Resumo. *A Internet of Things (IoT) tem sido marcada pelas interações entre dispositivos que cooperam para realizar atividades. A partir deste ambiente cibernético e conectado, um possível paradigma derivado é o Social IoT (SIoT), onde múltiplos tipos de relacionamentos e confiabilidade podem ser estabelecidos entre dispositivos. Neste cenário, abordamos as questões de como modelar laços sociais em IoT e na proposição de modelos para, automaticamente, classificar e prever relações em SIoT. Este artigo propõe a utilização de aprendizado por representação para classificar diferentes tipos de laços sociais em SIoT. Para isso, utiliza-se como estratégias para classificação Graph Neural Networks (GNN) ou Algoritmos Tradicionais de Classificação (ATC). Em nossos experimentos, GNN é rápido na etapa de treinamento e apresenta métricas F1- $\{\text{macro, micro}\}$ de 0.61 e 0.88, respectivamente. Ao usar ATC, o treinamento é $121 \times$ até $11.235 \times$ mais lento que GNN, ao passo que as métricas F1-score alcançam 0.86 e 0.95, respectivamente.*

1. Introdução

A Internet das Coisas (do inglês *Internet of Things (IoT)*) tradicional engloba diversas tecnologias e dispositivos que já estão presentes em nosso cotidiano, os quais usam a es-

estrutura de endereçamento da Internet e protocolos de comunicação. Essas características possibilitam que objetos interajam uns com os outros e colaborem por objetivos em comum. Com isso, surge o paradigma Internet das Coisas Sociais (do inglês *Social IoT (SIoT)*) [Atzori et al. 2012b] que traz a ideia de aplicar princípios das redes sociais à IoT, justamente para explorar o potencial das relações entre dispositivos.

Dessa forma, integrar os conceitos de redes sociais e IoT podem levar a várias vantagens [Nitti et al. 2014, Santos et al. 2018, Santos et al. 2016, Marche and Nitti 2020]. Primeiro, a estrutura da SIoT pode ser moldada para se obter navegabilidade na rede tal como feito de modo escalável em redes sociais humanas, o que é importante para IoT na descoberta de dispositivos, serviços e recursos de maneira distribuída [Aljubairy et al. 2020]. Segundo, é possível estabelecer graus de confiança a depender das interações de “amizade” entre os dispositivos [Marche et al. 2020]. Por fim, modelos e conhecimentos consolidados das redes sociais podem ser reutilizados nas soluções para IoT [Wei et al. 2018].

Neste sentido, existem ao menos dois desafios na SIoT que podem ser explorados: a identificação dos diferentes tipos de laços sociais que podem existir e a sua classificação. A complexidade desses desafios está na falta de consenso na definição dos tipos de laços sociais na SIoT e na dinamicidade em que nós (dispositivos IoT) e laços (relacionamentos) aparecem e desaparecem ao longo do tempo. Dessa forma, os autores em [Marche et al. 2018] propuseram ao menos quatro tipos de relacionamentos sociais que os dispositivos poderiam realizar uns com os outros. Esses relacionamentos podem ser baseados em regras estáticas, isto é, dispositivos que fazem laços sociais caso possuam as mesmas especificações físicas ou o mesmo proprietário. Além disso, as relações podem ser baseadas em regras dinâmicas como a localização, duração e quantidade de encontros entre os dispositivos. Por isso, propostas de soluções para esses desafios são importantes para, por exemplo, classificar os tipos de relações e, a partir delas, projetar regras de proteção à privacidade [Marche and Nitti 2020], prever demandas ou eventos na rede social de dispositivos [Aljubairy et al. 2020], entre outros.

Vale ressaltar que no contexto cibernético da SIoT, os dispositivos podem ser definidos em termos de suas conexões, o que naturalmente pode ser expressado como um grafo. Recentes avanços na área de aprendizado de representação (do inglês *representation learning*) tem possibilitado o processamento de dados representados no domínio dos grafos através de técnicas como *node embeddings* e *Graph Neural Networks (GNN)* [Scarselli et al. 2008]. Com essas técnicas, é possível, por exemplo, realizar a classificação, agrupamento e predição dos nós e arestas de um grafo. Como resultado, vemos florescer diversas aplicações em diferentes domínios: na biologia, detecção de *fake news*, sistemas de transporte inteligentes, entre outros [Sanchez-Lengeling et al. 2021]. Entretanto, o uso de *node embeddings* e GNNs para classificação das relações da SIoT é pouco explorado.

Neste artigo, propomos a utilização de aprendizado por representação para classificação dos tipos de relações sociais da IoT. O algoritmo criado recebe como entrada um grafo representando a rede social IoT. Em seguida, a entrada é processada utilizando GNN ou *node embeddings* combinado com Algoritmos Tradicionais de Classificação (ATC). Como saída, é produzido um modelo para classificação dos enlaces sociais ainda

não conhecidos. Em nossos experimentos, usamos como base de dados¹ a interação entre 16.216 dispositivos da IoT dispostos na cidade de Santander, localizada na Espanha, e comparamos com quatro diferentes técnicas de aprendizado de máquina. Como resultado, GNN se mostrou mais rápido na etapa de treinamento e classificação, enquanto apresenta métricas $F1 - \{macro, micro\}$ 0.61 e 0.88, respectivamente. Ao usar ATC, o treinamento e classificação estão entre 121 e 11.235 vezes mais lento que GNN, ao passo que as métricas $F1 - \{macro, micro\}$ alcançam maiores valores que estão entre 0.92 e 0.95, respectivamente.

Em resumo, as contribuições desse artigo são:

1. A utilização de aprendizado por representação para classificar e prever os tipos de laços da SIoT, podendo aplicar as técnicas:
 - *Graph Neural Networks* que corresponde a um tipo de Rede Neural capaz de operar sobre a estrutura de grafos;
 - *Node embedding* combinado com Algoritmos Tradicionais de Classificação, tais como, *Logistic Regression*, *Random Forest* e *Support Vector Machine*.
2. Os dados processados das relações SIoT e modelos estão disponibilizados em repositório online², possibilitando a geração de *insights* e trabalhos futuros.

O restante do artigo está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados. Na Seção 3, definimos o problema e descrevemos as abordagens de classificação utilizadas. Os ambiente de experimentação e a base de dados são apresentados na Seção 4. Os resultados são discutidos na Seção 5. Por fim, Seção 6 apresenta as considerações finais.

2. Trabalhos relacionados

2.1. Internet das Coisas Sociais e os tipos de relações entre dispositivos

SIoT foi primeiramente formalizada por [Atzori et al. 2012a], em que explorou-se a capacidade dos dispositivos da IoT em formar redes sociais de forma autônoma baseada em regras dos seus proprietários. Após isso, pesquisas se concentraram em identificar políticas, métodos e técnicas para o estabelecimento de relação entre os dispositivos sem a necessidade de intervenção humana [Aljubairy et al. 2020].

Nesse sentido, [Marche et al. 2018] propuseram 4 tipos de relacionamentos sociais que os dispositivos poderiam realizar uns com os outros, vide Tabela 1. Eles levaram em consideração não apenas que os dispositivos podem ser móveis (e.g. carros) ou estáticos (e.g. impressora), como também se pertencem a um domínio público ou são bens privados de um usuário. Além disso, esses relacionamentos podem ser baseados em regras estáticas, como dispositivos que fazem laços sociais caso possuam as mesmas especificações físicas, *POR*, ou sejam do mesmo proprietário, *OOR*. Por outro lado, as relações *C-LOR* e *SOR* utilizam regras dinâmicas como a localização, duração e quantidade de encontros entre os dispositivos. De acordo com [Marche et al. 2020], as relações *POR* e *OOR* são capazes de criar pequenas comunidades altamente conectadas dentro da

¹Disponível em: <http://www.social-iot.org/index.php?p=downloads>.

²Disponível em: <https://github.com/JJader/sioties>

Tabela 1. Principais relações sociais da SIoT [Marche et al. 2018]

Relação IoT	Tipo de relação entre dispositivos:	Grupo de dispositivos	Estática (E) / Dinâmica (D)
<i>Parental object relationship (POR)</i>	Do mesmo tipo, modelo e marca;	Todos	E
<i>Ownership object relationship (OOR)</i>	Pertencentes ao mesmo proprietário;	PE, PM, PubE	D
<i>Co-location object relationship (C-LOR)</i>	Frequentemente no mesmo local;	PE, PM, PubE	D
<i>Social object relationship (SOR)</i>	e se encontram de forma intermitente, Em que seus donos entram em contato durante a rotina diária;	PM	D
<i>SOR₂</i>	Variante da SOR que estabelece conexões entre dispositivos público móveis e privado móveis.	PM, PubM	D

PM - Privados Móveis; PE - Privados estáticos
 PubE - Público Estático; PubM - Público Móveis.

rede SIoT. Em complemento, a C-LOR e a SOR criam conexões entre as comunidades geradas pelas relações OOR e POR.

Vale ressaltar que diversos trabalhos fizeram uso das definições dos tipos de relações proposto por [Marche et al. 2018] para estudar a SIoT. [Aljubairy et al. 2020] utilizaram o relacionamento SOR para predição de futuras conexões entre os dispositivos utilizando séries temporais. Além disso, [Khanfor et al. 2019, Khanfor et al. 2020] manipularam essas relações para identificar comunidades dentro da estrutura social da IoT, para isso, os autores utilizaram tanto algoritmos de agrupamento quanto GNN. Não obstante, [Júnior et al. 2021] analisaram o comportamento dos dispositivos dentro da rede C-LOR, dessa forma, os autores foram capazes de quantificar as interações e os impactos de desempenho de solução de comunicação tolerantes a atraso.

No melhor de nossos conhecimentos, a utilização de modelos de aprendizado de máquina capazes de classificar os relacionamentos entre os dispositivos da SIoT ainda é uma necessidade em aberto dentro literatura. Dessa forma, se faz necessário adaptar novas técnicas de análise de dados e inteligência artificial em busca de gerar modelos capazes de identificar, compreender e predizer padrões sociais de estruturas complexas, sobretudo, das conexões SIoT de grande escala [Khanfor et al. 2020, Atzori et al. 2014].

2.2. Aprendizado de representação para grafos

Recentemente diversas pesquisas vem sendo realizadas para entender como modelos de aprendizado de máquina podem auxiliar no estudo de redes sociais. Paradigmas focados em GNN têm mostrado um alto potencial em predição e classificação de conexões [Khanfor et al. 2020], de forma que parte do aprendizado está em identificar representações de baixa dimensão para os nós da rede. Nessa etapa, duas abordagens podem ser consideradas: o aprendizado transdutivo e o indutivo.

O aprendizado transdutivo consiste em aprender a incorporação de cada nó separadamente, aproveitando padrões baseados em fatoração de matrizes [Gupta et al. 2021] como, por exemplo, o node2vec [Grover and Leskovec 2016]. As principais limitações dessas representações estão em realizar classificações com nós não vistos na etapa de treino e sua complexidade cresce com o tamanho da rede [Oliveau and Sahbi 2018]. Por sua vez, o aprendizado indutivo utiliza funções de agregação simétricas capazes de reconhecer parâmetros globais do grafo [Morris et al. 2019, Corso et al. 2020]. Dessa forma, modelos indutivos como o GraphSage [Hamilton et al. 2017] são capazes de reconhecer a inserção de novos nós.

Estratégias de aprendizado de representação estão sendo utilizadas dentro da SIoT para identificar, compreender e prever padrões sociais. Os autores em [Abdelrahman et al. 2020], por exemplo, focam na extração de dados espaço-temporais de edifícios inteligentes e na representação dessas informações utilizando *graph embeddings*. Esses autores utilizaram dois métodos de aprendizado de representação, a saber, *node2vec* e Spatio-Temporal Attentive RNN (STAR) [Xu et al. 2019]. Ambos os métodos não são capazes de reconhecer o surgimento de novos nós dentro da rede e, dessa forma, são retreinados caso surja dispositivos que não estavam no treinamento.

Em [Xing et al. 2021], os autores propuseram a utilização de Fatoração Não-Negativa de Matrizes para aprender a representações de nós IoT no contexto de Trigger-Action Programming (TAP). Resultados experimentais sugerem que a técnica proposta se mostrou eficaz na classificação de nós. Ainda no contexto de TAP, [Hu et al. 2019] propuseram a utilização de *random walks* no grafo IoT de contexto para aprender representações de nós. Nos experimentos, o método proposto apresentou, em todos os cenários exceto um, os melhores resultados na tarefa de classificação de nós.

É importante salientar que a rede SIoT tem alto potencial de crescimento. Estudos apontam que em 2024 haverá mais de 83 bilhões de conexões IoT [Barker and Rothmuller 2020], um crescimento de 130% se comparado com o ano de 2020. Sendo assim, os *frameworks* projetados nessas áreas precisam ser capazes de atrelar não apenas a precisão no aprendizado dos dados, como também, suprir a natureza de crescimento da rede. Dessa forma, nesse trabalho, utilizamos um modelo GNN indutivo e de aprendizado fim-a-fim, capaz de reconhecer a inserção de novos nós, como também realizar a tarefa de aprendizado de representação e classificação em conjunto, economizando tempo nos dois processos. Além disso, empregamos uma abordagem alternativa utilizando aprendizado de representação transdutivo junto com modelos ATC. Nessa outra abordagem, é possível identificar com maior precisão, sobre as métricas F1-micro e F1-macro, o padrão da rede SIoT sobre o custo de maior tempo de treinamento e classificação.

3. Fluxo de execução

Para o problema em questão, seja C o conjunto de classes existentes para os relacionamentos entre nós em uma rede SIoT. Neste trabalho, um rede SIoT é representada como um grafo $G = (V, E, r)$ com um conjunto V de vértices para representar os nós, um conjunto $E \subseteq V \times V$ de arestas para representar as interações entre os nós da rede e uma função $r : E \rightarrow C$ para mapear uma aresta para uma classe de relacionamento. O problema consiste em, dada uma aresta $e \in E$, prever a classe c da aresta, em outras palavras, precisamos aprender uma função de aproximação da função r .

Neste sentido, este trabalho busca utilizar técnicas de aprendizado de máquina para encontrar a função de aproximação r utilizando o aprendizado de representação da aresta e como entrada para algoritmos de aprendizado de máquina. O fluxo de execução dos algoritmos é ilustrada na Figura 1. Os estágios compreendidos entre 1 e 3 mostram o passo a passo utilizando GNN para gerar o modelo de classificação. Por sua vez, as etapas 1, 4, 5 e 6 ilustram um processamento alternativo, o qual permite que Algoritmos Tradicionais de Classificação sejam utilizados para produzir o modelo de classificação. O processo se inicia na etapa 1, em que as diferentes redes de relacionamentos sociais dos

dispositivos são integradas em um único grafo SIoT, onde: os nós representam os dispositivos e as arestas representam os múltiplos relacionamentos entre o pares de dispositivos.

Após a geração do grafo SIoT na etapa 1, a etapa 2 emprega um modelo GNN para realizar o aprendizado ponta-a-ponta (processo descrito na Seção 3.1), isto é, tanto a etapa de aprendizado de representação quanto a classificação são executadas conjuntamente. Por sua vez, a segunda abordagem emprega o aprendizado de representação na etapa 4 e classificação utilizando ATCs na etapa 5. A seguir, detalhamos as metodologias implementadas em cada abordagem.

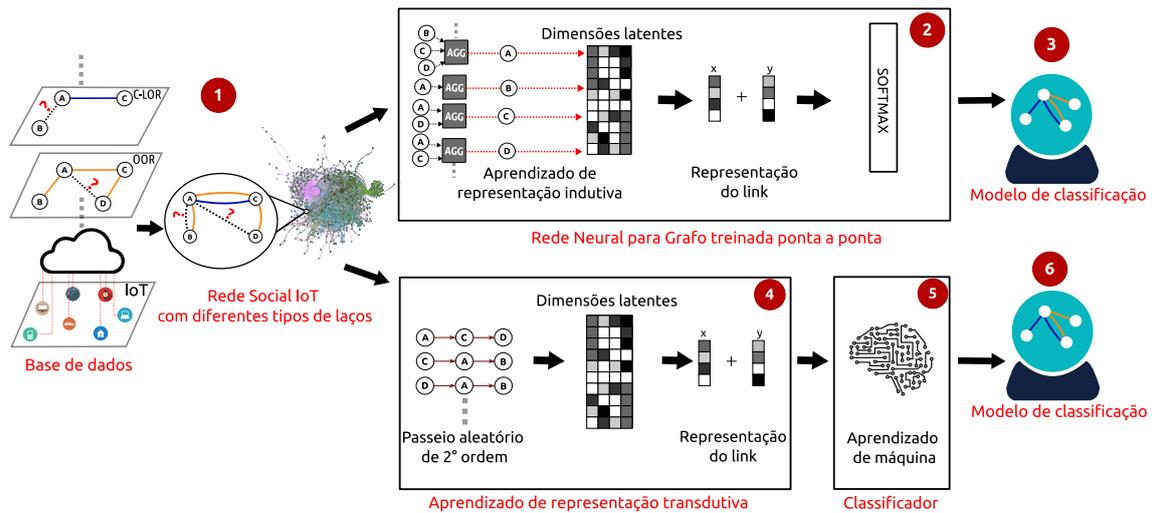


Figura 1. Fluxo de execução. Etapas (1-3) usam GNN para classificar enlaces. Etapas (1, 4-6) utilizam ATCs para realizar a classificação.

3.1. Rede Neural para Grafo treinada ponta-a-ponta

As GNNs compõem uma classe de modelos de *deep learning* que operam diretamente sobre grafos para aprender representações de vértices, arestas ou subgrafos em um novo espaço com dimensão inferior àquela do espaço original [Zhou et al. 2020]. Nesse contexto, GraphSAGE [Hamilton et al. 2017] é um modelo GNN que agrega coletivamente informações da estrutura do grafo através da troca de mensagens entre vértices de uma vizinhança definida pelo projetista da rede, por exemplo vértices que distam k saltos do vértice de interesse. GraphSAGE apresenta três características fundamentais: i) permite a generalização para vértices que não pertencem ao conjunto de treino, facilitando, então, sua aplicação em cenários dinâmicos, como é o caso de redes IoT; ii) permite realizar aprendizado ponta-a-ponta (do inglês *end-to-end*) [Glasmachers 2017], cujo aprendizado ocorre pela aplicação de métodos de gradiente ao sistema como um todo, em outras palavras, todos os módulos do sistema são diferenciáveis com respeito aos parâmetros do modelo a ser treinado; iii) pode utilizar as características dos nós para aprender *embeddings*.

Diferentemente de outros métodos presentes na literatura para aprendizado de representação de grafos [Grover and Leskovec 2016, Perozzi et al. 2014, Tang et al. 2015], por empregar aprendizado ponta-a-ponta, GraphSAGE aprende representações específicas para o problema em questão. Desse modo, conforme ilustrado na Figura 1 etapa 2, em comparação com o aprendizado utilizando ATC (etapas 4

e 5), não há necessidade de treinar separadamente um classificador (etapa 5) a partir das representações aprendidas na etapa anterior (etapa 4).

GraphSAGE fundamenta-se na agregação de informação de vértices vizinhos e, à medida que o algoritmo progride, os vértices obtêm incrementalmente informações sobre nós mais distantes no grafo. Portanto, seja \mathbf{h}_u^k a representação de um vértice u na iteração k e $N(u) = \{v | (u, v) \in E\}$ os vértices adjacentes a u . Inicialmente, cada vértice $u \in V$ agrega as representações dos vértices $v \in N(u)$ em um único vetor $\mathbf{h}_{N(u)}^{k-1}$. Em seguida, o resultado da concatenação dos vetores \mathbf{h}_u^{k-1} e $\mathbf{h}_{N(u)}^{k-1}$ é dado como entrada para um camada densa com uma função de ativação não-linear $\delta(\cdot)$, produzindo, então, a representação $\mathbf{h}_u^k = \delta(W^k[\mathbf{h}_u^{k-1}, \mathbf{h}_{N(u)}^{k-1}])$. Conforme ilustrado na etapa 2 da Figura 1, a representação \mathbf{h}_e de um aresta $e = (u, v) \in E$ é determinada pela soma dos *embeddings* dos vértices u e v , isto é, $\mathbf{h}_e = \mathbf{h}_u^k + \mathbf{h}_v^k$. Por fim, \mathbf{h}_e é fornecido como entrada para uma camada de saída *softmax*, obtendo, então, a classificação para a aresta e .

3.2. Aprendizado de representação transdutiva e classificador

A segunda abordagem de classificação de arestas (etapas 4-6 da Figura 1) leva em consideração dois estágios de aprendizado separados: i) inicialmente, ocorre a aprendizagem transdutiva de representação das conexões; e, ii) logo após um classificador baseado em modelos de ATCs é treinado para a predição dos tipos de relacionamento das arestas. Por se tratar de um método transdutivo, o modelo não é capaz de ser facilmente generalizado para entidades ausentes no treinamento, portanto a inserção de novos nós ou arestas requer o retreinamento (etapas 1, 4 e 5).

A etapa 4 da Figura 1 tem como objetivo aprender representações para os vértices e, em seguida, definir uma representação para as arestas. A saber, o aprendizado de representação busca aprender uma função de mapeamento $f : V \rightarrow \mathbb{R}^d$, em que $f(u_i)$ é o *embedding* do nó $u_i \in V$ e $d \ll |V|$. Para esse fim, utilizamos o *node2vec* [Grover and Leskovec 2016], o qual emprega uma estratégia *self-supervised learning*. Sucintamente, *node2vec* utiliza a aproximação do passeio aleatório de segunda ordem (*biased random walk*) para gerar a estrutura local de vizinhanças de cada nó. Por fim, modela-se a probabilidade de representação vetorial de cada nó dado a vizinhança que ele pertence. Para isso, utiliza-se uma função de distribuição multinomial medida pela similaridade de cosseno dos nós [Gupta et al. 2021]. Com base nas representações aprendidas para os nós, definimos a representação \mathbf{h}_e de uma aresta $e \in E$ a soma dos *embeddings* dos vértices u e v , isto é, $\mathbf{h}_e = f(u) + f(v)$. A etapa final desse processo é o treinamento do modelo baseado em ATCs utilizando a representação de arestas, ou seja, utilizando \mathbf{h}_e .

4. Experimentos

Nossos experimentos foram projetados para responder as seguintes questões:

- Q1. É possível classificar os tipos de laços sociais em SIoT?
- Q2. Como a escolha da abordagem afeta o desempenho?

Para responder a essas questões, utilizamos uma base de dados disponível na literatura e vários modelos de classificação em nossos experimentos conforme detalhado a seguir.

4.1. Base de dados

A base de dados utilizada em nossos experimentos modela a interação entre 16.216 dispositivos IoT reais e sintéticos dispostos na cidade de Santander, localizada na Espanha (vide Tabela 1). Os dispositivos são divididos entre móveis (e.g. carros, celulares), e estáticos (e.g. impressoras, sensores de casa); 14.600 dispositivos pertencem a usuários privados enquanto os demais pertencem a serviços públicos. Por fim, a base de dados disponibiliza um conjunto de *features*: geo-localização, identificador do usuário proprietário e, por fim, especificações técnicas (e.g. tipo, modelo e marca do dispositivo).

Processamos a base de dados para construir uma rede de relacionamentos para cada tipo de conexão IoT, cujos rótulos são OOR, C-LOR, SOR e SOR₂; esses rótulos compõem o conjunto C com base na definição do problema. Os relacionamentos são baseados no identificador do proprietário e na localização geográfica dos dispositivos. Dessa forma, a rede de relacionamentos para cada tipo de conexão é modelada como um grafo não direcionado e sem pesos, cujos nós são os dispositivos IoT e as arestas representam os relacionamentos. As quatro redes de relacionamento obtidas foram, então, unidas em uma única rede (grafo) SIoT; vale ressaltar que um par de nós pode possuir mais de um tipo de relacionamento, resultando, então, em um multigrafo. Em seguida, esse multigrafo foi transformado em um grafo simples, em que há no máximo uma aresta entre cada par de vértices. Para cada combinação possível entre os $|C|$ rótulos existentes, criamos um novo rótulo no grafo simples. Por exemplo, se, entre um par de vértices, houver apenas duas arestas com rótulos OOR e C-LOR, um novo rótulo é criado para representar esses dois relacionamentos no grafo simples, por exemplo OOR+C-LOR.

A base de dados foi dividida aleatoriamente em 70% para treino, 20% validação e 10% teste. Essa divisão foi realizada de forma a manter aproximadamente a mesma frequência das classes nos três conjuntos, isto é, empregamos um particionamento estratificado. Desse modo, seis classes, duas delas resultantes da combinação de dois tipos de relacionamento, estão presentes na base de dados com sua porcentagem indicada entre parênteses: OOR (50%); C-LOR (20%); SOR₂ (18%); SOR (7%); C-LOR+OOR (4.5%); C-LOR+SOR (0.5%).

4.2. Metodologia de avaliação

Para a avaliação dos resultados, utilizamos a métrica *F1 score*, a qual é definida como a média harmônica entre as métricas precisão (*precision*) e revocação (*recall*). Uma vez que a base de dados apresenta um desbalanceamento entre classes, reportamos as versões *micro* e *macro* desta métrica. Enquanto a versão *macro* calcula a métrica F1 para cada classe separadamente e, então, calcula a média aritmética entre esses valores, a versão *micro*, por sua vez, calcula a métrica de forma global.

4.2.1. Classificadores e calibração de hiperparâmetros

Em nossos experimentos, para evitar o vazamento de informações (do inglês *data leakage*), os hiperparâmetros foram calibrados com base no resultado da métrica *F1-micro* obtido utilizando o conjunto de validação. Por fim, reportamos os resultados do conjunto teste com base na melhor configuração de hiperparâmetros para cada classificador. A seguir, discutimos a metodologia para calibração de hiperparâmetros de cada um dos

classificadores utilizados. Utilizamos a técnica *grid search* para determinar a melhor combinação de hiperparâmetros

Classificador GNN. Para sua implementação, utilizamos o modelo GraphSAGE da biblioteca DGL³, em que escolhemos a função média como função agregadora e, a função ReLU como função de ativação. A arquitetura da rede neural possui uma camada de entrada com d dimensões, h camadas ocultas de dimensão $2d$ exceto a última camada oculta que possui dimensão d e, por fim, uma camada *softmax* de saída. Para fins de treinamento, utilizamos o algoritmo de otimização Adam. Para calibração de hiperparâmetros, nós investigamos valores para $d \in \{8, 16, 32, 64\}$, $h \in \{0, 1, 2\}$, fator de decaimento $wd \in \{1e-5, 1e-4, 1e-3, 1e-2\}$ e taxa de aprendizado $\alpha \in \{1e-5, 1e-4, 1e-3, 1e-2\}$.

Classificadores ATC. Utilizamos o *framework* node2vec⁴ para o aprendizado de representação de vértices do grafo, e os modelos Logistic Regression (LR), Random Forest (RF) e Support Vector Machine (SVM) como classificador. O *framework* node2vec possui dois hiperparâmetros p e q que controlam a largura e a profundidade da caminhada aleatória respectivamente e d dimensões de representação. Nesse sentido, investigamos valores para $p, q \in \{0.5, 1, 2\}$ e $d \in \{8, 16, 32, 64\}$, já os parâmetros *walk length* e *number of walks* do node2vec foram fixados em 80 e 10, respectivamente conforme sugerido em [Hu et al. 2019].

5. Resultados

Nesta seção, respondemos cada um dos questionamentos levantados na Seção 4. O desempenho dos melhores modelos sobre a métrica *F1-score* é apresentado na Figuras 2. A Figura 3 apresenta o desempenho dos classificadores em função da dimensão do *embedding*.

5.1. Classificação dos tipos de laços sociais em SIoT

Para responder a questão Q1, comparamos o desempenho dos classificadores apresentados na Seção 4.2.1 com aquele obtido por um simples classificador aleatório denominado Randômico (RAN). A saber, RAN classifica uma aresta e como classe c de forma aleatória com probabilidade p_c , em outras palavras, a classe prevista c é uma realização de uma distribuição categórica $c \sim \text{Cat}(|C|, p)$, em que $|C|$ representa o número de classes e $p \in [0, 1]^{|C|}$ é um vetor de probabilidades. Em nossos experimentos, p_c é definido pela frequência de arestas da classe c no conjunto de treino.

As Figuras 2a e 2b apresentam as métricas *F1-micro* e *F1-macro*, respectivamente, para cada classificador. É possível identificar que todos os classificadores superam o classificador RAN nas métricas consideradas, com exceção do LR na métrica *F1-micro*. Dessa forma, lembrando a pergunta Q1, os resultados nos permitem concluir que é possível classificar os tipos de laços sociais em redes SIoT.

5.2. Efetividade na classificação

Para responder a questão Q2, avaliamos o desempenho usando GNN ou Algoritmos Tradicionais de Classificação (LR, RF e SVM). Com base nas Figuras 2a e 2b, podemos

³<https://www.dgl.ai/>

⁴<http://snap.stanford.edu/node2vec/>

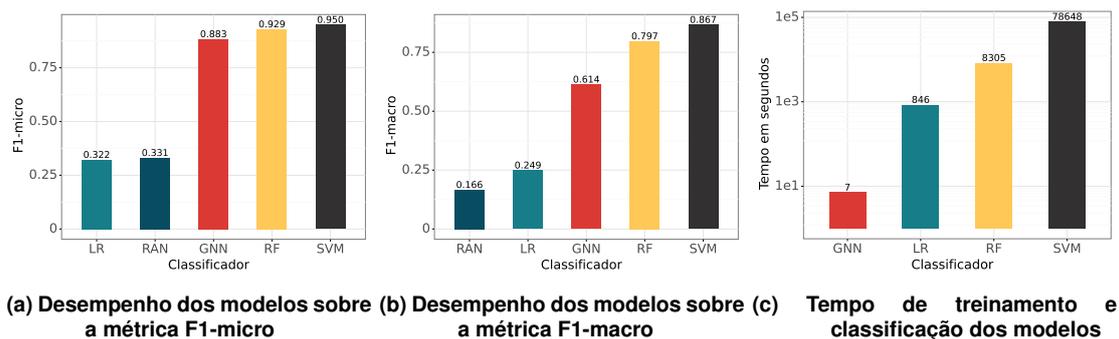


Figura 2. Desempenho dos melhores modelos sobre as métricas F1-micro e F1-macro e tempo de treinamento e classificação

concluir que ao usar GNN melhores resultados são obtidos se comparado com LR em todas as métricas. Por outro lado, a abordagem GNN não é capaz de fornecer resultados superiores àqueles do RF e do SVM, mantendo um desempenho global de 92% do melhor modelo clássico. Não obstante, o SVM apresenta resultados superiores àqueles obtidos por RF.

Uma análise mais detalhada dos resultados nos permite identificar que a etapa utilizando GNN não é capaz de classificar corretamente instâncias das três classes menos representativas na base de dados (i.e. SOR, C-LOR+OOR, C-LOR+SOR), alcançando, então, taxa de acerto próximo a zero. Desse modo, faz-se necessária a investigação de estratégias para lidar com o desbalanceamento de classes, seja do ponto de vista da base de dados seja do ponto de vista arquitetural de redes profundas. Além disso, os resultados superiores do SVM possui relação com o tamanho das dimensões escolhidas para a representação dos dispositivos, uma vez que esse modelo é conhecido por sua efetividade em altas dimensões.

Se por um lado, a abordagem utilizando GNN não fornece os melhores resultados, o que podemos concluir sobre o tempo de execução? A Figura 2c apresenta o tempo gasto em segundos para treinar e classificar os dados. Podemos concluir que GNN despendeu o menor tempo, aproximadamente 0.8% do tempo gasto por uma abordagem que utiliza modelos simples, como o LR. Um dos motivos para isso ocorrer é o tempo de execução na etapa de aprendizado de *embeddings*, realizado pelo `node2vec`, que levou em média 600 segundos para ocorrer. Com relação ao tempo de execução do SVM e RF, o GNN é cerca de 1186 e 11235 vezes menor, respectivamente. Dessa forma, apesar de não apresentar os melhores resultados na tarefa de classificação, o GNN despende um tempo para treinamento e classificação consideravelmente inferior ao seus pares. Portanto, por se tratar de um cenário dinâmico em que novos laços e dispositivos surgem e desaparecem a todo momento, a velocidade de treinamento e classificação é fator relevante. Isso porque, além de considerar a acurácia do modelo é importante que ele seja implementado em tempo hábil.

A Figura 3 complementa a argumentação ao apresentar o desempenho dos modelos em função da dimensão d dos *embeddings*. Percebe-se que o desempenho da abordagem utilizando GNN tende a aumentar até $d = 32$, realizando um leve decaimento na dimensão $d = 64$. Além disso, o modelo SVM tende a melhorar à medida que a dimensão aumenta. Por outro lado, o RF obteve a melhor desempenho para $d = 16$ e seu desempenho decresce conforme a dimensão aumenta, o que pode ser um indicativo da maldição

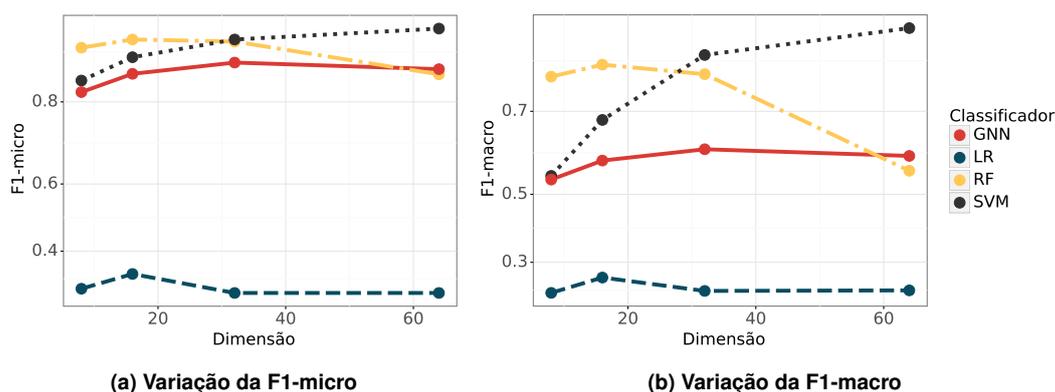


Figura 3. Desempenho dos modelos variando a dimensão da representação das relações

da dimensionalidade (do inglês *curse of dimensionality* [Bellman 1956]). Por fim, o LR obteve melhor resultado para $d = 16$, porém o desempenho desse modelo é consideravelmente inferior às outras abordagens, então, podemos suspeitar que LR é incapaz de aprender as não-linearidades existentes no dado. Portanto, modelos mais complexos se fazem necessários para solucionar o problema em questão de forma aceitável.

Em resumo, os resultados, dentro dos parâmetros testados, indicam que modelos simples como LR são pouco eficientes para identificar os padrões presentes dentro da rede de relacionamento da SIoT quando comparados com os outros classificadores utilizados. Além disso, o RF obteve o melhor desempenho dentre todos os modelos nas baixas dimensões, porém, o aumento da dimensão dos *embeddings* deteriora os resultados desse modelo. Por outro lado, o SVM obteve a maior acurácia sobretudo nas altas dimensões, no entanto, para isso necessitou de um tempo, no mínimo, 10 vezes maior de treinamento e classificação se comparado com seus pares. Por fim, embora o GNN não tenha atingido os melhores resultados, a principal vantagem desse modelo está no baixo tempo de treinamento e classificação, além da capacidade indutiva de reconhecer novos dispositivos que sejam adicionados na rede, sendo essa uma vantagem importante em um cenário dinâmico.

6. Conclusões

Os avanços na área da IoT possibilitam que dispositivos sejam capazes de realizar laços sociais uns com os outros, gerando então o paradigma SIoT. Por natureza, essas conexões podem representar múltiplos tipos de relacionamentos e obedecem às regras estabelecidas pelo proprietário. Entender como esses relacionamentos são estruturados dentro da rede, como também encontrar modelos capazes de classificar essas relações, é um problema em aberto dentro da literatura. Dessa forma, classificar os tipos de vínculos entre os dispositivos possibilita que estes sejam capazes de compartilhar conhecimentos e recursos de maneira distribuída, estabelecendo graus de confiança uns com os outros.

Nesse sentido, utilizar os avanços na área de aprendizado de máquina pode ser um caminho para encontrar técnicas escaláveis e com capacidade de identificar, compreender e prever padrões sociais dos dispositivos. Por isso, nesse artigo utilizamos abordagens de GNN e ATC com aprendizado por representação para classificar os tipos de laços sociais da IoT. Para mostrar a efetividade dessas abordagens, comparamos os modelos na tarefa de classificar conexões de múltiplos tipos em uma rede SIoT baseada em dispositi-

vos reais. Com os resultados pôde-se constatar que é possível classificar os tipos de laços sociais em redes SIoT utilizando modelos de aprendizado de máquina.

Especificamente falando, a abordagem de SVM em conjunto com o `node2vec` apresentou o melhor desempenho sobre as métricas $F1$ -*micro* e $F1$ -*macro*, seguida pelo RF e GNN. Com relação ao tempo de treinamento e classificação, os algoritmos GNN e LR executaram em tempo consideravelmente menor se comparado com outros modelos. Com isso, os resultados sugerem que a abordagem de GNN consegue atingir métricas $F1 - \{macro, micro\}$ 0.61 e 0.88, respectivamente, em um tempo 121 vezes menor se comparado com técnicas ATC sem a necessidade de retreino para a adição de novos nós. Ademais, as técnicas SVM e RF apresentam os melhores resultados, alcançando 0.95 e 0.92, respectivamente, sobre a métrica $F1 - micro$, ao custo de maior tempo de treinamento, classificação e necessidade de retreino. Porém, em um ambiente dinâmico como o da SIoT, é necessário equilíbrio entre tempo e acurácia dos resultados.

Pesquisas futuras devem ser realizadas a fim de se utilizar estratégias de balanceamento na base de dados, como por exemplo, a *Synthetic Minority Oversampling Technique* (SMOTE), para se obter exemplos sintéticos para as classes minoritárias e equilibrar sua distribuição. Não obstante, levar em consideração as características físicas dos dispositivos para a realização do *embedding*, de forma a explorar outros atributos além dos sociais. Além disso, a representação das arestas existentes entre os dispositivos deve ser testada com outros métodos, como os apresentados no artigo [Grover and Leskovec 2016] (*Average, Hadamard, Weighted-L1* e etc). Por fim, sabemos que as relações sociais não são permanentes e algumas delas podem se desfazer com o tempo, sendo assim, realizar previsões considerando intervalos de tempo pode trazer novas percepções à pesquisa.

Agradecimentos

Agradecemos ao CNPq, CAPES, FAPEMIG, FAPESB, UFOP e UFRB pelo apoio nacional e internacional para o desenvolvimento deste trabalho.

Referências

- Abdelrahman, M., Chong, A., and Miller, C. (2020). Build2vec: Building representation in vector space. *arXiv preprint arXiv:2007.00740*.
- Aljubairy, A., Zhang, W. E., Sheng, Q. Z., and Alhazmi, A. (2020). Siotpredict: A framework for predicting relationships in the social internet of things. In Dustdar, S., Yu, E., Salinesi, C., Rieu, D., and Pant, V., editors, *Advanced Information Systems Engineering*, Cham. Springer International Publishing.
- Atzori, L., Iera, A., and Morabito, G. (2014). From "smart objects" to "social objects": The next evolutionary step of the internet of things. *IEEE Communications Magazine*.
- Atzori, L., Iera, A., Morabito, G., and Nitti, M. (2012a). The social internet of things (siot) – when social networks meet the internet of things: Concept, architecture and network characterization. *Computer Networks*.
- Atzori, L., Iera, A., Morabito, G., and Nitti, M. (2012b). The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Computer networks*.

- Barker, S. and Rothmuller, M. (2020). The internet of things: Consumer, industrial & public services 2020-2024.
- Bellman, R. (1956). Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America*, 42(10):767.
- Corso, G., Cavalleri, L., Beaini, D., Liò, P., and Veličković, P. (2020). Principal neighbourhood aggregation for graph nets. *arXiv preprint arXiv:2004.05718*.
- Glasmachers, T. (2017). Limits of end-to-end learning. *Proceedings of Machine Learning Research*.
- Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*.
- Gupta, C., Jain, Y., De, A., and Chakrabarti, S. (2021). Integrating transductive and inductive embeddings improves link prediction accuracy. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.
- Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*.
- Hu, L., Wu, G., Xing, Y., and Wang, F. (2019). Things2Vec: Semantic modeling in the Internet of Things with graph representation learning. *IEEE Internet of Things Journal*.
- Júnior, J. J., Figueiredo, T. S., Torres, L. C., and Santos, B. P. (2021). Como os objetos da Internet das Coisas se comportam? Uma análise através de métricas quantitativas. In *Anais do V Workshop de Computação Urbana*. SBC.
- Khanfor, A., Ghazzai, H., Yang, Y., and Massoud, Y. (2019). Application of community detection algorithms on social internet-of-things networks. In *2019 31st International Conference on Microelectronics (ICM)*.
- Khanfor, A., Nammouchi, A., Ghazzai, H., Yang, Y., Haider, M. R., and Massoud, Y. (2020). Graph neural networks-based clustering for social internet of things. In *2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE.
- Marche, C., Atzori, L., and Nitti, M. (2018). A dataset for performance analysis of the social internet of things. In *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*.
- Marche, C., Atzori, L., Pilloni, V., and Nitti, M. (2020). How to exploit the social internet of things: Query generation model and device profiles' dataset. *Computer Networks*.
- Marche, C. and Nitti, M. (2020). Trust-related Attacks and their Detection: a Trust Management Model for the Social IoT. *IEEE Transactions on Network and Service Management*.
- Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. (2019). Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

- Nitti, M., Atzori, L., and Cvijikj, I. P. (2014). Network navigability in the social internet of things. In *2014 IEEE world forum on internet of things (WF-IoT)*. IEEE.
- Oliveau, Q. and Sahbi, H. (2018). From transductive to inductive semi-supervised attributes for ship category recognition. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*.
- Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA. Association for Computing Machinery.
- Sanchez-Lengeling, B., Reif, E., Pearce, A., and Wiltchko, A. B. (2021). A Gentle Introduction to Graph Neural Networks. *Distill*. <https://distill.pub/2021/gnn-intro>.
- Santos, B. P., Goussevskaia, O., Vieira, L. F., Vieira, M. A., and Loureiro, A. A. (2018). Mobile matrix: routing under mobility in IoT, IoMT, and social IoT. *Ad Hoc Networks*.
- Santos, B. P., Silva, L. A., Celes, C. S., Borges Neto, J. B., Peres, B. S., Vieira, M. A. M., Vieira, L. F. M., Goussevskaia, O. N., and Loureiro, A. A. (2016). Internet das coisas: da teoria à prática. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*.
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE transactions on neural networks*.
- Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*.
- Wei, D., Ning, H., Qian, Y., and Zhu, T. (2018). Social relationship for physical objects. *International Journal of Distributed Sensor Networks*.
- Xing, Y., Hu, L., Zhang, X., Wu, G., and Wang, F. (2021). Nonnegative Matrix Factorization Based Heterogeneous Graph Embedding Method for Trigger-Action Programming in IoT. *IEEE Transactions on Industrial Informatics*.
- Xu, D., Cheng, W., Luo, D., Liu, X., and Zhang, X. (2019). Spatio-temporal attentive rnn for node classification in temporal attributed graphs. In *IJCAI-19*.
- Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*.